

---

```

name: <unnamed>
log: /Users/connorbrennan/Library/CloudStorage/OneDrive-TheUniversityo
> fChicago/mmb/output/stepwise_regressions/y_timing_max_all_det.smcl
log type: smcl
opened on: 8 May 2025, 20:04:55
- capture ! rm ./output/stepwise_regressions/`depvar'`type'.pdf
= capture ! rm ./output/stepwise_regressions/y_timing_max_all_det.pdf
- di "*****"
> *****"
*****
> **
- di "Outcomes of bi-directional stepwise regressions with `depvar' with rule
> fixed effects"
= di "Outcomes of bi-directional stepwise regressions with y_timing_max with r
> ule fixed effects"
Outcomes of bi-directional stepwise regressions with y_timing_max with rule fi
> xed effects
- di "Independent Variable set: `type'"
= di "Independent Variable set: all_det"
Independent Variable set: all_det
- di "*****"
> *****"
*****
> **
- ${reg_`depvar'`type'}
= nbreg y_timing_max cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg
> wlth stky_pr_calvo ln_neq open if y_timing_max<60 , vce(r)

> ----- begin nbreg -----
- if _by() {
local BY "`by' `_byvars' `_byrc0':"
}
- `BY' _vce_parserun nbreg, mark(Exposure OFFset CLuster) : `0'
= _vce_parserun nbreg, mark(Exposure OFFset CLuster) : y_timing_max cb_au
> thors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq
> open if y_timing_max<60 , vce(r)

> ----- begin _vce_parserun -----
- version 9
- local version : di "version " string(_caller()) ":"
- local bycall = _by()
- _on_colon_parse `0'
= _on_colon_parse nbreg, mark(Exposure OFFset CLuster) : y_timing_max cb
> _authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_
> neq open if y_timing_max<60 , vce(r)

```

---

```

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before "`('before)'"
}
- if ``"before"' == ":" {
= if `nbreg, mark(Exposure OFFset CLuster) "' == ":" {
  sreturn local after ``"after"'
  exit
}
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``"' != ":" & ``: y_timing_max cb_authors_ext estimated wg_ndx
> _mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60
> , vce(r)"' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``"COLON"' != ":" {
= else if ``:"' != ":" {
  local before ``"before' `COLON"'
  local COLON
}
- }
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``:"' != ":" & `` y_timing_max cb_authors_ext estimated wg_ndx
> _mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60
> , vce(r)"' != "" {
  gettoken COLON after : after, parse(":") bind match(par) quotes
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``"COLON"' != ":" {
    local before ``"before' `COLON"'
    local COLON
  }
}
- if ``"COLON"' != ":" {

```

```

= if `:"" != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before ``before""
= sreturn local before `nbreg, mark(Exposure OFFset CLuster) ""
- sreturn local after ``after""
= sreturn local after ` y_timing_max cb_authors_ext estimated wg_ndx_
> mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60 ,
> vce(r)""

-----
> ----- end _on_colon_parse -----
- local ZERO ``s(before)""
= local ZERO `nbreg, mark(Exposure OFFset CLuster) ""
- local 0 ``s(after)""
= local 0 ` y_timing_max cb_authors_ext estimated wg_ndx_mult stky_pr_r
> otemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60 , vce(r)""
- quietly syntax [anything(equalok)] [if] [in] [fw aw pw iw] [, VCE(string
> asis) VCE1(string asis) LRMODEL * ]
- if ``vce"" == "" {
= if `r"" == "" {
MYRESULT
exit
}
- local AFTER ``0""
= local AFTER ` y_timing_max cb_authors_ext estimated wg_ndx_mult stky_
> pr_rotemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60 , vce(r)""
- local 0 ``ZERO""
= local 0 `nbreg, mark(Exposure OFFset CLuster) ""
- syntax name(name=caller id="calling command name") [, WTYPES(string) NOB
> OOTstrap NOJACKknife NOOTHERvce NOTEST PANELdata STdata bootopts(string asis
> ) jkopts(string asis) robustok multivce REQUIREDopts(namelist) NOEQLIST EQop
> ts(passthru) EQARGopts(passthru) noNEEDVARLIST COMMONopts(namelist) MARKopts
> (passthru) SMARKopts(passthru) RMCOLLopts(passthru) RMDCOLLopts(passthru) NU
> MDEPvars(passthru) NEEDEQUAL ALLDEPSMISSing PAREntheses UNPARFIRSTeq OR EQUA
> L UNEQUALFIRSTeq IGNORENOCONS ]
- if ``requiredopts"" != "" {
= if `"" != "" {
tempname reqopts
.`reqopts' = .optlist.new
foreach req of local requiredopts {
.`reqopts'.addopt `req', passthru
}
}
- if ``wtypes"" == "" {
= if "" == "" {

```

```

- local wtypes fw aw pw iw
- }
- if "`paneldata'" != "" {
= if "" != "" {
    local panelopt "CLuster(varname) I(varname)"
    }
- if "`stdata'" != "" {
= if "" != "" {
    local stopt "CLuster(varname) SHared(varname)"
    }
- local typelist svy jackknife bootstrap
- if "`stdata'" == "" & replay() & !_by() {
= if "" == "" & replay() & !_by() {
    local ecmd `e(cmd)'
    local ecmdname `e(cmdname)'
    if ``ecmd' == "" {
        error 301
    }
    else if ``ecmd' != ``caller' & ``ecmdname' == ``caller' & `:list
> ecmd in typelist' {
    `ecmd' `0'
    MYRESULT exit
    }
    MYRESULT
    exit
    }
- gettoken svy typelist : typelist
- if "`reqopts'" != "" {
= if "" != "" {
    local REQOPTLIST ``'.`reqopts'.dumpoptions'
    local REQOPTNAMES ``'.`reqopts'.dumpnames'
    }
- local 0 ``AFTER'
= local 0 `` y_timing_max cb_authors_ext estimated wg_ndx_mult stky_pr_r
> otemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60 , vce(r)'
- quietly syntax [anything(equalok)] [if/] [in] [`wtypes'] [, VCE(string a
> sis) VCE1(string asis) Robust NODOTS `panelopt' `stopt' `REQOPTLIST' * ]
= quietly syntax [anything(equalok)] [if/] [in] [fw aw pw iw] [, VCE(strin
> g asis) VCE1(string asis) Robust NODOTS * ]
- if `:list vce == vce1' {
= if 0 {
    local vce1
    }
- if ``vce1' != "" & ``multivce' == "" {
= if ``' != "" & "" == "" {
    di as err "option vce() specified multiple times"

```

```

    exit 198
  }
- if "`robust'" != "" & "`robustok'" == "" {
= if "" != "" & "" == "" {
    di as err "options vce() and robust may not be combined"
    exit 198
  }
- if ``"if'" != "" {
= if ``y_timing_max<60 "" != "" {
- local if ``if (`if)'"
= local if ``if (y_timing_max<60 )'"
- }
- if "`robust'" != "" {
= if "" != "" {
    local options ``robust' `options'"
  }
- if "`weight'" != "" {
= if "" != "" {
    local wgt [`weight'`exp']
  }
- if ``options'" != "" {
= if ``'" != "" {
    local opts ``:list retok options'"
  }
- gettoken uvcetype vceopts : vce, parse(" ,")
- capture unabVCetype, `uvcetype'
= capture unabVCetype, r

```

---

```

> ----- begin _vce_parserun.unabVCetype -----
- syntax [, B00Tstrap BStrap JACKknife JACKNife JKnife Robust * ]
- if "`bstrap'" != "" {
= if "" != "" {
    local bootstrap bootstrap
  }
- if "`jknife'" != "" | "`jackknife'" != "" {
= if "" != "" | "" != "" {
    local jackknife jackknife
  }
- local vcetype ``jackknife' `bootstrap' `robust' `options'"
= local vcetype ``robust'"
- sreturn local vcetype ``:list retok vcetype'"
= sreturn local vcetype ``robust'"

```

---

```

> ----- end _vce_parserun.unabVCetype -----
- if c(rc) {
    di as err "invalid vce() option"

```

```

    exit c(rc)
}
- local vcetype `s(vcetype)'
= local vcetype robust
- if "`lrmodel'"!="" {
= if ""!="" {
    _check_lrmodel `caller', vcetype(`vcetype')
}
- if "`vce1'"!="" & "`multivce'"!="" {
= if `""'!="" & ""!="" {
    gettoken uvceltype vcelopts : vce1, parse(" ,")
    capture unabVCEtype, `uvceltype'
    if c(rc) {
        di as err "invalid vce() option"
        exit c(rc)
    }
    local vceltype `s(vcetype)'
    if `:list vceltype in typelist' | `:list vcetype in typelist' {
        di as err "options vce(`vce') and vce(`vce1') cannot be combined"
        exit 198
    }
    MYRESULT
    exit
}
- if _caller() >= 17 {
- if "`vcetype'" == "cluster" {
= if "robust" == "cluster" {
    gettoken vceopts vcelopts: vceopts, parse(",")
    if `:list sizeof vceopts' > 1 {
        local typelist "`typelist' cluster_multiway"
        local vcetype cluster_multiway
        if `:list sizeof vcelopts' > 0 {
            gettoken comma vcelopts: vcelopts, parse(",")
            local vcelopts: list retokenize vcelopts
            di as err "{p}VCE option " "{bf:`vcelopts'} not allowed{p_end}"
            exit 198
        }
        if ("`cluster'"=="") & ("`noeqlist'"=="") {
            if "`caller'" == "regress" {
                Noc2Noconstant `opts'
                local opts ``s(opts)''
            }
            local opts ``opts' cluster(`vceopts)''
        }
    }
}
}
}

```

```

- }
- if ! `:list vcetype in typelist' {
= if ! 0 {
- if "`vcetype'" != "robust" & "`noothervce'" == "noothervce" {
= if "robust" != "robust" & "" == "noothervce" {
    di as err "option vce(`uvcetype') not allowed"
    exit 198
  }
- MYRESULT
}

> ----- begin _vce_parserun.MYRESULT -----
- args exit
- sreturn local exit ``exit'''
= sreturn local exit `'''

> ----- end _vce_parserun.MYRESULT -----
- exit

> ----- end _vce_parserun -----
- if "`s(exit)'" != "" {
= if "" != "" {
    version 10: ereturn local cmdline `nbreg `0'''
    exit
  }
- version 6, missing
- local version : di "version " string(_caller()) ":"
- if replay() {
    if "`e(cmd)'" != "nbreg" { error 301 }
    if _by() { error 190 }
    global S_1 = e(chi2_c)
    Display `0'
    error `e(rc)'
    exit
  }
- `version' `BY' Estimate `0'
= version 18: Estimate y_timing_max cb_authors_ext estimated wg_ndx_mult
> stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if y_timing_max<60 , vce(
> r)

> ----- begin nbreg.Estimate -----
- local vvpoui : di "version " string(_caller()) ", missing:"
- if _caller() >= 11 {
- local vv : di "version " string(_caller()) ":"
- local mm e2
- local negh negh
- }

```

```

- else {
  local vv "version 8.1:"
  local mm d2
}
- version 6, missing
- local awopt = cond(_caller()<7, "aw", "")
- syntax varlist(numeric fv ts) [fw `awopt' pw iw] [if] [in] [, IRr FROM(s
> tring) Level(cilevel) OFFset(varname numeric ts) Exposure(varname numeric ts
> ) noCONstant Robust CLuster(varname) SCore(string) NOLOg LOg noDISPLAY noLRt
> est Dispersion(string) CRITTYPE(passthru) VCE(passthru) moptobj(passthru) *
> ]
  = syntax varlist(numeric fv ts) [fw pw iw] [if] [in] [, IRr FROM(string)
> Level(cilevel) OFFset(varname numeric ts) Exposure(varname numeric ts) noCON
> stant Robust CLuster(varname) SCore(string) NOLOg LOg noDISPLAY noLRtest Dis
> version(string) CRITTYPE(passthru) VCE(passthru) moptobj(passthru) * ]
  - local fvops = "`s(fvops)'" == "true" | _caller() >= 11
  = local fvops = "" == "true" | _caller() >= 11
  - if _by() {
    _byoptnotallowed score() ``score'``
  }
  - _get_diopts diopts options, `options'
  = _get_diopts diopts options,

```

---

```

> ----- begin _get_diopts -----
  - version 11
  - syntax namelist(max=2) [, *]
  - gettoken name1 namelist : namelist
  - gettoken name2 namelist : namelist
  - GetDiopts diopts options, `options'
  = GetDiopts diopts options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
  - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels NO
> BASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NOOM
> ITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sforma
> t(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(st
> ring) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV AB
> BREV
  - syntax namelist(max=2) [, `DIOPTS' *]
  = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels ALL
> BASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EM
> PTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(s
> tring) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru
> ) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru
> ) NOABBREV ABBREV *]
  - opts_exclusive "`allbaselevels' `noallbaselevels'"

```



```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``coeflegend' `selegend'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``noci' `nopvalues'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts'"' != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts'"' != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
}
}

```

```

- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `pformat(`pformat')'
  }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']

```

```

    }
    - if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
    - Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
    - capture syntax [, W0rd WIdth]
    - if c(rc) {
        di as err "invalid fvwrapon() option;"
        syntax [, W0rd WIdth]
        exit 198
    }
    - opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
= local fvwrapon
    - if `:length local fvwrapon' {
= if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
= if "" != "" {
        FVWrap, `fvwrap'
    }

```



```

- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevels'
> `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend'
> `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwrap
> ' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
= c_local diopts
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options'''
= c_local options `'''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts'''
= c_local diopts `'''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels NO
> BASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NOOM
> ITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sforma
> t(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(st
> ring) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV AB
> BREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels ALL
> BASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EM
> PTYcells NOOMITTED OMITTED NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(s
> tring) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru
> ) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru
> ) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels'''
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`emptycells` `noemptycells`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel` `nofvlabel`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }

```

```

- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `"cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `"sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `"pformat(`pformat')"'
  }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
  }

```



```

- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
    }
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
    }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevels'

```

```

> `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend'
> `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwrap
> ' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
    = c_local DEFAULT
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
    - if ``name2'' == "" {
    = if "options" == "" {
        GetDiopts DUPS, `options'
    }
    - else {
    - GetDiopts DUPS options, `options'
    = GetDiopts DUPS options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels NO
> BASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NOOM
> ITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sforma
> t(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(st
> ring) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV AB
> BREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels ALL
> BASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EM
> PTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(s
> tring) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru
> ) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru
> ) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

```

```

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive ""`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

> 

---

 end opts\_exclusive 

---

```

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted

```



```

    }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`"cformat"'') > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat `"cformat"'
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`"sformat"'') > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat `"sformat"'
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`"pformat"'') > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `"pformat"'
    local pformat `pformat(`pformat')'
  }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
  }
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
  }
}

```

```

- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd Width]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd Width]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}

```

---

```

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevels'
> `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend'
> `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwrap
> ' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
= c_local DUPS

```

```

- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options''
= c_local options ``''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95
}

> ----- end _get_diopts -----
- c_local `name2' ``options''
= c_local options ``''
- }
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'' != "" {
= if ``'' != "" {
if (strpos(`DUPS', "(")) {
gettoken DUPS : DUPS, parse("(")
local DUPS `DUPS'()'
}
di as err "duplicate {bf:`DUPS'} option not allowed"
exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname''
= sreturn local level = ``95''
- }
}

> ----- end _get_diopts -----
- mlopts mlopts, `options' `log' `nolog'
= mlopts mlopts,

> ----- begin mlopts -----
- version 8.0
- local maxoptlist MLOPTs(string) TRace TRACEDiffs GRADient HESSian SHOW
> STEP ITERate(int -1) TOLerance(real -1) LTOLerance(real -1) GTOLerance(real
> -1) NONRTOLerance SHOWNRTolerance SHOWTOLerance NRTOLerance(real -1) QTOLera
> nce(real -1) DIFFicult HALFSTEPonly CONSTraints(string) noCNSNOTes COLLinear
> TECHnique(passthru) VCE(passthru) DOTS depsilon(passthru) SHOWH NOTCONCAVE(

```

```

> passthru) USEVIEWS NOLOg LOg nrderiv
    - syntax namelist(id="macro names" min=1 max=2) [, `maxoptlist' * ]
    = syntax namelist(id="macro names" min=1 max=2) [, MLOPTs(string) TRace
> TRACEDIffs GRADient HESSian SHOWSTEP ITERate(int -1) TOLerance(real -1) LTOL
> erance(real -1) GTOLerance(real -1) NONRTOLerance SHOWNRtolerance SHOWTOLera
> nce NRTOLerance(real -1) QTOLerance(real -1) DIFFicult HALFSTEPonly CONSTrai
> nts(string) noCNSNOTes COLlinear TECHnique(passthru) VCE(passthru) DOTS deps
> ilon(passthru) SHOWH NOTCONCAVE(passthru) USEVIEWS NOLOg LOg nrderiv * ]
    - gettoken dest rest : namelist
    - if "`rest'" == "" {
    = if "" == "" {
    - syntax namelist [, `maxoptlist' ]
    = syntax namelist [, MLOPTs(string) TRace TRACEDIffs GRADient HESSian SH
> OWSTEP ITERate(int -1) TOLerance(real -1) LTOLerance(real -1) GTOLerance(rea
> l -1) NONRTOLerance SHOWNRtolerance SHOWTOLerance NRTOLerance(real -1) QTOLe
> rance(real -1) DIFFicult HALFSTEPonly CONSTraints(string) noCNSNOTes COLline
> ar TECHnique(passthru) VCE(passthru) DOTS depsilon(passthru) SHOWH NOTCONCAV
> E(passthru) USEVIEWS NOLOg LOg nrderiv ]
    - }
    - local mlopts `mlopts' `trace' `tracediffs' `gradient' `hessian' `shows
> tep' `difficult' `halfsteponly' `nonrtolerance' `shownrtolerance' `showtoler
> ance' `dots' `depsilon' `showh' `notconcave' `useviews' `log' `nolog' `nrder
> iv'
    = local mlopts
    - if `tolerance' != -1 {
    = if -1 != -1 {
        local tol "tolerance(`tolerance')"
    }
    - if `ltolerance' != -1 {
    = if -1 != -1 {
        local ltol "ltolerance(`ltolerance')"
    }
    - if `gtolerance' != -1 {
    = if -1 != -1 {
        local gtol "gtolerance(`gtolerance')"
    }
    - if `nrtolerance' != -1 {
    = if -1 != -1 {
        local nrtol "nrtolerance(`nrtolerance')"
    }
    - if `qtolerance' != -1 {
    = if -1 != -1 {
        local qtol "qtolerance(`qtolerance')"
    }
    - if `iterate' != -1 {
    = if -1 != -1 {

```

```

        local iter "iterate(`iterate')"
    }
    - sreturn clear
    - if "`technique'\vce'" != "" {
= if "" != "" {
    ml_technique, `technique' `vce' nodefault
    if ``s(technique)'" != "" {
        local technique "technique(`s(technique))"
    }
    if "`s(vce)'" != "" {
        local vce "`s(vceopt)'"
    }
    sreturn local technique "`technique'"
    sreturn local vce "`vce'"
}
    - if "`constraints'" != "" {
= if "" != "" {
    capture confirm name `constraints'
    if _rc {
        CheckClist constraints , constraints(`constraints')
    }
    else {
        CheckCmat `constraints'
    }
    local const "constraints(`constraints)'"
    sreturn local constraints "`constraints'"
}
    - sreturn local collinear "`collinear'"
= sreturn local collinear ""
    - c_local `dest' `mlopts' `tol' `ltol' `gtol' `nrtol' `iter' `const' `cn
> snotes' `collinear' `technique' `vce' `qtol'
    = c_local mlopts
    - if "`rest'" != "" {
= if "" != "" {
    c_local `rest' ``options'"
}



---


> 

---

 end mlopts 

---


    - local coll `s(collinear)'
= local coll
    - local cns `s(constraints)'
= local cns
    - local mlopts `mlopts' `critttype'
= local mlopts
    - Dispers , `dispers'
= Dispers ,

```

```

> ----- begin nbreg.Dispers -----
- sret clear
- syntax [, Mean Constant ]
- if "`constan'"==" " {
= if ""==" " {
- sret local dispers "mean"
- exit

> ----- end nbreg.Dispers -----
- local dispers `s(dispers)'
= local dispers mean
- if "`dispers'"=="mean" {
= if "mean"=="mean" {
- local prog "nbreg_lf"
- local parm "alpha"
- local LLprog "LLalpha"
- }
- else {
  local prog "nbreg_al"
  local parm "delta"
  local LLprog "LLdelta"
}
- local title "Negative binomial regression"
- if ``score'""!=" " {
= if `""'!=" " {
  local nword : word count `score'
  if `nword'==1 & bsubstr("`score'",-1,1)=="*" {
    local score = bsubstr("`score'",1,length("`score'")-1)
    local score `score'1 `score'2
    local nword 2
  }
  confirm new variable `score'
  if `nword' != 2 {
    version 7: di as err "option {bf:score()} must contain the name of " "tw
> o new variables"
    exit 198
  }
  local scname1 : word 1 of `score'
  local scname2 : word 2 of `score'
  tempvar scvar1 scvar2
  local scopt "score(`scvar1' `scvar2')"
}
- if "`offset'"!=" " & "`exposur'"!=" " {
= if ""!=" " & ""!=" " {
  version 7: di as err "options {bf:offset()} and {bf:exposure()} may not

```

```

> be specified together"
    exit 198
}
- if "`constan'"!="" {
= if ""!="" {
    local nvar : word count `varlist'
    if `nvar' == 1 {
        version 7: di as err "independent variables required with " "option {bf:
> noconstant}"
        exit 102
    }
}
- marksample touse
- if ``cluster'"!="" {
= if `""'!="" {
    local clopt cluster(`cluster')
}
- _vce_parse, argopt(Cluster) opt(Robust oim opg) old: [`weight'`exp'], `v
> ce' `clopt' `robust'
= _vce_parse, argopt(Cluster) opt(Robust oim opg) old: [], vce(r)

> ----- begin _vce_parse -----
- version 10
- _on_colon_parse `0'
= _on_colon_parse , argopt(Cluster) opt(Robust oim opg) old: [], vce(r)
>
> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
    local before `"(before)'"
}
- if ``before'" == ":" {
= if `", argopt(Cluster) opt(Robust oim opg) old"' == ":" {
    sreturn local after ``after'"
    exit
}
- while ``COLON'" != ":" & ``after'" != "" {
= while `""' != ":" & ``: [] , vce(r) '' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {

```

```

        local before `before' (`COLON')
        local COLON
    }
- else if `"'COLON'" != ":" {
= else if `":'" != ":" {
    local before `"'before' `COLON'"
    local COLON
    }
- }
- while `"'COLON'" != ":" & `"'after'" != "" {
= while `":'" != ":" & `"' [], vce(r) "" != "" {
    gettoken COLON after : after, parse(":") bind match(par) quotes
    if "`par'" != "" {
        local before `before' (`COLON')
        local COLON
    }
    else if `"'COLON'" != ":" {
        local before `"'before' `COLON'"
        local COLON
    }
}
- if `"'COLON'" != ":" {
= if `":'" != ":" {
    di as err "" found where ':' expected"
    exit 198
}
- sreturn local before `"'before'"
= sreturn local before `"', argopt(CLister) opt(Robust oim opg) old""
- sreturn local after `"'after'"
= sreturn local after `"' [], vce(r) ""



---


> ----- end _on_colon_parse -----
- local 0 `"'s(before)'"
= local 0 `"', argopt(CLister) opt(Robust oim opg) old""
- local user `"'s(after)'"
= local user `"' [], vce(r) ""
- syntax [varname(default=none)], [ OPTlist(namelist) ARGOPTlist(namelis
> t) PWallowed(namelist) OLD_is_ok NOARGOPT_is_ok ]
- local touse `varlist'
= local touse
- tempname o a
- .`o' = . _optlist.new
= .__000001 = . _optlist.new
- class {
- instance:
-         array options = {}

```



```

-         array names      = {}
-    }
- local optlist : list uniq optlist
- foreach name of local optlist {
-     .`o'.addopt `name'
= .__000001.addopt Robust
}

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("Robust")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "Robust"
- .names.Arrpush "`mname'"
= .names.Arrpush "robust"
}

> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= .__000001.addopt oim
}

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("oim")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "oim"
- .names.Arrpush "`mname'"
= .names.Arrpush "oim"
}

> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= .__000001.addopt opg
}

> ----- begin _optlist.addopt -----

```

```

- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("opg")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
  }
- .options.Arrpush "`optspec'"
= .options.Arrpush "opg"
- .names.Arrpush "`mname'"
= .names.Arrpush "opg"

```

---

```

> ----- end _optlist.addopt -----
- }
- .`a' = ._optlist.new
= .__000002 = ._optlist.new
- local argoptlist : list uniq argoptlist
- foreach name of local argoptlist {
- .`a'.addopt `name'
= .__000002.addopt CCluster

```

---

```

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("CLuster")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
  }
- .options.Arrpush "`optspec'"
= .options.Arrpush "CLuster"
- .names.Arrpush "`mname'"
= .names.Arrpush "cluster"

```

---

```

> ----- end _optlist.addopt -----
- }
- local OPTNAMES ```.`o'.dumpnames''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/3 {
- local namelist `namelist' `.names[`i']'
= local namelist robust

```

```

- }
- local namelist `namelist' `.names['i']'
= local namelist robust oim
- }
- local namelist `namelist' `.names['i']'
= local namelist robust oim opg
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist "robust oim opg"
- class exit ``namelist'''
= class exit "robust oim opg"

```

---

```

> ----- end _optlist.dumpnames -----
= local OPTNAMES "robust oim opg"
- local ARGOPTNAMES ```.a'.dumpnames'''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arrnels' {
= forval i = 1/1 {
- local namelist `namelist' `.names['i']'
= local namelist cluster
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist "cluster"
- class exit ``namelist'''
= class exit "cluster"

```

---

```

> ----- end _optlist.dumpnames -----
= local ARGOPTNAMES "cluster"
- local AND : list OPTNAMES & ARGOPTNAMES
- if `:list sizeof AND' {
= if 0 {
di as err "vcetype ``:word 1 of `AND'' specified twice"
exit 198
}
- local pwallowed : list uniq pwallowed
- local pwlist robust cluster
- foreach n of local pwallowed {
local lc = lower("`n'")
local pwlist `pwlist' `lc'
}
- if "`old_is_ok'" != "" {
= if "old_is_ok" != "" {

```

```

- local ROBUST robust
- if `:list ROBUST in OPTNAMES' {
= if 1 {
- local robust Robust
- }
- local CLUSTER cluster
- if `:list CLUSTER in ARGOPTNAMES' {
= if 1 {
- local cluster CLuster(varname)
- }
- }
- local 0 ``user'''
= local 0 `` [], vce(r) ''
- syntax [fw aw pw iw] [, VCE(string asis) `robust' `cluster' ]
= syntax [fw aw pw iw] [, VCE(string asis) Robust CLuster(varname) ]
- if ``vce''' != "" {
= if ``r''' != "" {
- gettoken name args : vce, parse(" ,")
- local args : list retok args
- capture ParseVCE `o' `a' , `name'
= capture ParseVCE __000001 __000002 , r

```

---

```

> ----- begin _vce_parse.ParseVCE -----
- gettoken 0 0 : 0, parse(" ,")
- gettoken A 0 : 0, parse(" ,")
- local OPTSPECS ```.`0'.dumpoptions' `.`A'.dumpoptions'''

```

---

```

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/3 {
- local optlist `optlist' `.options['i']'
= local optlist Robust
- }
- local optlist `optlist' `.options['i']'
= local optlist Robust oim
- }
- local optlist `optlist' `.options['i']'
= local optlist Robust oim opg
- }
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist ``Robust oim opg'''
- class exit ``optlist'''
= class exit ``Robust oim opg'''

```

---

```

> ----- end _optlist.dumpoptions -----
> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/1 {
- local optlist `optlist' `.options[`i']'
= local optlist CCluster
- }
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist `CCluster''
- class exit ``optlist'''
= class exit `CCluster''
> ----- end _optlist.dumpoptions -----
= local OPTSPECS `Robust oim opg CCluster''
- local OPTNAMES `"".`0'.dumpnames' `.`A'.dumpnames'''
> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/3 {
- local namelist `namelist' `.names[`i']'
= local namelist robust
- }
- local namelist `namelist' `.names[`i']'
= local namelist robust oim
- }
- local namelist `namelist' `.names[`i']'
= local namelist robust oim opg
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `robust oim opg''
- class exit ``namelist'''
= class exit `robust oim opg''
> ----- end _optlist.dumpnames -----
> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/1 {
- local namelist `namelist' `.names[`i']'
= local namelist cluster

```

```

- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist "cluster"
- class exit ``namelist'''
= class exit "cluster"

```

---

```

> ----- end _optlist.dumpnames -----
= local OPTNAMES "robust oim opg cluster"
- syntax [, `OPTSPECS']
= syntax [, Robust oim opg CLuster]
- foreach OPT of local OPTNAMES {
- if ``OPT''' != "" {
= if "robust" != "" {
- return local vce `OPT'
= return local vce robust
- continue, break
}
}

```

---

```

> ----- end _vce_parse.ParseVCE -----
- if c(rc) {
di as err "vcetype {bf:`name'} not allowed"
exit 198
}
- local name `r(vce)'
= local name robust
- if "weight" == "pweight" & !`:list name in pwlist' {
= if "" == "pweight" & !1 {
di as err "pweights are not allowed with vce(`name')"
exit 198
}
- return add
- if ``robust''' != "" & !inlist("`name'", "", "robust", "eim", "oim", "clus
> ter") {
= if ``''' != "" & !inlist("robust", "", "robust", "eim", "oim", "cluster") {
opts_exclusive "`robust' vce(`name')"
}
- if "`name'" == "cluster" {
= if "robust" == "cluster" {
if ``args''' == "" {
di as err "invalid vce(cluster) option"
exit 198
}
confirm var `args'
unab args : `args'

```

```

local markout `args'
if "`cluster'" != "" {
  if !`:list args == cluster' {
    di as err "options cluster() and vce(cluster) are in conflict"
    exit 100
  }
}
}
- else if "`cluster'" != "" & !inlist("`name'", "eim", "oim") {
= else if "" != "" & !inlist("robust", "eim", "oim") {
  opts_exclusive "cluster() vce(`name')"
}
- if ```args'"" != "" {
= if `""' != "" {
  if !`:list name in ARGOPTNAMES' {
    di as err "invalid vce() option"
    exit 198
  }
  if "`name'" == "hc2" {
    confirm var `args'
    unab args : `args'
    local markout `args'
  }
  return local vceargs ```:list retok args'""
}
- else if `:list name in ARGOPTNAMES' {
= else if 0 {
  if "`noargopt_is_ok'" == "" {
    di as err "invalid vce() option"
    exit 198
  }
}
- }
- else {
  if "`weight'" == "pweight" | "`robust'" != "" {
    local vce robust
  }
  if "`cluster'" != "" {
    local vce cluster
    local args `cluster'
    local markout `cluster'
  }
  return local vce ```:vce'""
  return local vceargs ```:args'""
}
- if "`return(vce)'" != "" {

```

```

= if "robust" != "" {
- if ``return(vceargs)`` != "" {
= if ```` != "" {
  local ss " "
}
- return local vceopt ``vce(`return(vce)``ss``return(vceargs)``)``
= return local vceopt ``vce(robust)``
- }
- if ``return(vce)`` == "cluster" {
= if "robust" == "cluster" {
  return local cluster `return(vceargs)`
}
- if inlist("`return(vce)`, "robust", "cluster") {
= if inlist("robust", "robust", "cluster") {
- return local robust robust
- }
- if ``touse`` != "" {
= if "" != "" {
  if ``markout`` != "" {
    markout `touse' `markout', strok
  }
}
}

```

---

```

> ----- end _vce_parse -----
- local cluster `r(cluster)'
= local cluster
- local robust `r(robust)'
= local robust robust
- local vce ``r(vceopt)``
= local vce ``vce(robust)``
- if ``cluster`` != "" {
= if ```` != "" {
  markout `touse' `cluster', strok
}
- if ``exposur`` != "" {
= if "" != "" {
  capture assert `exposur' > 0 if `touse'
  if _rc {
    version 7: di as err "option {bf:exposure()} must be greater than zero"
    exit 459
  }
  tempvar offset
  qui gen double `offset' = ln(`exposur')
  local offvar "ln(`exposur')"
}
- if ``offset`` != "" {

```



```

= if ""!=" {
  markout `touse' `offset'
  local offopt "offset(`offset')"
  if "`offvar'"==" {
    local offvar "`offset'"
  }
}
- gettoken y xvars : varlist
- _fv_check_depvar `y'
= _fv_check_depvar y_timing_max
- tsunab y : `y'
= tsunab y : y_timing_max

> ----- begin tsunab -----
- version 6
- gettoken user 0: 0, parse(" :")
- gettoken colon 0: 0, parse(" :")
- if ``colon' != ":" { error 198 }
= if `:"' != ":" { error 198 }
- syntax [varlist(ts default=empty)] [, MIN(integer 1) MAX(integer 32000
> ) NAME(string)]
- c_local `user' `varlist'
= c_local y y_timing_max
- local n : word count `varlist'
= local n : word count y_timing_max
- if `n'>=`min' & `n'<=`max' { exit }
= if 1>=1 & 1<=32000 { exit }

> ----- end tsunab -----
- local yname : subinstr local y "." "_"
- if "`weight'"!=" {
= if ""!=" {
  if "`weight'"=="fweight" {
    local wt `[fw`exp']'
  }
  else local wt `[aw`exp']'
}
- summarize `y' `wt' if `touse', meanonly
= summarize y_timing_max if __000000, meanonly
- if r(N) == 0 { error 2000 }
- if r(N) == 1 { error 2001 }
- if r(min) < 0 {
  version 7: di as err "{bf:`y'} must be greater than or equal to zero"
  exit 459
}
- if r(min) == r(max) & r(min) == 0 {

```

```

    version 7: di as err "{bf:`y'}` is zero for all observations"
    exit 498
  }
- tempname mean
- scalar `mean' = r(mean)
= scalar __000001 = r(mean)
- if "`display'"==" " {
= if ""==" " {
- capture assert `y' == int(`y') if `touse'
= capture assert y_timing_max == int(y_timing_max) if __000000
- if _rc {
    di in gr "note: you are responsible for " "interpretation of non-count d
> ep. variable"
  }
- }
- if "`display'"==" " & "`weight'"=="aweight" {
= if ""==" " & ""=="aweight" {
    di in gr "note: you are responsible for interpretation " "of analytic we
> ights"
  }
- if `fvops' {
= if 1 {
- local rmcoll "version 11: _rmcoll"
- }
- else local rmcoll _rmcoll
- if "`display'"!=" " & "`weight'"=="aweight" {
= if ""!=" " & ""=="aweight" {
    `rmcoll' `xvars' [iw`exp'] if `touse', `constan' `coll'
  }
- else `rmcoll' `xvars' [weight`exp'] if `touse', `constan' `coll'
= else version 11: _rmcoll cb_authors_ext estimated wg_ndx_mult stky_pr_r
> otemberg wlth stky_pr_calvo ln_neq open [] if __000000,
- local xvars `r(varlist)'
= local xvars cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth
> stky_pr_calvo ln_neq open
- _parse_iterlog, `log' `nolog'
= _parse_iterlog,

```

---

```

> ----- begin _parse_iterlog -----
- version 16
- syntax [, NOLOG LOG]
- if "`log'`nolog'" != "" {
= if "" != "" {
    opts_exclusive "`log' `nolog'"
  }
- else if "`c(iterlog)'" == "on" {

```

```

= else if "on" == "on" {
- local log log
- }
- else {
  local nolog nolog
  }
- sreturn local log `log'
= sreturn local log log
- sreturn local nolog `nolog'
= sreturn local nolog

```

---

```

> ----- end _parse_iterlog -----
- local log ``s(nolog)''
= local log ""
- if ``log'!='' | ``display'!='' { local nohead "*" }
= if ''!='' | ''!='' { local nohead "*" }
- if ``from'=='' & ``lrtest'=='' {
= if ''=='' & ''=='' {
- if ``robust'`cns'`cluster'!='' | ``weight'=="pweight" {
= if "robust"!='' | ''=="pweight" {
- local lrtest "nolrtest"
- }
- `nohead' di in gr _n "Fitting Poisson model:"
= di in gr _n "Fitting Poisson model:"

```

Fitting Poisson model:

```

- `vvpoi' poisson `y' `xvars' [`weight'`exp'] if `touse', nodisplay `offop
> t' `constan' `mlopts' `robust'
= version 18, missing: poisson y_timing_max cb_authors_ext estimated wg_nd
> x_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open [] if __000000, nodi
> splay robust

```

---

```

> ----- begin poisson -----
- if _by() {
  local BY ``by' `_byvars' `_byrc0':''
  }
- `BY' _vce_parserun poisson, mark(Exposure OFFset CLuster) : `0'
= _vce_parserun poisson, mark(Exposure OFFset CLuster) : y_timing_max c
> b_authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_
> neq open [] if __000000, nodisplay robust

```

---

```

> ----- begin _vce_parserun -----
- version 9
- local version : di "version " string(_caller()) ":"
- local bycall = _by()
- _on_colon_parse `0'

```

```

= _on_colon_parse poisson, mark(Exposure OFFset CLuster) : y_timing_ma
> x cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo
> ln_neq open [] if __000000, nodisplay robust

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
    local before `("`before')"'
}
- if ``"before"' == ":" {
= if ``poisson, mark(Exposure OFFset CLuster) "' == ":" {
    sreturn local after ``"after"'
    exit
}
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``"' != ":" & ``": y_timing_max cb_authors_ext estimated wg_n
> dx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open [] if __000000, nod
> isplay robust"' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
    local before `before' (`COLON')
    local COLON
}
- else if ``"COLON"' != ":" {
= else if ``:"' != ":" {
    local before ``"before' `COLON"'
    local COLON
}
- }
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``:"' != ":" & ``" y_timing_max cb_authors_ext estimated wg_n
> dx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open [] if __000000, nod
> isplay robust"' != "" {
    gettoken COLON after : after, parse(":") bind match(par) quotes
    if "`par'" != "" {
        local before `before' (`COLON')
        local COLON
    }
    else if ``"COLON"' != ":" {
        local before ``"before' `COLON"'
        local COLON
    }

```

```

    }
    }
    - if ``COLON`` != ":" {
= if ``:`` != ":" {
    di as err "" found where ':' expected"
    exit 198
    }
    - sreturn local before ``before``
    = sreturn local before ``poisson, mark(Exposure OFFset CLuster) ``
    - sreturn local after ``after``
    = sreturn local after `` y_timing_max cb_authors_ext estimated wg_nd
> x_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open [] if __000000, nodi
> splay    robust``
    _____
> _____ end _on_colon_parse _____
    - local ZERO ``s(before)``
    = local ZERO ``poisson, mark(Exposure OFFset CLuster) ``
    - local 0 ``s(after)``
    = local 0 `` y_timing_max cb_authors_ext estimated wg_ndx_mult stky_pr
> _rottemberg wlth stky_pr_calvo ln_neq open [] if __000000, nodisplay    robus
> t``
    - quietly syntax [anything(equalok)] [if] [in] [fw aw pw iw] [, VCE(st
> ring asis) VCE1(string asis) LRMODEL * ]
    - if ``vce`` == "" {
    = if ```` == "" {
    - MYRESULT
    _____
> _____ begin _vce_parserun.MYRESULT _____
    - args exit
    - sreturn local exit ``exit``
    = sreturn local exit ````
    _____
> _____ end _vce_parserun.MYRESULT _____
    - exit
    _____
> _____ end _vce_parserun _____
    - if ``s(exit)`` != "" {
= if "" != "" {
    version 10: ereturn local cmdline ``poisson `0``
    exit
    }
    - version 6, missing
    - local version : di "version " string(_caller()) ":"
    - if replay() {
        if ``e(cmd)`` != "poisson" {
            error 301

```

```

    }
    if _by() { error(190) }
    Display `0'
    error `e(rc)'
    exit
  }
  - `version' `BY' Estimate `0'
  = version 18: Estimate y_timing_max cb_authors_ext estimated wg_ndx_mul
> t stky_pr_rottemberg wlth stky_pr_calvo ln_neq open [] if __000000, nodisplay
> robust

----- begin poisson.Estimate -----
>
  - local vv "version 8.1:"
  - local mm d2
  - local eval poiss_lf
  - version 6, missing
  - local awopt = cond(_caller()<7, "aw", "")
  - syntax varlist(numeric fv ts) [fw `awopt' pw iw] [if] [in] [, IRr FR
> OM(string) Level(cilevel) OFFset(varname numeric ts) Exposure(varname numeri
> c ts) noCONstant Robust CLuster(varname) SCore(string) noDISPLAY CRITTYPE(pa
> ss thru) MATAevaluator moptobj(passthru) * ]
    = syntax varlist(numeric fv ts) [fw pw iw] [if] [in] [, IRr FROM(stri
> ng) Level(cilevel) OFFset(varname numeric ts) Exposure(varname numeric ts) n
> oCONstant Robust CLuster(varname) SCore(string) noDISPLAY CRITTYPE(passthru)
> MATAevaluator moptobj(passthru) * ]
    - local fvops = "`s(fvops)'" == "true" | _caller() >= 11
    = local fvops = "" == "true" | _caller() >= 11
    - if `fvops' {
    = if 1 {
    - local vv : di "version " string(max(11,_caller())) ", missing:"
    - local mm e2
    - if "`mataevaluator'" != "" {
    = if "" != "" {
      local eval mopt__poisson_e2()
    }
    - local negh negh
    - }
    - if _by() {
      _byoptnotallowed score ``score'""
    }
    - _get_diopts diopts options, `options'
    = _get_diopts diopts options,

----- begin _get_diopts -----
>
  - version 11
  - syntax namelist(max=2) [, *]

```

```

- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevel
> s NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells
> NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sf
> ormat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAP0
> N(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBRE
> V ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels
> ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcell
> s EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cform
> at(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(pass
> thru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(pass
> thru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```



```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'`` != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'`` != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----

```

```

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted

```

```

- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'

```

```

if fmtwidth("`sformat'") > 8 {
local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat "`sformat'"
local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth("`pformat'") > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat "`pformat'"
local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, W0rd WIdth]
exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselev
> els' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflege
> nd' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fv
> wrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
    = c_local diopts
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options `''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

    - c_local `name1' ``diopts''
    = c_local diopts `''
    - local snames : s(macros)
    - foreach sname of local snames {
    - local S_`sname' = s(`sname')
    = local S_level = s(level)

```

```

- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevel
> s NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells
> NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sf
> ormat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAP0
> N(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBRE
> V ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels
> ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcell
> s EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cform
> at(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(pass
> thru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(pass
> thru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```



```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'

```

```

= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {

```

```

local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat ``sformat''
local sformat `sformat(sformat)''
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth(`pformat') > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat ``pformat''
local pformat `pformat(pformat)''
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, W0rd WIdth]
exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
local ++n

```

```

        gettoken item`n' opts: opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
}
}
> ----- end opts_exclusive -----

    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }
}
> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselev
> els' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflege
> nd' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fv
> wrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abbrev'
    = c_local DEFAULT
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options'''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
}
> ----- end _get_diopts.GetDiopts -----

    - if ``name2''' == "" {
    = if ``options''' == "" {
        GetDiopts DUPS, `options'
    }
    - else {
    - GetDiopts DUPS options, `options'
    = GetDiopts DUPS options,

```



```

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevel
> s NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells
> NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sf
> ormat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAP0
> N(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBRE
> V ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels
> ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcell
> s EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cform
> at(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(pass
> thru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(pass
> thru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {

```

```

= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`nocl' `nopvalues'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - opts_exclusive "`versus' `compare'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'`opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "'`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'`opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'

```

```

= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"

```

```

    local sformat ``sformat(`sformat')''
  }
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`"pformat"') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `"pformat"'
  local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {

```



```

        - exit
    end opts_exclusive

>
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

>
    end _get_diopts.Wrapon

    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselev
> els' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflege
> nd' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fv
> wrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
    = c_local DUPS
    - if `K' == 2 {
    = if 2 == 2 {
        - c_local `c_opts' ``options'
    = c_local options `''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

>
    end _get_diopts.GetDiopts

    - c_local `name2' ``options'
    = c_local options `''
    - }
    - local DUPS : list DUPS - DEFAULT
    - gettoken DUPS : DUPS
    - if ``DUPS' != "" {
    = if `'' != "" {
        if (strpos(`DUPS', "(")) {
            gettoken DUPS : DUPS, parse("(")
            local DUPS `DUPS'()'
        }
    }

```

```

        di as err "duplicate {bf:`DUPS`} option not allowed"
        exit 198
    }
    - sreturn clear
    - foreach sname of local snames {
    - sreturn local `sname' = ``S_`sname'``
    = sreturn local level = ``95''
    - }
}

> _____ end _get_diopts _____

    - mlopts mlopts, `options'
    = mlopts mlopts,

> _____ begin mlopts _____

    - version 8.0
    - local maxoptlist MLOPTs(string) TRace TRACEDiffs GRADient HESSian
> SHOWSTEP ITERate(int -1) TOLerance(real -1) LTOLerance(real -1) GTOLerance(r
> eal -1) NONRTOLerance SHOWNRtolerance SHOWTOLerance NRTOLerance(real -1) QTOL
> Lerance(real -1) DIFFicult HALFSTEPonly CONSTraints(string) noCNSNOTes COLli
> near TECHnique(passthru) VCE(passthru) DOTS depsilon(passthru) SHOWH NOTCONC
> AVE(passthru) USEVIEWS NOLOg LOg nrderiv
        - syntax namelist(id="macro names" min=1 max=2) [, `maxoptlist' * ]
        = syntax namelist(id="macro names" min=1 max=2) [, MLOPTs(string) TR
> ace TRACEDiffs GRADient HESSian SHOWSTEP ITERate(int -1) TOLerance(real -1)
> LTOLerance(real -1) GTOLerance(real -1) NONRTOLerance SHOWNRtolerance SHOWTOL
> Lerance NRTOLerance(real -1) QTOLerance(real -1) DIFFicult HALFSTEPonly CONS
> Traints(string) noCNSNOTes COLLinear TECHnique(passthru) VCE(passthru) DOTS
> depsilon(passthru) SHOWH NOTCONCAVE(passthru) USEVIEWS NOLOg LOg nrderiv * ]
        - gettoken dest rest : namelist
        - if "`rest'" == "" {
        = if "" == "" {
        - syntax namelist [, `maxoptlist' ]
        = syntax namelist [, MLOPTs(string) TRace TRACEDiffs GRADient HESSia
> n SHOWSTEP ITERate(int -1) TOLerance(real -1) LTOLerance(real -1) GTOLerance
> (real -1) NONRTOLerance SHOWNRtolerance SHOWTOLerance NRTOLerance(real -1) Q
> TOLerance(real -1) DIFFicult HALFSTEPonly CONSTraints(string) noCNSNOTes COL
> linear TECHnique(passthru) VCE(passthru) DOTS depsilon(passthru) SHOWH NOTCO
> NCAVE(passthru) USEVIEWS NOLOg LOg nrderiv ]
        - }
        - local mlopts `mlopts' `trace' `tracediffs' `gradient' `hessian' `s
> howstep' `difficult' `halfsteponly' `nonrtolerance' `shownrtolerance' `showt
> olerance' `dots' `depsilon' `showh' `notconcave' `useviews' `log' `nolog' `n
> rderiv'
        = local mlopts
        - if `tolerance'!= -1 {
        = if -1!= -1 {

```

```

    local tol "tolerance(`tolerance')"
  }
- if `ltolerance' != -1 {
= if -1 != -1 {
    local ltol "ltolerance(`ltolerance')"
  }
- if `gtolerance' != -1 {
= if -1 != -1 {
    local gtol "gtolerance(`gtolerance')"
  }
- if `nrtolerance' != -1 {
= if -1 != -1 {
    local nrtol "nrtolerance(`nrtolerance')"
  }
- if `qtolerance' != -1 {
= if -1 != -1 {
    local qtol "qtolerance(`qtolerance')"
  }
- if `iterate' != -1 {
= if -1 != -1 {
    local iter "iterate(`iterate')"
  }
- sreturn clear
- if "`technique' `vce'" != "" {
= if "" != "" {
    ml_technique, `technique' `vce' nodefault
    if ``s(technique)'' != "" {
      local technique "technique(`s(technique)')"
    }
    if ``s(vce)'' != "" {
      local vce ``s(vceopt)''
    }
    sreturn local technique ``technique''
    sreturn local vce ``vce''
  }
- if "`constraints'" != "" {
= if "" != "" {
    capture confirm name `constraints'
    if _rc {
      CheckClist constraints , constraints(`constraints')
    }
    else {
      CheckCmat `constraints'
    }
    local const "constraints(`constraints')"
    sreturn local constraints ``constraints''
  }

```

```

    }
    - sreturn local collinear "`collinear'"
    = sreturn local collinear ""
    - c_local `dest' `mlopts' `tol' `ltol' `gtol' `nrtol' `iter' `const'
> `cnsnotes' `collinear' `technique' `vce' `qtol'
    = c_local mlopts
    - if "`rest'" != "" {
    = if "" != "" {
        c_local `rest' ``options''
    }
}



---


> 

---

 end mlopts 

---



- local coll `s(collinear)'
= local coll
- local cns `s(constraints)'
= local cns
- local mlopts `mlopts' `crittype'
= local mlopts
- if ``score'""!="" {
= if `""!="" {
    confirm new variable `score'
    local nword : word count `score'
    if `nword' > 1 {
di in red "score() must contain the name of only " "one new variable
> "

    exit 198
    }
    tempvar scvar
    local scopt score(`scvar')
    }
- if "`offset'""!="" & "`exposur'""!="" {
= if `""!="" & `""!="" {
    di in red "only one of offset() or exposure() can be specified"
    exit 198
    }
- if "`constan'""!="" {
= if `""!="" {
    local nvar : word count `varlist'
    if `nvar' == 1 {
di in red "independent variables required with " "noconstant option"
    exit 100
    }
    }
- marksample touse
- if ``cluster'""!="" {
= if `""!="" {

```

```

    markout `touse' `cluster', strok
    local clopt cluster(`cluster')
  }
- if "`exposur'"!="" {
= if ""!="" {
  capture assert `exposur' > 0 if `touse'
  if _rc {
    di in red "exposure() must be greater than zero"
    exit 459
  }
  tempvar offset
  qui gen double `offset' = ln(`exposur')
  local offvar "ln(`exposur')"
```

---

```

  }
- if "`offset'"!="" {
= if ""!="" {
  markout `touse' `offset'
  local offopt "offset(`offset')"
```

---

```

  if "`offvar'"="" {
    local offvar "`offset'"
  }
}
- gettoken y xvars : varlist
- _fv_check_depvar `y'
= _fv_check_depvar y_timing_max
- tsunab y : `y'
= tsunab y : y_timing_max

> ----- begin tsunab -----
- version 6
- gettoken user 0: 0, parse(" :")
- gettoken colon 0: 0, parse(" :")
- if ``colon' != ":" { error 198 }
= if ``:' != ":" { error 198 }
- syntax [varlist(ts default=empty)] [, MIN(integer 1) MAX(integer 3
> 2000) NAME(string)]
- c_local `user' `varlist'
= c_local y y_timing_max
- local n : word count `varlist'
= local n : word count y_timing_max
- if `n'>=`min' & `n'<=`max' { exit }
= if 1>=1 & 1<=32000 { exit }

> ----- end tsunab -----
- local yname : subinstr local y "." "_"
- if "`weight'"!="" {
```

```

= if ""!=" {
  if "`weight'"=="fweight" {
    local wt `[fw`exp']"
  }
  else if "`weight'"=="iweight" {
    local wt `[iw`exp']"
  }
  else local wt `[aw`exp']"
}
- summarize `y' `wt' if `touse', meanonly
= summarize y_timing_max if __000002, meanonly
- if r(N) == 0 { error 2000 }
- if r(N) == 1 { error 2001 }
- if r(min) < 0 {
  di in red "`y' must be greater than or equal to zero"
  exit 459
}
- if r(min) == r(max) & r(min) == 0 {
  di in red "`y' is zero for all observations"
  exit 498
}
- tempname mean nobs
- scalar `mean' = r(mean)
= scalar __000003 = r(mean)
- scalar `nobs' = r(N)
= scalar __000004 = r(N)
- if "`display'"==" {
= if "nodisplay"==" {
  capture assert `y' == int(`y') if `touse'
  if _rc {
    __New_Note `0'
  }
}
- if "`display'"==" & "`weight'"=="aweight" {
= if "nodisplay"==" & ""=="aweight" {
  di in blu "note: you are responsible for interpretation " "of analyt
> ic weights."
}
- if "`constan'`cns'"==" & "`cluster'"==" & "`weight'"!="pweight" {
= if ""==" & ""==" & ""!="pweight" {
- tempname c
- if "`offset'"!=" {
= if ""!=" {
  SolveC `y' `offset' [`weight'`exp'] if `touse', n(`nobs') mean(`mean
> ')
  scalar `c' = r(_cons)

```

```

        if `c'>=. {
            di in blu "note: exposure = exp(`offvar') overflows;" _n "      coul
> d not estimate constant-only model" _c
            if "`exposur'"==" " {
                di in blu _n "      use exposure() option for exposure = `offvar'."
            }
            else di in blu "."
            local cfail 1
        }
        local c "`c'+`offset'"
    }
    - else scalar `c' = ln(`mean')
    = else scalar __000005 = ln(__000003)
    - if "`cfail'"==" " {
    = if ""==" " {
        - LikePois `y' `c' [`weight'`exp'] if `touse', n(`nobs')
    = LikePois y_timing_max __000005 [] if __000002, n(__000004)
    }
}

> ----- begin poisson.LikePois -----
    - gettoken y 0 : 0
    - gettoken xb 0 : 0
    - syntax [fw aw pw iw] [if] , Nobs(string)
    - if "`weight'"!=" " {
    = if ""!=" " {
        if "`weight'"=="fweight" {
            local wt `[fw`exp']'
        }
        else if "`weight'"=="iweight" {
            local wt `[iw`exp']'
        }
        else local wt `[aw`exp']'
    }
    - tempvar lnf
    - qui gen double `lnf' = `y'*(`xb')-exp(`xb')-lngamma(`y'+1) `if'
    = qui gen double __000006 = y_timing_max*(__000005)-exp(__000005)-ln
> gamma(y_timing_max+1) if __000002
    - summarize `lnf' `wt' `if', meanonly
    = summarize __000006 if __000002, meanonly
    - if r(N) != `nobs' { exit }
    = if r(N) != __000004 { exit }
    - if "`weight'"=="aweight" {
    = if ""=="aweight" {
        ret scalar lnf = r(N)*r(mean)
    }
    - else ret scalar lnf = r(sum)
}

```

```

> ----- end poisson.LikePois -----
    - if r(lnf)<. {
    - local ll0 "lf0(1 `r(lnf)')"
```

= local ll0 "lf0(1 -437.385284075989)"

```

    - }
    - else {
      di in blu "note: could not compute " "constant-only model log likeli
> hood."
    }
    - }
    - }
    - if `fvops' {
    = if 1 {
    - local rmcoll "version 11: _rmcoll"
    - }
    - else local rmcoll _rmcoll
    - if "`display'"!=" & "`weight'"=="aweight" {
    = if "nodisplay"!=" & ""=="aweight" {
      `rmcoll' `xvars' [iw`exp'] if `touse', `constan' `coll'
    }
    - else `rmcoll' `xvars' [`weight'`exp'] if `touse', `constan' `coll'
    = else version 11: _rmcoll cb_authors_ext estimated wg_ndx_mult stky_
> pr_rottemberg wlth stky_pr_calvo ln_neq open [] if __000002,
    - local xvars `r(varlist)'
    = local xvars cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg w
> lth stky_pr_calvo ln_neq open
    - if "`from'"==" {
    = if ""==" {
    - `vv' Ipois `y' `xvars' [`weight'`exp'] if `touse', n(`nobs') mean(`m
> ean') `constan' `offopt'
    = version 18, missing: Ipois y_timing_max cb_authors_ext estimated wg_
> ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open [] if __000002, n(
> __000004) mean(__000003)

> ----- begin poisson.Ipois -----
    - local vv : display "version " string(_caller()) ", missing:"
    - version 6, missing
    - syntax varlist(fv ts) [fw aw pw iw/] [if] , Nobs(string) Mean(stri
> ng) [ noCONstant OFFset(string) ]
    - gettoken y xvars : varlist
    - tempvar xb z
    - tempname b1 b2 lnf1
    - if "`weight'"!=" {
    = if ""!=" {
      local awt `[aw=`exp']"'
      local wt `"(`exp')*"'

```



```

        local exp "`='exp'"
    }
    - quietly {
    - if "`offset'"!=" {
= if ""!=" {
    tempvar ynew
    gen double `ynew' = `y' - `offset'*`mean' `if'
    local poff "+`offset'"
    }
    - else local ynew `y'
= else local ynew y_timing_max
    - `vv' _regress `ynew' `xvars' `awt' `if', `constan'
    = version 18, missing: _regress y_timing_max cb_authors_ext estimat
> ed wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if __000002
> ,

> ----- begin _ereturn_display -----
    - version 11
    - if "`e(b)'" == "" & "`e(V)'" == "" {
= if "matrix" == "" & "matrix" == "" {
        exit
    }
    - if "`e(cmd)'" == "regress" {
= if "regress" == "regress" {
        - local ALTOPT Beta
        - }
        - syntax [, eform(passthru) First neq(integer -1) PPlus `ALTOPT' *]
= syntax [, eform(passthru) First neq(integer -1) PPlus Beta *]
        - _get_mcompare, `options'
= _get_mcompare, level(95.00) fwrap(1)

> ----- begin _get_mcompare -----
    - version 11
    - local cmd ``e(cmd)'"
= local cmd `"regress"'
    - syntax [name(name=caller)] [, *]
    - local is_pwc = inlist("pwcompare", "`cmd'", "`caller'") | inli
> st("pwmean", "`cmd'", "`caller'")
        = local is_pwc = inlist("pwcompare", "regress", "") | inlist("pw
> mean", "regress", "")
        - local is_marg = inlist("margins", "`cmd'", "`e(cmd2)'", "`call
> er'")
        = local is_marg = inlist("margins", "regress", "", "")
        - local OPTIONS NOADJust BONferroni DUNCan DUNNett SCHeffe SIDak
> SNK TUKey ADJustall
        - if `is_pwc' {

```

```

= if 0 {
  syntax [name] [, `OPTIONS' MCOMPare(string) *]
}
- else {
- syntax [name] [, MCOMPare(string) *]
- local suboption mcompare()
- }
- sreturn clear
- sreturn local options ``options'''
= sreturn local options ``level(95.00) fvwrap(1)'''
- if `is_pwc' {
= if 0 {
  local METHOD `noadjust' `bonferroni' `duncan' `dunnett' `schef
> fe' `sidak' `snk' `tukey'
  local METHOD : list uniq METHOD
  opts_exclusive ``METHOD'''
  local ALL `adjustall'
  opts_exclusive ``noadjust' `ALL'''
}
- local 0 `", `mcompare'''
= local 0 `", ""
- capture syntax [, `OPTIONS']
= capture syntax [, NOADJust BONferroni DUNCan DUNNett SCHeffe S
> IDak SNK TUKey ADJustall]
- if c(rc) {
  di as err "invalid mcompare() option;"
  di as err "method ``mcompare'' not allowed"
  error 198
}
- local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `dunnet
> t' `scheffe' `sidak' `snk' `tukey'
= local METHOD
- local METHOD : list uniq METHOD
- local ALL `ALL' `adjustall'
= local ALL
- local ALL : list uniq ALL
- opts_exclusive ``METHOD''' `suboption'
= opts_exclusive "" mcompare()

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {

```

```

= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- if "`METHOD'" == "" {
= if "" == "" {
- if "`ALL'" != "" {
= if "" != "" {
di as err "multiple comparison method required with the adjust
> all option"
    exit 198
}
- local METHOD noadjust
- local noadjust noadjust
- }
- opts_exclusive "`ALL' `noadjust'" `suboption'
= opts_exclusive " noadjust" mcompare()

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts noadjust
- local n 0
- while `""opts'" != "" {
= while `"noadjust'" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while `""opts'" != "" {
= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local EXCLUDE duncan dunnett snk tukey

```

```

- if `:list METHOD in EXCLUDE' {
= if 0 {
  if `is_pwc' {
    local opt : copy local METHOD
  }
  else {
    local opt "mcompare(`METHOD')"
  }
  local cmdlist anova manova regress mvreg pwmean
  if "`cmd'" == "pwcompare" {
    local cmd ``e(cmd_est)'''
  }
  if `is_marg' {
    local cmd margins
  }
  if !`:list cmd in cmdlist' | !`is_pwc' {
    di as err ``method `opt' is not allowed with results from `cmd
> ''

    exit 322
  }
  if ``e(prefix)''' == "svy" {
    di as err ``method `opt' is not allowed with results using the
> svy prefix''

    exit 322
  }
  local vce ``e(vce)'''
  if !inlist("`vce'", "", "ols") {
    di as err ``method `opt' is not allowed with results using vce
> (`vce')''

    exit 322
  }
}
- local EXCLUDE duncan dunnett snk tukey scheffe
- if `:list METHOD in EXCLUDE' {
= if 0 {
  if "`ALL'" != "" {
    di as err ``option adjustall is not allowed with method `METHO
> D'''

    exit 198
  }
}
- sreturn local method ``METHOD'''
= sreturn local method ``noadjust''
- sreturn local adjustall ``ALL'''
= sreturn local adjustall ``'''

```

---

```

> ----- end _get_mcompare -----
    - if "`s(method)'" != "" {
    = if "noadjust" != "" {
    - loca mcopt mcompare(`s(method)' `s(adjustall)')
    = loca mcopt mcompare(noadjust )
    - }
    - _get_diopts diopts, `s(options)'
    = _get_diopts diopts, level(95.00) fvwrap(1)

> ----- begin _get_diopts -----
    - version 11
    - syntax namelist(max=2) [, *]
    - gettoken name1 namelist : namelist
    - gettoken name2 namelist : namelist
    - GetDiopts diopts options, `options'
    = GetDiopts diopts options, level(95.00) fvwrap(1)

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"

```



```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {

```

```

= if 0 {
    confirm numeric format `cformat'
    if fmtwidth(`""cformat'") > 9 {
        local cformat "%9.0g"
        di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat `""cformat'""
    local cformat `cformat(`cformat')""
}
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth(`""sformat'") > 8 {
        local sformat "%8.2f"
        di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat `""sformat'""
    local sformat `sformat(`sformat')""
}
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`""pformat'") > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `""pformat'""
    local pformat `pformat(`pformat')""
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wraon , `fvwraon'
= Wraon ,

```

---

> \_\_\_\_\_ begin \_get\_diopts.Wraon \_\_\_\_\_

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
}

```

```

    }
    - opts_exclusive "`word' `width'" fvwrapon
    = opts_exclusive " " fvwrapon
  }
  > ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
      local ++n
      gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
      - exit
    }
  }
  > ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
      c_local fvwrapon fvwrapon(`fvwrapon')
    }
  }
  > ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "fvwrap(1)" != "" {
      - FVWrap, `fvwrap'
    = FVWrap, fvwrap(1)
  }
  > ----- begin _get_diopts.FVWrap -----
    - syntax, fvwrap(integer)
    - if `fvwrap' < 0 {
    = if 1 < 0 {
      di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "non
> negative integer required{p_end}"
      exit 198
    }
  }
  > ----- end _get_diopts.FVWrap -----
    - }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef

```

```

> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'
    = c_local diopts                                fvwrap(1)
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options ``''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
    _____
> _____ end _get_diopts.GetDiopts _____
    - c_local `name1' ``diopts''
    = c_local diopts ``fvwrap(1)''
    - local snames : s(macros)
    - foreach sname of local snames {
    - local S_`sname' = s(`sname')
    = local S_level = s(level)
    - }
    - local S_`sname' = s(`sname')
    = local S_method = s(method)
    - }
    - local S_`sname' = s(`sname')
    = local S_options = s(options)
    - }
    - sreturn clear
    - GetDiopts DEFAULT
    _____
> _____ begin _get_diopts.GetDiopts _____
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEle
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND c

```

```

> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'`` != "" {
    = while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`allbaselevels' `nbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'`` != "" {
    = while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`baselevels' `nbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```



```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding` `compare`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus` `compare`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {

```

```

= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `"cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `"sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `"pformat(`pformat')"'
  }

```

```

    }
    - if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
    }
    - if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
    - Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
    - capture syntax [, W0rd WIdth]
    - if c(rc) {
        di as err "invalid fvwrapon() option;"
        syntax [, W0rd WIdth]
        exit 198
    }
    - opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
= local fvwrapon
    - if `:length local fvwrapon' {
= if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----

```

```

- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'

= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options''
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- if ``name2'' == "" {
= if ``'' == "" {
- GetDiopts DUPS, `options'
= GetDiopts DUPS,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV

- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels''
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nbaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nbaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```



```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

```

```

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {

```

```

- exit


---


> 

---

 end opts_exclusive 

---


- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- opts_exclusive "`versus' `compare'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "


---



```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'
    = local omitted
    - local lstretch `lstretch' `nolstretch'
    = local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
    = local fvlabel
    - local abbrev `abbrev' `noabbrev'
    = local abbrev
    - if "`allbaselevels'`baselevels'" == "" {
    = if "" == "" {
    - if c(showbaselevels) == "all" {
        local allbaselevels allbaselevels
    }
    - else if c(showbaselevels) == "on" {
        local baselevels baselevels
    }
    - }
    - if "`emptycells'" == "" {
    = if "" == "" {
    - if c(showemptycells) == "off" {
        local emptycells noemptycells
    }
    - }
    - if "`omitted'" == "" {

```

```

= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `"cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `"sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `"pformat(`pformat')"'
  }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
  }
- if reldif(`level', c(level)) > 1e-3 {

```

```

= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wraapon , `fvwraapon'
= Wraapon ,

> ----- begin _get_diopts.Wraapon -----
- capture syntax [, WOrd Width]
- if c(rc) {
  di as err "invalid fvwraapon() option;"
  syntax [, WOrd Width]
  exit 198
}
- opts_exclusive "`word' `width'" fvwraapon
= opts_exclusive " " fvwraapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwraapon `word' `width'
= local fvwraapon
- if `:length local fvwraapon' {
= if 0 {
  c_local fvwraapon fvwraapon(`fvwraapon')
}

> ----- end _get_diopts.Wraapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef

```

```
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'
```

```
    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
      c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
```

---

```
> ----- end _get_diopts.GetDiopts -----
- }
- else {
  GetDiopts DUPS options, `options'
  c_local `name2' ``options''
}
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'' != "" {
= if ``' != "" {
  if (strpos(`DUPS', "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname''
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local method = `noadjust'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local options = `level(95.00) fvwrap(1)'
- }
```

---



```

> ----- end _get_diopts -----
    - if !':length local first' {
    = if !0 {
    - local first showeqns
    - }
    - if `neq' < 1 {
    = if -1 < 1 {
    - _ms_eq_info
    - local neqopt neq(`r(k_eq)')
    = local neqopt neq(1)
    - }
    - else {
      local neqopt neq(`neq')
    }
    - _coef_table, nodiparm nocnsreport noeqcheck eformall `eform' `fi
> rst' `neqopt' `plus' `beta' `mcopt' `diopts'
    = _coef_table, nodiparm nocnsreport noeqcheck eformall showeqns n
> eq(1) mcompare(noadjust ) fwrap(1)

> ----- begin _coef_table -----
    - version 11
    - local vv : di "version " string(max(11,_caller())) ", missing:
> "
    - if (!c(noisily) & c(coeftabresults) == "off") {
      exit
    }
    - _check_eclass
    - syntax [, BMATrix(passthru) VMATrix(passthru) * ]
    - if "`e(b)'" == "" & "`e(V)'" == "" & "`bmatrix'`vmatrix'"==""
> {
    = if "matrix" == "" & "matrix" == "" & ""=="" {
      exit
    }
    - local mc_cmds contrast margins pwcompare pwmean
    - local cmd "`e(cmd)'"
    = local cmd "regress"
    - local keepmc : list cmd in mc_cmds
    - if inlist("`cmd'", "pwcompare", "pwmean") {
    = if inlist("regress", "pwcompare", "pwmean") {
      local groups GROUPS
    }
    - if "`cmd'" == "regress" {
    = if "regress" == "regress" {
      - local beta Beta
      - }
    - if "`cmd'" == "gsem" {

```

```

= if "regress" == "gsem" {
    local fvignore fvignore(int 0)
    local flignore flignore
}
- if "`cmd'" == "sem" {
= if "regress" == "sem" {
    local standardized STANDARDIZED
    local showginvariant SHOWGINvariant
    local nolabel NOLABel LABel NOFVLABel FVLABel
    local nofootnote NOFOOTnote
    local wrap wrap(numlist max=1) fvwrap(passthru)
}
- if "`e(mi)'"=="mi" {
= if ""=="mi" {
    local dftable DFTable
    local dfonly DFONLY
    local noclustreport NOCLUSTReport
    local pisematrix PISEMATrix(string)
}
- syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st
> ring) DFMATrix(string) EQMATrix(string) ROWMATrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMATrix(string) MEMATrix(string) BSTDMATrix(string) CNSMATrix(string) P
> CLASSMATrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) `beta' `standardized' NOCI NOPValues `dftable' `dfonly' dfci DFPVal
> ues dfmissing `groups' COEFLegend selegend `fvignore' `flignore' `showginvar
> iant' `nolabel' `nofootnote' `wrap' sort `pisematrix' depname(string) COEFTi
> tle(string) coeftitle2(string) ptitle(string) cititle(string) NOMCLEgend noC
> NSReport FULLCNSReport `noclustreport' cformat(passthru) sformat(passthru) p
> format(passthru) NOFirst First SHOWEQns neq(integer -1) NODIPARM NOTEST SEP
> rator(integer 0) NOSKIP OFFSETONLY1 Plus NOEQCHECK EFORMALL CITYPE(string) e
> qselect(string) noEFORMNOTE * ]
= syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st
> ring) DFMATrix(string) EQMATrix(string) ROWMATrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMATrix(string) MEMATrix(string) BSTDMATrix(string) CNSMATrix(string) P
> CLASSMATrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) Beta NOCI NOPValues dfci DFPValues dfmissing COEFLegend selegen
> d sort depname(string) COEFTitle(string) coeftitle2(string) ptitle(st
> ring) cititle(string) NOMCLEgend noCNSReport FULLCNSReport cformat(passthru
> ) sformat(passthru) pformat(passthru) NOFirst First SHOWEQns neq(integer -1)
> NODIPARM NOTEST SEPARATOR(integer 0) NOSKIP OFFSETONLY1 Plus NOEQCHECK EFOR
> MALL CITYPE(string) eqselect(string) noEFORMNOTE * ]
- if "`wrap'" != "" {
= if "" != "" {
    opts_exclusive `wrap(`wrap') `fvwrap'

```

```

    local fvwrap fvwrap(`wrap')
  }
- if "`nolabel'`label'" != "" {
= if "" != "" {
  opts_exclusive "`nolabel' `label'"
  opts_exclusive "`nofvlabel' `fvlabel'"
  opts_exclusive "`nolabel' `fvlabel'"
  opts_exclusive "`nofvlabel' `label'"
  if "`nolabel'" != "" {
    local fvlabel nofvlabel
  }
  else {
    local fvlabel fvlabel
  }
}
- else local fvlabel `nofvlabel' `fvlabel'
= else local fvlabel
- local type `coeflegend' `selegend'
= local type
- opts_exclusive "`type' `standardized'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- if (`rowmatrix'=="" & "`norowci'!="") {
= if (`"'=="" & `"'!="") {
  di as err "{bf:norowci} requires {bf:rowmatrix()}"
  exit 198
}
- if (`dftable'!="" & "`norowci'!="") {
= if (`"'!="" & `"'!="") {
  di as err "{bf:norowci} is not allowed with {bf:`dftable'}"

```

```

        exit 198
    }
    - if ("`dfonly`"!=" & "`norowci`"!=") {
= if ("!=" & "!=") {
    di as err "{bf:norowci} is not allowed with {bf:`dfonly`}"
    exit 198
}
    - if ("`noci`"!=" & "`norowci`"!=") {
= if ("!=" & "!=") {
    di as err "{bf:norowci} is not allowed with {bf:`noci`}"
    exit 198
}
    - if "`standardized`" != "" & "`bstdmatrix`" != "" {
= if "" != "" & "" != "" {
    local beta beta
    local standardized
}
    - if `:length local type' == 0 {
= if 0 == 0 {
    - local type `beta' `noci' `nopvalues' `dftable' `dfonly' `group
> s' `dfci' `dfpvalues'
    = local type
    - }
    - opts_exclusive "`type'"
= opts_exclusive ""

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while "`opts'" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`first' `nofirst'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`first' `showeqns'"
= opts_exclusive " showeqns"

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts showeqns
- local n 0
- while `"'opts'"' != "" {
= while `"showeqns"' != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while `"'opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 1 < 2 {
- exit

> ----- end opts_exclusive -----
- local cnsreport `cnsreport' `fullcnsreport'
= local cnsreport nocnsreport
- opts_exclusive "`cnsreport'"
= opts_exclusive "nocnsreport"

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts nocnsreport
- local n 0
- while ```opts'`` != "" {
= while `"nocnsreport" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ```opts'`` != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
  }
- if `n' < 2 {
= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----

- if ```showeqns'`` != "" {
= if `"showeqns" != "" {
- local nofirst nofirst
- }
- if ```ematrix'`noeqcheck'`` == "" {
= if `"noeqcheck" == "" {
    if ```e(error)'`` == "matrix" {
      local ematrix e(error)
    }
  }
- _get_mcompare, `options'
= _get_mcompare, mcompare(noadjust ) fvwrap(1)
}

> ----- begin _get_mcompare -----

- version 11
- local cmd ```e(cmd)'``
= local cmd `"regress"
- syntax [name(name=caller)] [, *]
- local is_pwc = inlist("pwcompare", ``cmd'', ``caller'') | in
> list("pwmean", ``cmd'', ``caller'')
= local is_pwc = inlist("pwcompare", "regress", "") | inlist("
> pwmean", "regress", "")
- local is_marg = inlist("margins", ``cmd'', ``e(cmd2)'', ``ca
> ller'')
= local is_marg = inlist("margins", "regress", "", "")

```

```

- local OPTIONS NOADJust BONferroni DUNCan DUNNett SCHeffe SID
> ak SNK TUKey ADJustall
- if `is_pwc' {
= if 0 {
  syntax [name] [, `OPTIONS' MCOMPare(string) *]
}
- else {
- syntax [name] [, MCOMPare(string) *]
- local suboption mcompare()
- }
- sreturn clear
- sreturn local options ``options'''
= sreturn local options ``fvwrap(1)'''
- if `is_pwc' {
= if 0 {
  local METHOD `noadjust' `bonferroni' `duncan' `dunnett' `sch
> effe' `sidak' `snk' `tukey'
  local METHOD : list uniq METHOD
  opts_exclusive ``METHOD'''
  local ALL `adjustall'
  opts_exclusive ``noadjust' `ALL'''
}
- local 0 `", `mcompare'''
= local 0 `", noadjust'''
- capture syntax [, `OPTIONS']
= capture syntax [, NOADJust BONferroni DUNCan DUNNett SCHeffe
> SIDak SNK TUKey ADJustall]
- if c(rc) {
  di as err "invalid mcompare() option;"
  di as err "method ``mcompare'' not allowed"
  error 198
}
- local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `dunn
> ett' `scheffe' `sidak' `snk' `tukey'
= local METHOD noadjust
- local METHOD : list uniq METHOD
- local ALL `ALL' `adjustall'
= local ALL
- local ALL : list uniq ALL
- opts_exclusive ``METHOD''' `suboption'
= opts_exclusive "noadjust" mcompare()

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts noadjust
- local n 0
- while ``opts'' != "" {
= while "noadjust" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
  }
- if `n' < 2 {
= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----

- if "METHOD" == "" {
= if "noadjust" == "" {
    if "ALL" != "" {
di as err "multiple comparison method required with the adju
> stall option"
    exit 198
    }
    local METHOD noadjust
    local noadjust noadjust
    }
- opts_exclusive "ALL" `noadjust' `suboption'
= opts_exclusive " noadjust" mcompare()
}

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts noadjust
- local n 0
- while ``opts'' != "" {
= while "noadjust" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'' != "" {
= while `'' != "" {
    local ++n

```



```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 1 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - local EXCLUDE duncan dunnett snk tukey
    - if `:list METHOD in EXCLUDE' {
    = if 0 {
        if `is_pwc' {
            local opt : copy local METHOD
        }
        else {
            local opt "mcompare(`METHOD')"
        }
        local cmdlist anova manova regress mvreg pwmean
        if "`cmd'" == "pwcompare" {
            local cmd ``e(cmd_est)'''
        }
        if `is_marg' {
            local cmd margins
        }
        if !`:list cmd in cmdlist' | !`is_pwc' {
            di as err ``method `opt' is not allowed with results from `c
> md'''
            exit 322
        }
        if ``e(prefix)''' == "svy" {
            di as err ``method `opt' is not allowed with results using t
> he svy prefix''
            exit 322
        }
        local vce ``e(vce)'''
        if !inlist("`vce'", "", "ols") {
            di as err ``method `opt' is not allowed with results using v
> ce(`vce')'''
            exit 322
        }
    }
    - local EXCLUDE duncan dunnett snk tukey scheffe
    - if `:list METHOD in EXCLUDE' {
    = if 0 {
        if "`ALL'" != "" {
            di as err ``option adjustall is not allowed with method `MET
> HOD'''

```

```

        exit 198
    }
}
- sreturn local method ``METHOD'''
= sreturn local method "noadjust"
- sreturn local adjustall ``ALL'''
= sreturn local adjustall ''''

> ----- end _get_mcompare -----

- local method ``s(method)'''
= local method "noadjust"
- local all ``s(adjustall)'''
= local all ''''
- local options ``s(options)'''
= local options "fvwrap(1)"
- if "method" != "noadjust" {
= if "noadjust" != "noadjust" {
    local keepmc 1
}
- opts_exclusive "all" groups"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while ``''' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----

- if "method" == "dunnett" {
= if "noadjust" == "dunnett" {
    opts_exclusive "method" groups"
}
- _get_diopts diopts options, `options' `fvwrap' `fvlabel'
= _get_diopts diopts options, fvwrap(1)

> ----- begin _get_diopts -----

```

```

- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options, fvwrap(1)

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``baselevels' `nobaselevels'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``emptycells' `noemptycells'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells

```



```

- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {

```

```

= if 0 {
    confirm numeric format `sformat'
    if fmtwidth(`""sformat'") > 8 {
        local sformat "%8.2f"
        di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat `""sformat'""
    local sformat `sformat(`sformat')""
}
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`""pformat'") > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `""pformat'""
    local pformat `pformat(`pformat')""
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, W0rd WIdth]
    exit 198
}
- opts_exclusive ""`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

> ----- end _get_diopts.Wrapon -----

- if "`fvwrap'" != "" {
= if "fvwrap(1)" != "" {
- FVWrap, `fvwrap'
= FVWrap, fvwrap(1)

> ----- begin _get_diopts.FVWrap -----

- syntax, fvwrap(integer)
- if `fvwrap' < 0 {
= if 1 < 0 {
    di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n
> onnegative integer required{p_end}"
    exit 198
}

> ----- end _get_diopts.FVWrap -----

- }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local diopts          fvwrap(1)
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options''
= c_local options `""'
- }
- sreturn local coding `coding'

```

```

= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts'''
= c_local diopts ``fvwrap(1)''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- local S_`sname' = s(`sname')
= local S_options = s(options)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels''
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - local K : list sizeof namelist

```



```

- gettoken c_diopts c_opts : namelist
- opts_exclusive ""`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while "" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while "" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if ``allbaselevels'`baselevels' == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
    }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
    }
- }
- if ``emptycells' == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
    }
- }
- if ``omitted' == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
    }
}

```

```

- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wraon , `fvwraon'

```

```

= Wrapon ,


---


> 

---

 begin _get_diopts.Wrapon 

---


- capture syntax [, WOrd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}


---


> 

---

 end _get_diopts.Wrapon 

---


- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DEFAULT

```

```

- if `K' == 2 {
= if 1 == 2 {
  c_local `c_opts' ``options'
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- if ``name2' == "" {
= if "options" == "" {
  GetDiopts DUPS, `options'
}
- else {
- GetDiopts DUPS options, `options'
= GetDiopts DUPS options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels` `nobaselevels`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels` `nobaselevels`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive ""emptycells' `noemptycells'""
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive ""fvlabel' `nofvlabel'""
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive ""omitted' `noomitted'""
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```



```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {

```

```

- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> \_\_\_\_\_ begin \_get\_diopts.Wrapon \_\_\_\_\_

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}

```

---

```

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DUPS
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options'

```

```

= c_local options `""'
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- c_local `name2' ``options'""
= c_local options `""'
- }
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'"" != "" {
= if `""' != "" {
  if (strpos(`DUPS'"" , "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'""
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'""
= sreturn local level = `95'""
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local method = `noadjust'""
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local options = `fvwrap(1)""
- }

```

---

```

> ----- end _get_diopts -----
- local legend coeflegend selegend
- local legend : list legend & diopts
- local diopts : list diopts - legend
- local lstretch lstretch nolstretch
- local lstretch : list lstretch & diopts
- local diopts : list diopts - lstretch
- local markdown markdown

```

```

- local markdown : list markdown & diopts
- local diopts : list diopts - markdown
- _get_diopts ignore, `diopts' `cformat' `sformat' `pformat'
= _get_diopts ignore, fvwrap(1)

```

---

```

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options, fvwrap(1)

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----

```

```

- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```



```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K :list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

```

```

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
    }
  }
}

```

```

di "{txt}note: invalid cformat(), using default"
}
sreturn local cformat ``cformat'''
local cformat `cformat(`cformat')''
}
- if `:length local sformat' {
= if 0 {
confirm numeric format `sformat'
if fmtwidth(`sformat') > 8 {
local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat ``sformat'''
local sformat `sformat(`sformat')''
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth(`pformat') > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat ``pformat'''
local pformat `pformat(`pformat')''
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, W0rd WIdth]
exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "fvwrap(1)" != "" {
    - FVWrap, `fvwrap'
    = FVWrap, fvwrap(1)

> ----- begin _get_diopts.FVWrap -----
    - syntax, fvwrap(integer)
    - if `fvwrap' < 0 {
    = if 1 < 0 {
        di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n
> onnegative integer required{p_end}"
        exit 198
    }

> ----- end _get_diopts.FVWrap -----
    - }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abb
> rev'
    = c_local diopts          fvwrap(1)

```

```

- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options''
= c_local options ``''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95
}

> ----- end _get_diopts.GetDiopts -----

- c_local `name1' ``diopts''
= c_local ignore ``fvwrap(1)''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_options = s(options)
- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- sreturn clear
- GetDiopts DEFAULT
}

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels''

```



```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive ``allbaselevels' `nbaselevels'
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive ``baselevels' `nbaselevels'
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``coeflegend' `selegend'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``noci' `nopvalues'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"

```

```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'
    = local omitted
    - local lstretch `lstretch' `nolstretch'
    = local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
    = local fvlabel
    - local abbrev `abbrev' `noabbrev'
    = local abbrev
    - if ``allbaselevels'`baselevels' == "" {
    = if "" == "" {
    - if c(showbaselevels) == "all" {
        local allbaselevels allbaselevels
    }
    - else if c(showbaselevels) == "on" {
        local baselevels baselevels
    }
    - }
    - if ``emptycells' == "" {
    = if "" == "" {
    - if c(showemptycells) == "off" {
        local emptycells noemptycells
    }

```

```

- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `pformat(`pformat')'
  }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']

```

```

    }
    - if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
    - Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
    - capture syntax [, WOrd WIdth]
    - if c(rc) {
        di as err "invalid fvwrapon() option;"
        syntax [, WOrd WIdth]
        exit 198
    }
    - opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
= local fvwrapon
    - if `:length local fvwrapon' {
= if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
= if "" != "" {
        FVWrap, `fvwrap'
    }

```



```

        - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

        = c_local DEFAULT
        - if `K' == 2 {
        = if 1 == 2 {
            c_local `c_opts' ``options''
        }
        - sreturn local coding `coding'
        = sreturn local coding
        - sreturn local versus `versus'
        = sreturn local versus
        - sreturn local compare `compare'
        = sreturn local compare
        - sreturn local level `level'
        = sreturn local level 95
    _____
> _____ end _get_diopts.GetDiopts _____
        - if ``name2'' == "" {
        = if ``'' == "" {
        - GetDiopts DUPS, `options'
        = GetDiopts DUPS,
    _____
> _____ begin _get_diopts.GetDiopts _____
        - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITTED OMITTED NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(strin
> g) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
        - syntax namelist(max=2) [, `DIOPTS' *]
        = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITTED OMITTED NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
        - opts_exclusive ``allbaselevels' `noallbaselevels''
        = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
        - version 8.2
        - args opts optname errcode optslab
        - local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - local K : list sizeof namelist

```

```

- gettoken c_diopts c_opts : namelist
- opts_exclusive ""coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while "" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""`opts'" != "" {
= while "" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ""abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if ``allbaselevels'`baselevels' == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
    }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
    }
- }
- if ``emptycells' == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
    }
- }
- if ``omitted' == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
    }
}

```

```

- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wraon , `fvwraon'

```



```

= Wrapon ,


---


> 

---

 begin _get_diopts.Wrapon 

---


- capture syntax [, WOrd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}


---


> 

---

 end _get_diopts.Wrapon 

---


- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DUPS

```

```

- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options'
  }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts -----
- }
- else {
  GetDiopts DUPS options, `options'
  c_local `name2' ``options'
}
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS' != "" {
= if `"' != "" {
  if (strpos(`DUPS', "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname'
= sreturn local options = `fvwrap(1)'
- }
- sreturn local `sname' = ``S_`sname'
= sreturn local method = `noadjust'
- }

```

---

```

> ----- end _get_diopts -----
- local cformat ``s(cformat)'
= local cformat `"'
- local sformat ``s(sformat)'

```

```

= local sformat `""'
- local pformat ``s(pformat)''
= local pformat `""'
- _get_eformopts , eformopts(`options') allowed(__all__) soption
> s
= _get_eformopts , eformopts() allowed(__all__) soptions
_____
> _____ begin _get_eformopts _____
- version 8.0
- syntax [, eformopts(string asis) soptions ALLOWed(string) ]
- local 0 , `eformopts'
= local 0 ,
- local EFALL NOHR hr NOSHR shr or IRr RRr tr TRatio
- if "`soptions'" != "" {
= if "soptions" != "" {
- syntax [, EForm1(passthru) EForm `EFALL' * ]
= syntax [, EForm1(passthru) EForm NOHR hr NOSHR shr or IRr RR
> r tr TRatio * ]
- local opts ``options''
= local opts `""'
- local 0 , `eform1' `eform' `nohr' `hr' `noshr' `shr' `or' `i
> rr' `rrr' `tr' `tratio'
= local 0 ,
- }
- if "`allowed'" == "__all__" {
= if "__all__" == "__all__" {
- local allowed `EFALL'
= local allowed NOHR hr NOSHR shr or IRr RRr tr TRatio
- }
- foreach ef of local allowed {
- capture confirm name `ef'
= capture confirm name NOHR
- if _rc {
di as err "`ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr
- }
- capture confirm name `ef'
= capture confirm name hr
- if _rc {
di as err "`ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'

```

```

= local efopts nohr hr
- }
- capture confirm name `ef'
= capture confirm name NOSHr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr
- }
- capture confirm name `ef'
= capture confirm name shr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr
- }
- capture confirm name `ef'
= capture confirm name or
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or
- }
- capture confirm name `ef'
= capture confirm name IRr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr
- }
- capture confirm name `ef'
= capture confirm name RRr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr

```



```

= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- if ```eform1'``'=="Odds ratio" {
= if ``'``'=="Odds ratio" {
    local eform_cons_ti = "Odds"
    local eform_cons_note = "odds ratios"
}
- else if ```eform1'``'=="Haz. ratio" {
= else if ``'``'=="Haz. ratio" {
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
}
- else if ```eform1'``'=="Time ratio" {
= else if ``'``'=="Time ratio" {
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
}
- else if ```eform1'``'=="IRR" {
= else if ``'``'=="IRR" {
    local eform_cons_ti = "Inc. rate"
    local eform_cons_note = "incidence-rate ratios"
    local eform_cons_ti2 = "incidence rate"
}
- else if ```eform1'``'=="Risk ratio" {
= else if ``'``'=="Risk ratio" {
    local eform_cons_ti = "Risk"
    local eform_cons_note = "risk ratios"
}
- else if ```eform1'``'=="Hlth ratio" {
= else if ``'``'=="Hlth ratio" {
    local eform_cons_ti = "Health"
    local eform_cons_note = "health ratios"
}
- local k : list sizeof eform
- if `k' {
= if 0 {
    opts_exclusive "`eform'"
    if `:length local eform1' {
    opts_exclusive "eform() `eform'"
    }
}
- if `k' {
= if 0 {

```

```

if ("`eform`"=="eform") local eform1 = "exp(b)"
else if ("`eform`"=="hr") {
  local eform1 = "Haz. ratio"
  local eform_cons_ti = "Hazard"
  local eform_cons_note = "hazard ratios"
}
else if ("`eform`"=="shr") {
  local eform1 = "SHR"
  local eform_cons_ti = "Subhazard"
  local eform_cons_note = "subhazard ratios"
}
else if ("`eform`"=="tr") {
  local eform1 = "Time ratio"
  local eform_cons_ti = "Time"
  local eform_cons_note = "time ratios"
}
else if ("`eform`"=="tratio") {
  local eform1 = "Time ratio"
  local eform_cons_ti = "Time"
  local eform_cons_note = "time ratios"
}
else if ("`eform`"=="or") {
  local eform1 = "Odds ratio"
  if ("`e(cmd)`"=="asclogit" | "`e(cmd)`"=="cmclogit" | "`e(cm
> d)`"=="cmmixlogit" | "`e(cmd)`"=="cmxtmixlogit" ) {
    local eform_cons_ti = "Rel. risk"
    local eform_cons_note = "relative-risk ratios"
    local eform_cons_ti2 = "relative risk"
  }
  else {
    local eform_cons_ti = "Odds"
    local eform_cons_note = "odds ratios"
  }
}
else if ("`eform`"=="irr") {
  local eform1 = "IRR"
  local eform_cons_ti = "Inc. rate"
  local eform_cons_note = "incidence-rate ratios"
  local eform_cons_ti2 = "incidence rate"
}
else if ("`eform`"=="rrr") {
  local eform1 = "RRR"
  local eform_cons_ti = "Rel. risk"
  local eform_cons_note = "relative-risk ratios"
  local eform_cons_ti2 = "relative risk"
}

```

```

else if ("`eform`"!="nohr") & ("`eform`"!="noshr") {
  local eform1 = upper("`eform`")
}
}
- sreturn clear
- sreturn local options ``opts``
= sreturn local options ````
- sreturn local opt `eform`
= sreturn local opt
- sreturn local str `eform1`
= sreturn local str
- if ``eform1`` != "" {
= if ```` != "" {
  sreturn local eform eform(`eform1`)
}
- else sreturn local eform ""
- sreturn local eform_cons_ti `eform_cons_ti`
= sreturn local eform_cons_ti
- sreturn local eform_cons_note `eform_cons_note`
= sreturn local eform_cons_note
- sreturn local eform_cons_ti2 `eform_cons_ti2`
= sreturn local eform_cons_ti2
}

> ----- end _get_eformopts -----
- local tr_item = ``s(eform_cons_note)``
= local tr_item = ````
- if inlist("`bmatrix`", "", "e(b)") & "`e(k_eform)`" == "0" {
= if inlist("", "", "e(b)") & "" == "0" {
  local eform
}
- else local eform ``s(str)``
= else local eform ````
- local eform_cons_ti ``s(eform_cons_ti)``
= local eform_cons_ti ````
- local eform_cons_ti2 ``s(eform_cons_ti2)``
= local eform_cons_ti2 ````
- if ("`e(consonly)`"!="1" | ``eform_cons_ti``=="") {
= if ("!="1" | ``==") {
- local eformdi ``s(str)``
= local eformdi ````
- }
- else {
  local eformdi ``eform_cons_ti``
  local tr_item = cond(missing("`eform_cons_ti2`"), strlower(`eform_cons_ti`), ``eform_cons_ti2``)
}
}

```



```

- local coefttl = cond(`""`eform'""==`""', `""`coef title'""', `""`ef
> ormdi'""')
= local coefttl = cond(`""'==`""', `""', `""')
- local options `""`s(options)'""
= local options `""'
- _get_diparmopts, diparmopts(`options') level(`level')
= _get_diparmopts, diparmopts() level(95)


---


> 

---

 begin _get_diparmopts 

---


- if _caller() < 8.2 {
  _get_diparmopts_8 `0'
  exit
}
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom pl
> us SYNTAXonly * ]
- local goptions `options'
= local goptions level(95)
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk , level(95)


---


> 

---

 begin _get_diparmopts.SpliceGlobals 

---


- syntax name(id="macro name" name=c_diparm) [, diparm(strin
> g asis) Level(cilevel) dof(passthru) syntaxonly ]
- if `""`dof'"" == "" & `""`e(df_r)'"" != "" {
= if `""' == "" & `212' != "" {
- capture confirm integer number `e(df_r)'
= capture confirm integer number 212
- if !c(rc) {
- local gdof dof(`e(df_r)')
= local gdof dof(212)
- }
- capture
- }
- else local gdof `""`dof'""
= else local gdof `""'
- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)
- if (`""`diparm'"" == "") exit
= if (`""' == "") exit


---


> 

---

 end _get_diparmopts.SpliceGlobals 

---


- local 0 `", `diparmopts'""

```

```

= local 0 `", ""
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while ``diparm''' != "" {
= while `''' != "" {
    local ++k
    local single 1
    local after
    capture _on_colon_parse `diparm'
    if !c(rc) {
        local single 0
        local diparm ``s(before)'''
        local after ``s(after)'''
    }
    capture
    SpliceGlobals diparm , diparm(`diparm') `syntaxonly' `goptio

> ns'

    local diparm`k' ``diparm'''
    local diparmopts ``diparmopts' : `diparm'''
    if `single' | ``:list retok after''' == "" {
        local 0 `", `options'''
        syntax [, diparm(string asis) * ]
    }
    else {
        local diparm ``after'''
    }
}

- if ``soptions''' == "" {
= if `''' == "" {
- syntax [, diparm(string asis) ]
- }
- if ("`execute'" == "") {
= if ("" == "") {
- sreturn clear
- sreturn local options ``options'''
= sreturn local options `'''
- sreturn local diparm ``diparmopts'''
= sreturn local diparm `'''
- forval i = 1/`k' {
= forval i = 1/0 {
    sreturn local diparm`i' ``diparm`i'''''
}
- sreturn local k `k'
= sreturn local k 0
- exit

```

---

```

> end _get_diparmopts
- local NODIPARM : length local nodiparm
- if `NODIPARM' {
= if 8 {
- local options
- }
- local GTOPTS parse(":") bind quotes
- local k 0
- if !`NODIPARM' {
= if !8 {
  if ``e(diparm)'' != "" {
    local i 0
    local diparm ``e(diparm)''
  }
  else if ``e(diparm1)'' != "" {
    local i 1
    local diparm ``e(diparm1)''
  }
  while `:length local diparm' {
    gettoken diparm rest : diparm, `GTOPTS'
    while `:length local rest' {
      local ++k
      local diparm`k' : copy local diparm
      gettoken COLON rest : rest, `GTOPTS'
      gettoken diparm rest : rest, `GTOPTS'
    }
    local ++k
    local diparm`k' : copy local diparm
    local ++i
    local diparm ``e(diparm`i)''
  }
}
- if `:length local options' {
= if 0 {
  local 0 `", `options''
  syntax [, diparm(string asis) *]
  if `k' & `:length local diparm' {
    local ++k
    local diparm`k' __sep__
  }
  while `:length local diparm' {
    gettoken diparm rest : diparm, `GTOPTS'
    while `:length local rest' {
      local ++k
      local diparm`k' : copy local diparm

```

```

    gettoken COLON rest : rest, `GT_OPTS'
    gettoken diparm rest : rest, `GT_OPTS'
  }
  local ++k
  local diparm`k' : copy local diparm
  local 0 `", `options'""
  syntax [, diparm(string asis) *]
}
}
- if "`markdown'" != "" {
= if "" != "" {
  local lsize = c(linesize)
  set linesize 255
}
- `vv' mata: _coef_table()
= version 11, missing: mata: _coef_table()
}

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options, fwrap(1)
}

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "
}

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - local K : list sizeof namelist

```



```

- gettoken c_diopts c_opts : namelist
- opts_exclusive ""coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""opts'" != "" {
= while ""'" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive ""versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ""opts'" != "" {
= while ""'" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive ""abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if ``allbaselevels'`baselevels' == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
    }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
    }
- }
- if ``emptycells' == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
    }
- }
- if ``omitted' == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
    }
}

```

```

- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wraon , `fvwraon'

```

```

= Wrapon ,


---


> 

---

 begin _get_diopts.Wrapon 

---


- capture syntax [, WOrd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}


---


> 

---

 end _get_diopts.Wrapon 

---


- if "`fvwrap'" != "" {
= if "fvwrap(1)" != "" {
- FVWrap, `fvwrap'
= FVWrap, fvwrap(1)


---


> 

---

 begin _get_diopts.FVWrap 

---


- syntax, fvwrap(integer)
- if `fvwrap' < 0 {
= if 1 < 0 {
  di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n

```

```

> onnegative integer required{p_end}"
    exit 198
  }
}

> ----- end _get_diopts.FVWrap -----
- }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local diopts          fvwrap(1)
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options ``''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
}

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts''
= c_local __00000B ``fvwrap(1)''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_k = s(k)
- }
- sreturn clear
- GetDiopts DEFAULT
}

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N

```

```

> OABBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----

```

```

- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
  }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
  }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```



```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
= local baselevels
    - local emptycells `emptycells' `noemptycells'
= local emptycells
    - local omitted `omitted' `noomitted'
= local omitted
    - local lstretch `lstretch' `nolstretch'
= local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
    - local abbrev `abbrev' `noabbrev'
= local abbrev
    - if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
        - if c(showbaselevels) == "all" {
            local allbaselevels allbaselevels
        }
    }

```

```

- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {

```

```

        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat ``pformat''
    local pformat `pformat(pformat)''
    }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
    }
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
    }
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {

```

```

        = if 0 {
            c_local fvwrapon fvwrapon(`fvwrapon')
        }
    }
}

> ----- end _get_diopts.Wrapon -----

- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options'
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95
}

> ----- end _get_diopts.GetDiopts -----

- if ``name2' == "" {
= if `'' == "" {
- GetDiopts DUPS, `options'
= GetDiopts DUPS,

}

> ----- begin _get_diopts.GetDiopts -----

- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV

- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA

```

```

> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPVALUES MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *)
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`baselevels' `nobaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```



```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {

```

```

- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')"'
}

```

```

- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {

```

```

        = if "" != "" {
            FVWrap, `fvwrap'
        }
        - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

        = c_local DUPS
        - if `K' == 2 {
        = if 1 == 2 {
            c_local `c_opts' ``options''
        }
        - sreturn local coding `coding'
        = sreturn local coding
        - sreturn local versus `versus'
        = sreturn local versus
        - sreturn local compare `compare'
        = sreturn local compare
        - sreturn local level `level'
        = sreturn local level 95
    }
}

> ----- end _get_diopts.GetDiopts -----

- }
- else {
    GetDiopts DUPS options, `options'
    c_local `name2' ``options''
}
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'' != "" {
= if `'' != "" {
    if (strpos(`DUPS'', "(")) {
        gettoken DUPS : DUPS, parse("(")
        local DUPS `DUPS'()'
    }
    di as err "duplicate {bf:`DUPS'} option not allowed"
    exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname''
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local k = `0'

```

```

- }


---


> 

---

 end _get_diopts
- if "`markdown'" != "" {
= if "" != "" {
    set linesize `lsize'
  }
- if "`eqselect'" == "" & "`eformnote'" != "noeformnote" {
= if "" == "" & "" != "noeformnote" {
- if ``"eformdi" != "" & ("`e(cmd)'" != "gnbreg" | "`e(prefix)'" ==
> "svy") {
= if ``" != "" & ("regress" != "gnbreg" | "" == "svy") {
    local k_eform `r(k_eform)'
    if ("`k_eform'" == "") {
      local k_eform = 1
    }
    local k_eq `e(k_eq)'
    if ("`k_eq'" == "") {
      local k_eq = 1
    }
    if (`k_eform' < `k_eq' & `k_eform') {
      if ("`e(cmd2)'" == "stintreg" | "`e(cmd2)'" == "streg") & ("`e
> (cmd)'" == "weibull" | "`e(cmd)'" == "gompertz") {
      }
      else {
        _eform_multeq_note efnote : `k_eform' ``tr_item'
        di as txt "`efnote'"
      }
    }
  }
- if ("`"eformdi" != "" & "`e(noconstant)'" == "0" & ``"eform_cons
> _ti" != "" & "`e(consonly)'" != "1") {
= if (`" != "" & "" == "0" & ``" != "" & "" != "1") {
    if ("`e(cmd)'" == "asclogit" | "`e(cmd)'" == "cmclogit" | "`e(cmd)
> "" == "cmmixlogit" | "`e(cmd)'" == "cmxtmixlogit" ) {
    local asvar = ``"e(ranvars)'"`e(fixvars)'"`e(indvars)'" != ""
    if ``"e(casevars)'" != "" & `asvar' {
      local note1 "Exponentiated coefficients represent odds ratios
> for "
      local note2 "alternative-specific variables (first equation) a
> nd "
      local note3 "{help j_cmrisk##|_new:relative-risk ratios} "
      local note4 "for case-specific variables."
      di as txt "{p 0 6 2}Note: `note1'`note2'`note3'`note4'`p_end'"
    }
  }
}

```



```

if "`eform_cons_ti'" == "Inc. rate" {
  local eform_cons_note "incidence rate"
}
else if "`eform_cons_ti'" == "Rel. risk" {
  local eform_cons_note "relative risk"
}
else if "`eform_cons_ti'" == "Health" {
  local eform_cons_note "health (probability of no disease)"
}
else {
  local eform_cons_note = strlower("`eform_cons_ti'")
}
local eform_cons_note "baseline `eform_cons_note'"
if ("`e(cmd2)'"=="") {
  local cmd `e(cmd)'
}
else {
  local cmd `e(cmd2)'
}
local f2 = udsustr("`cmd'",1,2)
local is_re = ("`f2'"=="xt" | "`f2'"=="me")
if `is_re' {
  local extranote " (conditional on zero random effects)"
}
di as txt "{p 0 6 2}Note: {bf:_cons} estimates " "`eform_cons_
> note`extranote'" _c
if ("`e(cmd)'"=="mlogit" | "`e(cmd)'"=="asclogit" | "`e(cmd)'"
> == "cmclogit" | "`e(cmd)'"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
  di as txt " for each outcome" _c
}
di as txt ".{p_end}"
}
- }
- return add
- if !`keepmc' {
= if !0 {
- return local mcmethod
- return local mctitle
- }

_____
> _____ end _coef_table _____
_____
> _____ end _ereturn_display _____

- mat `b1' = (1/`mean')*e(b)
= mat __000008 = (1/__000003)*e(b)
- mat score double `xb' = `b1' `if'

```

```

= mat score double __000006 = __000008 if __000002
- LikePois `y' `xb'`poff' [`weight'`exp'] `if', n(`nobs')
= LikePois y_timing_max __000006 [] if __000002, n(__000004)

> ----- begin poisson.LikePois -----
- gettoken y 0 : 0
- gettoken xb 0 : 0
- syntax [fw aw pw iw] [if] , Nobs(string)
- if "`weight'"!=" {
= if ""!=" {
  if "`weight'"=="fweight" {
    local wt `[fw`exp']
  }
  else if "`weight'"=="iweight" {
    local wt `[iw`exp']
  }
  else local wt `[aw`exp']
}
- tempvar lnf
- qui gen double `lnf' = `y'*(`xb')-exp(`xb')-lgamma(`y'+1) `if'
= qui gen double __00000B = y_timing_max*(__000006)-exp(__000006)-
> lgamma(y_timing_max+1) if __000002
- summarize `lnf' `wt' `if', meanonly
= summarize __00000B if __000002, meanonly
- if r(N) != `nobs' { exit }
= if r(N) != __000004 { exit }
- if "`weight'"=="aweight" {
= if ""=="aweight" {
  ret scalar lnf = r(N)*r(mean)
}
- else ret scalar lnf = r(sum)

> ----- end poisson.LikePois -----
- scalar `lnf1' = r(lnf)
= scalar __00000A = r(lnf)
- }
- if "`constan'"==" {
= if ""==" {
- SolveC `y' `xb'`poff' [`weight'`exp'] `if', n(`nobs') mean(`mean')
= SolveC y_timing_max __000006 [] if __000002, n(__000004) mean(__00
> 0003)

> ----- begin poisson.SolveC -----
- gettoken y 0 : 0
- gettoken xb 0 : 0
- syntax [fw aw pw iw] [if] , Nobs(string) Mean(string)

```

```

- if "`weight'"=="pweight" {
= if ""=="pweight" {
    local weight "aweight"
  }
- capture confirm variable `xb'
= capture confirm variable __000006
- if _rc {
    tempvar xbnew
    qui gen double `xbnew' = (`xb') `if'
    local xb `xbnew'
  }
- summarize `xb' [`weight'`exp'] `if', meanonly
= summarize __000006 [] if __000002, meanonly
- if r(N) != `nobs' { exit }
= if r(N) != __000004 { exit }
- if r(max) - r(min) > 2*709 {
    exit
  }
- if r(max) > 709 | r(min) < -709 {
    tempname shift
    if r(max) > 709 { scalar `shift' = 709 - r(max) }
    else scalar `shift' = -709 - r(min)
    local shift "+`shift'"
  }
- tempvar expoff
- qui gen double `expoff' = exp(`xb'`shift') `if'
= qui gen double __00000B = exp(__000006) if __000002
- summarize `expoff' [`weight'`exp'], meanonly
= summarize __00000B [], meanonly
- if r(N) != `nobs' { exit }
= if r(N) != __000004 { exit }
- return scalar _cons = ln(`mean')-ln(r(mean))`shift'
= return scalar _cons = ln(__000003)-ln(r(mean))

```

---

> end poisson.SolveC

```

- tempname c
- scalar `c' = r(_cons)
= scalar __00000B = r(_cons)
- if `c'<. {
= if __00000B<. {
- local c "+`c'"
= local c "+__00000B"
- }
- else local c
- LikePois `y' `xb'`poff'`c' [`weight'`exp'] `if', n(`nobs')
= LikePois y_timing_max __000006+__00000B [] if __000002, n(__000004)

```

```

> )


---


> 

---

 begin poisson.LikePois 

---


    - gettoken y 0 : 0
    - gettoken xb 0 : 0
    - syntax [fw aw pw iw] [if] , Nobs(string)
    - if "`weight'"!="" {
= if ""!="" {
    if "`weight'"=="fweight" {
        local wt `[fw`exp']''
    }
    else if "`weight'"=="iweight" {
        local wt `[iw`exp']''
    }
    else local wt `[aw`exp']''
    }
    - tempvar lnf
    - qui gen double `lnf' = `y'*(`xb')-exp(`xb')-lgamma(`y'+1) `if'
= qui gen double __00000C = y_timing_max*(__000006+__00000B)-exp(_
> _000006+__00000B)-lgamma(y_timing_max+1) if __000002
    - summarize `lnf' `wt' `if', meanonly
= summarize __00000C if __000002, meanonly
    - if r(N) != `nobs' { exit }
= if r(N) != __000004 { exit }
    - if "`weight'"=="aweight" {
= if ""=="aweight" {
        ret scalar lnf = r(N)*r(mean)
    }
    - else ret scalar lnf = r(sum)


---


> 

---

 end poisson.LikePois 

---


    - tempname lnf1c
    - scalar `lnf1c' = r(lnf)
= scalar __00000C = r(lnf)
    - }
    - else local lnf1c .
    - capture {
    - gen double `z' = `y'*exp(-(`xb'`poff'`c'))-1 +`xb'`c' `if'
= gen double __000007 = y_timing_max*exp(-(__000006+__00000B))-1 +__
> 000006+__00000B if __000002
    - `vv' _regress `z' `xvars' [aw=`wt'exp(`xb'`poff'`c')]' `if', `const
> an'
        = version 18, missing: _regress __000007 cb_authors_ext estimated w
> g_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open [aw=exp(__000006
> +__00000B)] if __000002,


---



```

```

> ----- begin _ereturn_display -----
- version 11
- if "`e(b)'" == "" & "`e(V)'" == "" {
= if "matrix" == "" & "matrix" == "" {
    exit
  }
- if "`e(cmd)'" == "regress" {
= if "regress" == "regress" {
- local ALTOPT Beta
- }
- syntax [, eform(passthru) First neq(integer -1) PLus `ALTOPT' *]
= syntax [, eform(passthru) First neq(integer -1) PLus Beta *]
- _get_mcompare, `options'
= _get_mcompare, level(95.00) fwrap(1)

> ----- begin _get_mcompare -----
- version 11
- local cmd ``e(cmd)''
= local cmd `regress'
- syntax [name(name=caller)] [, *]
- local is_pwc = inlist("pwcompare", "`cmd'", "`caller'") | inli
> st("pwmean", "`cmd'", "`caller'")
= local is_pwc = inlist("pwcompare", "regress", "") | inlist("pw
> mean", "regress", "")
- local is_marg = inlist("margins", "`cmd'", "`e(cmd2)'", "`call
> er'")
= local is_marg = inlist("margins", "regress", "", "")
- local OPTIONS NOADJUST BONferroni DUNCAn DUNNett SCHeffe SIDak
> SNK TUKey ADJustall
- if `is_pwc' {
= if 0 {
    syntax [name] [, `OPTIONS' MCOMPare(string) *]
  }
- else {
- syntax [name] [, MCOMPare(string) *]
- local suboption mcompare()
- }
- sreturn clear
- sreturn local options ``options''
= sreturn local options `level(95.00) fwrap(1)''
- if `is_pwc' {
= if 0 {
    local METHOD `noadjust' `bonferroni' `duncan' `dunnett' `schef
> fe' `sidak' `snk' `tukey'
    local METHOD : list uniq METHOD
    opts_exclusive ``METHOD''

```

```

        local ALL `adjustall'
        opts_exclusive "`noadjust' `ALL'"
    }
    - local 0 `", `mcompare'"
    = local 0 `", ""
    - capture syntax [, `OPTIONS']
    = capture syntax [, NOADJUST BONferroni DUNCAn DUNNett SCHeffe S
> IDak SNK TUKey ADJustall]
    - if c(rc) {
        di as err "invalid mcompare() option;"
        di as err "method '`mcompare' not allowed"
        error 198
    }
    - local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `dunnet
> t' `scheffe' `sidak' `snk' `tukey'
    = local METHOD
    - local METHOD : list uniq METHOD
    - local ALL `ALL' `adjustall'
    = local ALL
    - local ALL : list uniq ALL
    - opts_exclusive "`METHOD'" `suboption'
    = opts_exclusive "" mcompare()

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - if "`METHOD'" == "" {
    = if "" == "" {
    - if "`ALL'" != "" {
    = if "" != "" {
        di as err "multiple comparison method required with the adjust
> all option"
        exit 198

```

```

    }
    - local METHOD noadjust
    - local noadjust noadjust
    - }
    - opts_exclusive "`ALL' `noadjust'" `suboption'
    = opts_exclusive " noadjust" mcompare()
  }
}

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts noadjust
    - local n 0
    - while ``opts' != "" {
    = while `"noadjust" != "" {
    - local ++n
    - gettoken item`n' opts : opts, bind
    = gettoken item1 opts : opts, bind
    - }
    - while ``opts' != "" {
    = while `"' != "" {
      local ++n
      gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 1 < 2 {
    - exit
  }
}

> ----- end opts_exclusive -----
    - local EXCLUDE duncan dunnett snk tukey
    - if `:list METHOD in EXCLUDE' {
    = if 0 {
      if `is_pwc' {
        local opt : copy local METHOD
      }
      else {
        local opt "mcompare(`METHOD')"
      }
      local cmdlist anova manova regress mvreg pwmean
      if "`cmd'" == "pwcompare" {
        local cmd ``e(cmd_est)'"
      }
      if `is_marg' {
        local cmd margins
      }
      if !`:list cmd in cmdlist' | !`is_pwc' {

```

```

di as err `"method `opt' is not allowed with results from `cmd
> ""
    exit 322
    }
    if ``e(prefix)'"' == "svy" {
di as err `"method `opt' is not allowed with results using the
> svy prefix"'
    exit 322
    }
    local vce ``e(vce)'"'
    if !inlist("`vce'", "", "ols") {
di as err `"method `opt' is not allowed with results using vce
> (`vce')"'
    exit 322
    }
    }
    - local EXCLUDE duncan dunnett snk tukey scheffe
    - if `:list METHOD in EXCLUDE' {
= if 0 {
    if "`ALL'" != "" {
di as err `"option adjustall is not allowed with method `METHO
> D'"'
    exit 198
    }
    }
    - sreturn local method ``METHOD'"'
    = sreturn local method `noadjust"'
    - sreturn local adjustall ``ALL'"'
    = sreturn local adjustall `"'
    _____
> _____ end _get_mcompare _____
    - if "`s(method)'" != "" {
= if "noadjust" != "" {
    - loca mcopt mcompare(`s(method)' `s(adjustall)')
    = loca mcopt mcompare(noadjust )
    - }
    - _get_diopts diopts, `s(options)'
    = _get_diopts diopts, level(95.00) fwrap(1)
    _____
> _____ begin _get_diopts _____
    - version 11
    - syntax namelist(max=2) [, *]
    - gettoken name1 namelist : namelist
    - gettoken name2 namelist : namelist
    - GetDiopts diopts options, `options'
    = GetDiopts diopts options, level(95.00) fwrap(1)

```



```

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'`` != "" {
    = while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'`` != "" {
    = while `""' != "" {
        local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel

```

```

- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
}

```

```

        sreturn local sformat ``sformat'''
        local sformat `sformat(`sformat)'''
    }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`pformat') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat ``pformat'''
    local pformat `pformat(`pformat)'''
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> \_\_\_\_\_ begin \_get\_diopts.Wrapon \_\_\_\_\_

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, W0rd WIdth]
    exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> \_\_\_\_\_ begin opts\_exclusive \_\_\_\_\_

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```



```

        = if 0 < 2 {
        - exit
    }
}

> ----- end opts_exclusive -----

    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }
}

> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap'" != "" {
    = if "fvwrap(1)" != "" {
    - FVWrap, `fvwrap'
    = FVWrap, fvwrap(1)
}

> ----- begin _get_diopts.FVWrap -----

    - syntax, fvwrap(integer)
    - if `fvwrap' < 0 {
    = if 1 < 0 {
        di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "non
> negative integer required{p_end}"
        exit 198
    }
}

> ----- end _get_diopts.FVWrap -----

    - }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'

    = c_local diopts                fvwrap(1)
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options `''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'

```

```

= sreturn local level 95


---


> 

---

 end _get_diopts.GetDiopts 

---


- c_local `name1' ``diopts'''
= c_local diopts ``fvwrap(1)'''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- local S_`sname' = s(`sname')
= local S_options = s(options)
- }
- sreturn clear
- GetDiopts DEFAULT


---


> 

---

 begin _get_diopts.GetDiopts 

---


- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels'''
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
local ++n
gettoken item`n' opts : opts, bind

```

```

        }
        - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`lstretch` `nolstretch`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport` `fullcnsreport`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
= local baselevels
    - local emptycells `emptycells' `noemptycells'
= local emptycells
    - local omitted `omitted' `noomitted'
= local omitted
    - local lstretch `lstretch' `nolstretch'
= local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
    - local abbrev `abbrev' `noabbrev'
= local abbrev
    - if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
    - if c(showbaselevels) == "all" {
        local allbaselevels allbaselevels
    }
    - else if c(showbaselevels) == "on" {
        local baselevels baselevels
    }
    - }
    - if "`emptycells'" == "" {
= if "" == "" {
    - if c(showemptycells) == "off" {
        local emptycells noemptycells
    }
    - }
    - if "`omitted'" == "" {
= if "" == "" {
    - if c(showomitted) == "off" {
        local omitted noomitted
    }
    - }
    - if "`lstretch'" == "" {
= if "" == "" {
    - if c(lstretch) == "off" {
        local lstretch nolstretch
    }

```



```

- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth(`""cformat'""') > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat `""cformat'""'
  local cformat `cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`""sformat'""') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `""sformat'""'
  local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`""pformat'""') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `""pformat'""'
  local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```
> ----- begin _get_diopts.Wrapon -----
```

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
}

```

```

syntax [, W0rd WIdth]
exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}

```

---

```

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abbre
> v'

= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
  c_local `c_opts' ```options'""
}
- sreturn local coding `coding'
= sreturn local coding

```

```

- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- if ``name2'' == "" {
= if `'' == "" {
- GetDiopts DUPS, `options'
= GetDiopts DUPS,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITtd OMITtd NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITtd OMITtd NOLSTRETCH LSTRETCH COEFLegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----

```

```

- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

```



```

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
    }
  }
}

```

```

di "{txt}note: invalid cformat(), using default"
}
sreturn local cformat ``cformat''
local cformat `cformat(`cformat')''
}
- if `:length local sformat' {
= if 0 {
confirm numeric format `sformat'
if fmtwidth(`sformat') > 8 {
local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat ``sformat''
local sformat `sformat(`sformat')''
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth(`pformat') > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat ``pformat''
local pformat `pformat(`pformat')''
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, W0rd WIdth]
exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'

    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options'
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

```

```

> ----- end _get_diopts.GetDiopts -----
- }
- else {
  GetDiopts DUPS options, `options'
  c_local `name2' ``options'
  }
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS' != "" {
= if `"' != "" {
  if (strpos(`DUPS', "(") {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname'
= sreturn local method = `noadjust'
- }
- sreturn local `sname' = ``S_`sname'
= sreturn local options = `level(95.00) fvwrap(1)'
- }

> ----- end _get_diopts -----
- if !:length local first' {
= if !0 {
- local first showeqns
- }
- if `neq' < 1 {
= if -1 < 1 {
- _ms_eq_info
- local neqopt neq(`r(k_eq)')
= local neqopt neq(1)
- }
- else {
  local neqopt neq(`neq')
}
- _coef_table, nodiparm nocnsreport noeqcheck eformall `eform' `fi
> rst' `neqopt' `plus' `beta' `mcopt' `diopts'

```

```

= _coef_table, nodiparm nocnsreport noeqcheck eformall showeqns n
> eq(1) mcompare(noadjust ) fvwrap(1)

> ----- begin _coef_table -----
- version 11
- local vv : di "version " string(max(11,_caller())) ", missing:
> "
- if (!c(noisily) & c(coeftabresults) == "off") {
  exit
}
- _check_eclass
- syntax [, BMATrix(passthru) VMATrix(passthru) * ]
- if "`e(b)'" == "" & "`e(V)'" == "" & "`bmatrix'`vmatrix'" == ""
> {
= if "matrix" == "" & "matrix" == "" & "" == "" {
  exit
}
- local mc_cmds contrast margins pwcompare pwmean
- local cmd "`e(cmd)'"
= local cmd "regress"
- local keepmc : list cmd in mc_cmds
- if inlist("`cmd'", "pwcompare", "pwmean") {
= if inlist("regress", "pwcompare", "pwmean") {
  local groups GROUPS
}
- if "`cmd'" == "regress" {
= if "regress" == "regress" {
- local beta Beta
- }
- if "`cmd'" == "gsem" {
= if "regress" == "gsem" {
  local fvignore fvignore(int 0)
  local flignore flignore
}
- if "`cmd'" == "sem" {
= if "regress" == "sem" {
  local standardized STANDARDIZED
  local showginvariant SHOWGINvariant
  local nolabel NOLABEL LABEL NOFVLABEL FVLABEL
  local nofootnote NOFOOTnote
  local wrap wrap(numlist max=1) fvwrap(passthru)
}
- if "`e(mi)'" == "mi" {
= if "" == "mi" {
  local dftable DFTable
  local dfonly DFONLY

```

```

        local noclustreport NOCLUSTReport
        local pisematrix PISEMATrix(string)
    }
    - syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st
> ring) DFMATrix(string) EQMATrix(string) ROWMATrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMATrix(string) MEMATrix(string) BSTDMATrix(string) CNSMATrix(string) P
> CLASSMATrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) `beta' `standardized' NOCI NOPValues `dftable' `dfonly' dfci DFPVal
> ues dfmissing `groups' COEFLegend selegend `fvignore' `flignore' `showginvar
> iant' `nolabel' `nofootnote' `wrap' sort `pisematrix' depname(string) COEFTi
> tle(string) coeftitle2(string) ptitle(string) cititle(string) NOMCLEgend noC
> NSReport FULLCNSReport `noclustreport' cformat(passthru) sformat(passthru) p
> format(passthru) NOFirst First SHOWEQns neq(integer -1) NODIPARM NOTEST SEPa
> rator(integer 0) NOSKIP OFFSETONLY1 PPlus NOEQCHECK EFORMALL CITYPE(string) e
> qselect(string) noEFORMNOTE * ]

    = syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st
> ring) DFMATrix(string) EQMATrix(string) ROWMATrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMATrix(string) MEMATrix(string) BSTDMATrix(string) CNSMATrix(string) P
> CLASSMATrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) Beta NOCI NOPValues dfci DFPValues dfmissing COEFLegend selegen
> d sort depname(string) COEFTitle(string) coeftitle2(string) ptitle(st
> ring) cititle(string) NOMCLEgend noCNSReport FULLCNSReport cformat(passthru
> ) sformat(passthru) pformat(passthru) NOFirst First SHOWEQns neq(integer -1)
> NODIPARM NOTEST SEPARATOR(integer 0) NOSKIP OFFSETONLY1 PPlus NOEQCHECK EFOR
> MALL CITYPE(string) eqselect(string) noEFORMNOTE * ]

    - if "`wrap'" != "" {
    = if "" != "" {
        opts_exclusive `"wrap(`wrap')' `fvwrap'"
        local fvwrap fvwrap(`wrap')
    }
    - if "`nolabel'\`label'" != "" {
    = if "" != "" {
        opts_exclusive "`nolabel' `label'"
        opts_exclusive "`nofvlabel' `fvlabel'"
        opts_exclusive "`nolabel' `fvlabel'"
        opts_exclusive "`nofvlabel' `label'"
        if "`nolabel'" != "" {
            local fvlabel nofvlabel
        }
        else {
            local fvlabel fvlabel
        }
    }
    - else local fvlabel `nofvlabel' `fvlabel'

```

```

= else local fvlabel
- local type `coeflegend' `selegend'
= local type
- opts_exclusive ``type' `standardized'
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
  }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- if (``rowmatrix'=="" & ``norowci'!="") {
= if (``'=="" & ``'!="") {
    di as err "{bf:norowci} requires {bf:rowmatrix()}"
    exit 198
  }
- if (``dftable'!="" & ``norowci'!="") {
= if (``'!="" & ``'!="") {
    di as err "{bf:norowci} is not allowed with {bf:`dftable'}"
    exit 198
  }
- if (``dfonly'!="" & ``norowci'!="") {
= if (``'!="" & ``'!="") {
    di as err "{bf:norowci} is not allowed with {bf:`dfonly'}"
    exit 198
  }
- if (``noci'!="" & ``norowci'!="") {
= if (``'!="" & ``'!="") {
    di as err "{bf:norowci} is not allowed with {bf:`noci'}"
    exit 198
  }
- if ``standardized' != "" & ``bstdmatrix' != "" {
= if ``' != "" & ``' != "" {
    local beta beta
    local standardized

```

```

    }
    - if `:length local type' == 0 {
    = if 0 == 0 {
    - local type `beta' `nocl' `nopvalues' `dftable' `dfonly' `group
> s' `dfci' `dfpvalues'
    = local type
    - }
    - opts_exclusive ``type''
    = opts_exclusive ""

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts'' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive ``first' `nofirst''
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts'' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive ``first' `showeqns''

```



```

= opts_exclusive " showeqns"
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts showeqns
- local n 0
- while ``opts'' != "" {
= while "showeqns" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 1 < 2 {
- exit
> ----- end opts_exclusive -----
- local cnsreport `cnsreport' `fullcnsreport'
= local cnsreport nocnsreport
- opts_exclusive "cnsreport"
= opts_exclusive "nocnsreport"
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts nocnsreport
- local n 0
- while ``opts'' != "" {
= while "nocnsreport" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'' != "" {
= while `'' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----
- if ``showeqns''' != "" {
= if `"showeqns"' != "" {
- local nofirst nofirst
- }
- if ``ematrix'\`noeqcheck''' == "" {
= if `"noeqcheck"' == "" {
  if ``e(error)''' == "matrix" {
    local ematrix e(error)
  }
}
- _get_mcompare, `options'
= _get_mcompare, mcompare(noadjust ) fvwrap(1)

> ----- begin _get_mcompare -----
- version 11
- local cmd ``e(cmd)'''
= local cmd `"regress"'
- syntax [name(name=caller)] [, *]
- local is_pwc = inlist("pwcompare", "`cmd'", "`caller'") | in
> list("pwmean", "`cmd'", "`caller'")
= local is_pwc = inlist("pwcompare", "regress", "") | inlist("
> pwmean", "regress", "")
- local is_marg = inlist("margins", "`cmd'", "`e(cmd2)'", "`ca
> ller'")
= local is_marg = inlist("margins", "regress", "", "")
- local OPTIONS NOADJUST BONferroni DUNCAn DUNNett SCHeffe SID
> ak SNK TUKey ADJustall
- if `is_pwc' {
= if 0 {
  syntax [name] [, `OPTIONS' MCOMPare(string) *]
}
- else {
- syntax [name] [, MCOMPare(string) *]
- local suboption mcompare()
- }
- sreturn clear
- sreturn local options ``options'''
= sreturn local options `"fvwrap(1)'"
- if `is_pwc' {
= if 0 {
  local METHOD `noadjust' `bonferroni' `duncan' `dunnett' `sch

```

```

> effe' `sidak' `snk' `tukey'
    local METHOD : list uniq METHOD
    opts_exclusive ``METHOD''
    local ALL `adjustall'
    opts_exclusive ``noadjust' `ALL''
    }
    - local 0 `", `mcompare'''
    = local 0 `", noadjust''
    - capture syntax [, `OPTIONS']
    = capture syntax [, NOADJUST BONferroni DUNCAn DUNNett SCHeffe
> SIDak SNK TUKey ADJustall]
    - if c(rc) {
        di as err "invalid mcompare() option;"
        di as err "method ``mcompare'' not allowed"
        error 198
    }
    - local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `dunn
> ett' `scheffe' `sidak' `snk' `tukey'
    = local METHOD noadjust
    - local METHOD : list uniq METHOD
    - local ALL `ALL' `adjustall'
    = local ALL
    - local ALL : list uniq ALL
    - opts_exclusive ``METHOD'' `suboption'
    = opts_exclusive "noadjust" mcompare()



---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts noadjust
    - local n 0
    - while ```opts''' != "" {
    = while ``noadjust'' != "" {
    - local ++n
    - gettoken item`n' opts : opts, bind
    = gettoken item1 opts : opts, bind
    - }
    - while ```opts''' != "" {
    = while `'''' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 1 < 2 {
    - exit

```

```

> ----- end opts_exclusive -----
- if "`METHOD'" == "" {
= if "noadjust" == "" {
  if "`ALL'" != "" {
    di as err "multiple comparison method required with the adju
> stall option"
    exit 198
  }
  local METHOD noadjust
  local noadjust noadjust
}
- opts_exclusive "`ALL' `noadjust'" `suboption'
= opts_exclusive " noadjust" mcompare()

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts noadjust
- local n 0
- while ``opts'" != "" {
= while `"noadjust'" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 1 < 2 {
- exit

> ----- end opts_exclusive -----
- local EXCLUDE duncan dunnett snk tukey
- if `:list METHOD in EXCLUDE' {
= if 0 {
  if `is_pwc' {
    local opt : copy local METHOD
  }
  else {
    local opt "mcompare(`METHOD')"
  }
}

```

```

        local cmdlist anova manova regress mvreg pwmean
        if "`cmd'" == "pwcompare" {
            local cmd ``e(cmd_est)'''
        }
        if `is_marg' {
            local cmd margins
        }
        if !`:list cmd in cmdlist' | !`is_pwc' {
            di as err `"method `opt' is not allowed with results from `c
> md'''

            exit 322
        }
        if ``e(prefix)''' == "svy" {
            di as err `"method `opt' is not allowed with results using t
> he svy prefix'''

            exit 322
        }
        local vce ``e(vce)'''
        if !inlist("`vce'", "", "ols") {
            di as err `"method `opt' is not allowed with results using v
> ce(`vce')'''

            exit 322
        }
        }
        - local EXCLUDE duncan dunnett snk tukey scheffe
        - if `:list METHOD in EXCLUDE' {
        = if 0 {
            if "`ALL'" != "" {
                di as err `"option adjustall is not allowed with method `MET
> HOD'''

                exit 198
            }
        }
        - sreturn local method ``METHOD'''
        = sreturn local method `noadjust'
        - sreturn local adjustall ``ALL'''
        = sreturn local adjustall `'''

        _____
> _____ end _get_mcompare _____

        - local method ``s(method)'''
        = local method `noadjust'
        - local all ``s(adjustall)'''
        = local all `'''
        - local options ``s(options)'''
        = local options `fvwrap(1)'''
        - if "`method'" != "noadjust" {

```

```

= if "noadjust" != "noadjust" {
  local keepmc 1
}
- opts_exclusive "`all' `groups'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- if "`method'" == "dunnett" {
= if "noadjust" == "dunnett" {
  opts_exclusive "`method' `groups'"
}
- _get_diopts diopts options, `options' `fvwrap' `fvlabel'
= _get_diopts diopts options, fvwrap(1)


---


> 

---

 begin _get_diopts 

---


- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options, fvwrap(1)


---


> 

---

 begin _get_diopts.GetDiopts 

---


- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPvalues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE

```

```

> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITTED OMITTED NOLSTRETCH LSTRETCH COEFLEGEND SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *)
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`baselevels' `nobaselevels'"
    = opts_exclusive " "


---



```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```



```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

```

```

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive ``coding' `compare'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``versus' `compare'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
}

```

```

- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
  }

```

```

        sreturn local pformat ``pformat'''
        local pformat `pformat(pformat)'''
    }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

```

```

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "fvwrap(1)" != "" {
- FVWrap, `fvwrap'
= FVWrap, fvwrap(1)

> ----- begin _get_diopts.FVWrap -----
- syntax, fvwrap(integer)
- if `fvwrap' < 0 {
= if 1 < 0 {
di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n
> onnegative integer required{p_end}"
exit 198
}

> ----- end _get_diopts.FVWrap -----
- }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local diopts fvwrap(1)
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options''
= c_local options ``''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts''
= c_local diopts `fvwrap(1)'
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }

```

```

- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- local S_`sname' = s(`sname')
= local S_options = s(options)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(strin
> g) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----

```



```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"

```

```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels

```

```

- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'

```

```

}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`"'sformat'"') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `"'sformat'"'
  local sformat `"'sformat(`sformat')'"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`"'pformat'"') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `"'pformat'"'
  local pformat `"'pformat(`pformat')'"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "'word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

        = local opts
        - local n 0
        - while ``opts'' != "" {
        = while `'' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if ``fvwrap'' != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local DEFAULT
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----
    - if ``name2'' == "" {
    = if ``options'' == "" {

```



```

        GetDiopts DUPS, `options'
    }
    - else {
    - GetDiopts DUPS options, `options'
    = GetDiopts DUPS options,
    _____
> _____ begin _get_diopts.GetDiopts _____
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`versus' `compare'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - opts_exclusive "`abbrev' `noabbrev'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'

```

```

= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {

```

```

confirm numeric format `sformat'
if fmtwidth("`sformat'") > 8 {
  local sformat "%8.2f"
  di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat "`sformat'"
local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```



```

        = while `""` != "" {
            local ++n
            gettoken item`n` opts : opts, bind
        }
        - if `n` < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - local fvwrapon `word` `width`
    = local fvwrapon
    - if `:length local fvwrapon` {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon`)
    }

> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap`" != "" {
    = if "" != "" {
        FVWrap, `fvwrap`
    }
    - c_local `c_diopts` `levelopt` `vsquish` `allbaselevels` `b
> aselevels` `cnsreport` `fullcnsreport` `emptycells` `omitted` `lstretch` `co
> eflegend` `selegend` `cformat` `sformat` `pformat` `coding` `versus` `compar
> e` `fvwrap` `fvwrapon` `fvlabel` `nocl` `nopvalues` `markdown` `citype` `abb
> rev`

    = c_local DUPS
    - if `K` == 2 {
    = if 2 == 2 {
        - c_local `c_opts` ``options``
        = c_local options ``
        - }
        - sreturn local coding `coding`
        = sreturn local coding
        - sreturn local versus `versus`
        = sreturn local versus
        - sreturn local compare `compare`
        = sreturn local compare
        - sreturn local level `level`
        = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

    - c_local `name2` ``options``
    = c_local options ``
    - }
    - local DUPS : list DUPS - DEFAULT

```

```

- gettoken DUPS : DUPS
- if ``DUPS'' != "" {
= if `""' != "" {
  if (strpos(`DUPS', "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'""
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local method = `noadjust'
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local options = `fvwrap(1)'
- }

```

---

```

> ----- end _get_diopts -----
- local legend coeflegend selegend
- local legend : list legend & diopts
- local diopts : list diopts - legend
- local lstretch lstretch nolstretch
- local lstretch : list lstretch & diopts
- local diopts : list diopts - lstretch
- local markdown markdown
- local markdown : list markdown & diopts
- local diopts : list diopts - markdown
- _get_diopts ignore, `diopts' `cformat' `sformat' `pformat'
= _get_diopts ignore, fvwrap(1)

```

---

```

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options, fvwrap(1)

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY

```

```

> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
      - syntax namelist(max=2) [, `DIOPTS' *]
      = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
      - opts_exclusive "`allbaselevels' `noallbaselevels'"
      = opts_exclusive " "

> ----- begin opts_exclusive -----
      - version 8.2
      - args opts optname errcode optslab
      - local opts `opts'
      = local opts
      - local n 0
      - while ``opts' != "" {
      = while `"' != "" {
          local ++n
          gettoken item`n' opts : opts, bind
          }
      - if `n' < 2 {
      = if 0 < 2 {
      - exit

> ----- end opts_exclusive -----
      - opts_exclusive "`allbaselevels' `nobaselevels'"
      = opts_exclusive " "

> ----- begin opts_exclusive -----
      - version 8.2
      - args opts optname errcode optslab
      - local opts `opts'
      = local opts
      - local n 0
      - while ``opts' != "" {
      = while `"' != "" {
          local ++n
          gettoken item`n' opts : opts, bind
          }
      - if `n' < 2 {
      = if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'`` != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'`` != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {

```



```

- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {

```

```

= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`"pformat"') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `"pformat"'
    local pformat `pformat(pformat)'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"opts"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----

```

```

- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
  }
}

> ----- end _get_diopts.Wrapon -----

- if "`fvwrap'" != "" {
= if "fvwrap(1)" != "" {
- FVWrap, `fvwrap'
= FVWrap, fvwrap(1)
}

> ----- begin _get_diopts.FVWrap -----

- syntax, fvwrap(integer)
- if `fvwrap' < 0 {
= if 1 < 0 {
    di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n
> onnegative integer required{p_end}"
    exit 198
  }
}

> ----- end _get_diopts.FVWrap -----

- }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local diopts          fvwrap(1)
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options'
= c_local options `''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

- c_local `name1' ``diopts'

```

```

= c_local ignore `fvwrap(1)`
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_options = s(options)
- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nibasebaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nibasebaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts"" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`nocl' `nopvalues'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0

```



```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'

```

```

if fmtwidth("`cformat'") > 9 {
local cformat "%9.0g"
di "{txt}note: invalid cformat(), using default"
}
sreturn local cformat "`cformat'"
local cformat `cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
confirm numeric format `sformat'
if fmtwidth("`sformat'") > 8 {
local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat "`sformat'"
local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth("`pformat'") > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat "`pformat'"
local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, WOrd WIdth]
exit 198
}
- opts_exclusive "`word' `width'" fvwrapon

```

```

= opts_exclusive " " fvwrapon
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}
> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}
> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'
= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options'
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare

```

```

- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- if ``name2''' == "" {
= if `''' == "" {
- GetDiopts DUPS, `options'
= GetDiopts DUPS,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(strin
> g) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"

```

```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0

```



```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels

```

```

- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'

```

```

}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`"'sformat'"') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `"'sformat'"'
  local sformat `"'sformat(`sformat')'"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`"'pformat'"') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `"'pformat'"'
  local pformat `"'pformat(`pformat')'"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "'word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

        = local opts
        - local n 0
        - while ``opts'' != "" {
        = while `'' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if ``fvwrap'' != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----
    - }
    - else {

```

```

        GetDiopts DUPS options, `options'
        c_local `name2' ``options'""
    }
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'"" != "" {
= if `""' != "" {
    if (strpos(`DUPS'"" , "(")) {
        gettoken DUPS : DUPS, parse("(")
        local DUPS `DUPS'()'""
    }
    di as err "duplicate {bf:`DUPS'} option not allowed"
    exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'""
= sreturn local level = `95'""
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local options = `fvwrap(1)'""
- }
- sreturn local `sname' = ``S_`sname'""
= sreturn local method = `noadjust'""
- }

> ----- end _get_diopts -----

- local cformat ``s(cformat)'""
= local cformat `""'
- local sformat ``s(sformat)'""
= local sformat `""'
- local pformat ``s(pformat)'""
= local pformat `""'
- _get_eformopts , eformopts(`options') allowed(__all__) soption
> s
= _get_eformopts , eformopts() allowed(__all__) soptions

> ----- begin _get_eformopts -----
- version 8.0
- syntax [, eformopts(string asis) soptions ALLOWed(string) ]
- local 0 , `eformopts'
= local 0 ,
- local EFALL NOHR hr NOSHR shr or IRr RRr tr TRatio
- if ``soptions'"" != "" {
= if "soptions" != "" {
- syntax [, EForm1(passthru) EForm `EFALL' * ]

```

```

= syntax [, EForm1(passthru) EForm NOHR hr NOSHR shr or IRr RR
> r tr TRatio * ]
- local opts ``options''
= local opts ``''
- local 0 , `eform1' `eform' `nohr' `hr' `noshr' `shr' `or' `i
> rr' `rrr' `tr' `tratio'
= local 0 ,
- }
- if ``allowed'' == "__all__" {
= if "__all__" == "__all__" {
- local allowed `EFALL'
= local allowed NOHR hr NOSHR shr or IRr RRR tr TRatio
- }
- foreach ef of local allowed {
- capture confirm name `ef'
= capture confirm name NOHR
- if _rc {
di as err ``ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr
- }
- capture confirm name `ef'
= capture confirm name hr
- if _rc {
di as err ``ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr
- }
- capture confirm name `ef'
= capture confirm name NOSHR
- if _rc {
di as err ``ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr
- }
- capture confirm name `ef'
= capture confirm name shr
- if _rc {
di as err ``ef' is not a valid name"
exit 198
}

```

```

    }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr
- }
- capture confirm name `ef'
= capture confirm name or
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or
- }
- capture confirm name `ef'
= capture confirm name IRr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr
- }
- capture confirm name `ef'
= capture confirm name RRr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr
- }
- capture confirm name `ef'
= capture confirm name tr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr
- }
- capture confirm name `ef'
= capture confirm name TRatio
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }

```



```

- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr tratio
- }
- local efopts : list uniq efopts
- syntax [, EForm1(string) EForm `allowed' ]
= syntax [, EForm1(string) EForm NOHR hr NOSHR shr or IRr RRr

> tr TRatio ]

- foreach ef of local efopts {
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- if ```eform1''''=="Odds ratio" {
= if `""'=="Odds ratio" {
    local eform_cons_ti = "Odds"
    local eform_cons_note = "odds ratios"
  }
- else if ```eform1''''=="Haz. ratio" {
= else if `""'=="Haz. ratio" {
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
  }
- else if ```eform1''''=="Time ratio" {

```

```

= else if `""'=="Time ratio" {
  local eform_cons_ti = "Time"
  local eform_cons_note = "time ratios"
}
- else if ```eform1'""'=="IRR" {
= else if `""'=="IRR" {
  local eform_cons_ti = "Inc. rate"
  local eform_cons_note = "incidence-rate ratios"
  local eform_cons_ti2 = "incidence rate"
}
- else if ```eform1'""'=="Risk ratio" {
= else if `""'=="Risk ratio" {
  local eform_cons_ti = "Risk"
  local eform_cons_note = "risk ratios"
}
- else if ```eform1'""'=="Hlth ratio" {
= else if `""'=="Hlth ratio" {
  local eform_cons_ti = "Health"
  local eform_cons_note = "health ratios"
}
- local k : list sizeof eform
- if `k' {
= if 0 {
  opts_exclusive ``eform'
  if `:length local eform1' {
    opts_exclusive "eform() `eform'"
  }
}
- if `k' {
= if 0 {
  if (``eform'=="eform") local eform1 = "exp(b)"
  else if (``eform'=="hr") {
    local eform1 = "Haz. ratio"
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
  }
  else if (``eform'=="shr") {
    local eform1 = "SHR"
    local eform_cons_ti = "Subhazard"
    local eform_cons_note = "subhazard ratios"
  }
  else if (``eform'=="tr") {
    local eform1 = "Time ratio"
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
  }
}

```

```

else if ("`eform'"=="tratio") {
  local eform1 = "Time ratio"
  local eform_cons_ti = "Time"
  local eform_cons_note = "time ratios"
}
else if ("`eform'"=="or") {
  local eform1 = "Odds ratio"
  if ("`e(cmd)'"=="asclogit" | "`e(cmd)'"=="cmclogit" | "`e(cm
> d)'"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
    local eform_cons_ti = "Rel. risk"
    local eform_cons_note = "relative-risk ratios"
    local eform_cons_ti2 = "relative risk"
  }
  else {
    local eform_cons_ti = "Odds"
    local eform_cons_note = "odds ratios"
  }
}
else if ("`eform'"=="irr") {
  local eform1 = "IRR"
  local eform_cons_ti = "Inc. rate"
  local eform_cons_note = "incidence-rate ratios"
  local eform_cons_ti2 = "incidence rate"
}
else if ("`eform'"=="rrr") {
  local eform1 = "RRR"
  local eform_cons_ti = "Rel. risk"
  local eform_cons_note = "relative-risk ratios"
  local eform_cons_ti2 = "relative risk"
}
else if ("`eform'"!="nohr") & ("`eform'"!="noshr") {
  local eform1 = upper("`eform'")
}
}

- sreturn clear
- sreturn local options ```opts'``
= sreturn local options `''`
- sreturn local opt `eform'
= sreturn local opt
- sreturn local str `eform1'
= sreturn local str
- if ```eform1'`` != "" {
= if `''` != "" {
  sreturn local eform eform(```eform1'``')
}
- else sreturn local eform ""

```

```

- sreturn local eform_cons_ti `eform_cons_ti'
= sreturn local eform_cons_ti
- sreturn local eform_cons_note `eform_cons_note'
= sreturn local eform_cons_note
- sreturn local eform_cons_ti2 `eform_cons_ti2'
= sreturn local eform_cons_ti2

```

---

```

> ----- end _get_eformopts -----
- local tr_item = ``s(eform_cons_note)''
= local tr_item = ``''
- if inlist("`bmatrix'", "", "e(b)") & "`e(k_eform)'" == "0" {
= if inlist("", "", "e(b)") & "" == "0" {
  local eform
  }
- else local eform ``s(str)''
= else local eform ``''
- local eform_cons_ti ``s(eform_cons_ti)''
= local eform_cons_ti ``''
- local eform_cons_ti2 ``s(eform_cons_ti2)''
= local eform_cons_ti2 ``''
- if (`e(consonly)'!="1" | ``eform_cons_ti'')=="" ) {
= if (""!="1" | ``'')=="" ) {
- local eformdi ``s(str)''
= local eformdi ``''
- }
- else {
  local eformdi ``eform_cons_ti''
  local tr_item = cond(missing("`eform_cons_ti2'"), strlower("`
> eform_cons_ti''), ``eform_cons_ti2'')
  }
- local coefttl = cond(``eform'')==``, ``coeftitle'', ``ef
> ormdi'')
= local coefttl = cond(``'')==``, ``', ``')
- local options ``s(options)''
= local options ``''
- _get_diparmopts, diparmopts(`options') level(`level')
= _get_diparmopts, diparmopts() level(95)

```

---

```

> ----- begin _get_diparmopts -----
- if _caller() < 8.2 {
  _get_diparmopts_8 `0'
  exit
  }
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom pl

```

```

> us SYNTAXonly * ]
    - local goptions `options'
    = local goptions level(95)
    - SpliceGlobals junk , `goptions'
    = SpliceGlobals junk , level(95)


---


> _____ begin _get_diparmopts.SpliceGlobals _____
    - syntax name(id="macro name" name=c_diparm) [, diparm(strin
> g asis) Level(cilevel) dof(passthru) syntaxonly ]
    - if ``dof''' == "" & ``e(df_r)''' != "" {
    = if `''' == "" & `212''' != "" {
    - capture confirm integer number `e(df_r)'
    = capture confirm integer number 212
    - if !c(rc) {
    - local gdof dof(`e(df_r)')
    = local gdof dof(212)
    - }
    - capture
    - }
    - else local gdof ``dof'''
    = else local gdof `'''
    - local glevel `level'
    = local glevel 95
    - local lopt level(`glevel')
    = local lopt level(95)
    - if (``diparm''' == "") exit
    = if (``''' == "") exit


---


> _____ end _get_diparmopts.SpliceGlobals _____
    - local 0 `", `diparmopts'''
    = local 0 `", ""
    - local diparmopts
    - local k 0
    - syntax [, diparm(string asis) * ]
    - while ``diparm''' != "" {
    = while `''' != "" {
        local ++k
        local single 1
        local after
        capture _on_colon_parse `diparm'
        if !c(rc) {
            local single 0
            local diparm ``s(before)'''
            local after ``s(after)'''
        }
        capture

```

```

SpliceGlobals diparm , diparm(`diparm`) `syntaxonly' `goptio
> ns'

local diparm`k' ```diparm'''
local diparmopts ```diparmopts' : `diparm'''
if `single' | ```:list retok after''' == "" {
local 0 `", `options'''
syntax [, diparm(string asis) * ]
}
else {
local diparm ```after'''
}
}
- if ```soptions''' == "" {
= if `''' == "" {
- syntax [, diparm(string asis) ]
- }
- if ("`execute" == "") {
= if ("" == "") {
- sreturn clear
- sreturn local options ```options'''
= sreturn local options `'''
- sreturn local diparm ```diparmopts'''
= sreturn local diparm `'''
- forval i = 1/`k' {
= forval i = 1/0 {
sreturn local diparm`i' ```diparm`i''''
}
- sreturn local k `k'
= sreturn local k 0
- exit

> ----- end _get_diparmopts -----
- local NODIPARM : length local nodiparm
- if `NODIPARM' {
= if 8 {
- local options
- }
- local GTOPTS parse(":") bind quotes
- local k 0
- if !`NODIPARM' {
= if !8 {
if ```e(diparm)''' != "" {
local i 0
local diparm ```e(diparm)'''
}
else if ```e(diparm1)''' != "" {

```

```

local i 1
local diparm ``e(diparm1)'''
}
while `:length local diparm' {
  gettoken diparm rest : diparm, `GTPTS'
  while `:length local rest' {
    local ++k
    local diparm`k' : copy local diparm
    gettoken COLON rest : rest, `GTPTS'
    gettoken diparm rest : rest, `GTPTS'
  }
  local ++k
  local diparm`k' : copy local diparm
  local ++i
  local diparm ``e(diparm`i)'''
}
}
- if `:length local options' {
= if 0 {
  local 0 `", `options'''
  syntax [, diparm(string asis) *]
  if `k' & `:length local diparm' {
    local ++k
    local diparm`k' __sep__
  }
  while `:length local diparm' {
    gettoken diparm rest : diparm, `GTPTS'
    while `:length local rest' {
      local ++k
      local diparm`k' : copy local diparm
      gettoken COLON rest : rest, `GTPTS'
      gettoken diparm rest : rest, `GTPTS'
    }
    local ++k
    local diparm`k' : copy local diparm
    local 0 `", `options'''
    syntax [, diparm(string asis) *]
  }
}
- if "`markdown'" != "" {
= if "" != "" {
  local lsize = c(linesize)
  set linesize 255
}
- `vv' mata: _coef_table()
= version 11, missing: mata: _coef_table()

```

```

> ----- begin _get_diopts -----
    - version 11
    - syntax namelist(max=2) [, *]
    - gettoken name1 namelist : namelist
    - gettoken name2 namelist : namelist
    - GetDiopts diopts options, `options'
    = GetDiopts diopts options, fwrap(1)

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(strin
> g) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----

```



```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"

```

```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels

```

```

- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'

```

```

}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`"'sformat'"') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `"'sformat'"'
  local sformat `"'sformat(`sformat')'"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`"'pformat'"') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `"'pformat'"'
  local pformat `"'pformat(`pformat')'"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> 

---

 begin \_get\_diopts.Wrapon 

---

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "'word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> 

---

 begin opts\_exclusive 

---

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

        = local opts
        - local n 0
        - while ``opts'' != "" {
        = while `'' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "fvwrap(1)" != "" {
    - FVWrap, `fvwrap'
    = FVWrap, fvwrap(1)

> ----- begin _get_diopts.FVWrap -----
    - syntax, fvwrap(integer)
    - if `fvwrap' < 0 {
    = if 1 < 0 {
        di as err "{p}invalid option {bf:fvwrap(`fvwrap')}"; " "n
> onnegative integer required{p_end}"
        exit 198
    }

> ----- end _get_diopts.FVWrap -----
    - }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local diopts                fvwrap(1)
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options `''

```



```

- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts'""
= c_local __00000D ``fvwrap(1)""
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_k = s(k)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive ``allbaselevels' `noallbaselevels'""
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

        }
        - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}
> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}
> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist

```

```

- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
    }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
    }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
    }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
    }
- }

```

```

- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `"cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `"sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `"pformat(`pformat')"'
  }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
  }
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
  }
- Wraon , `fvwraon'
= Wraon ,

```



```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd Width]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd Width]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DEFAULT
- if `K' == 2 {

```

```

= if 1 == 2 {
    c_local `c_opts' ``options'
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
    - if ``name2' == "" {
= if ``' == "" {
    - GetDiopts DUPS, `options'
= GetDiopts DUPS,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while ``' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----

```

```

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }

```



```

- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth(`""cformat'""') > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat `""cformat'""'
  local cformat `"cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`""sformat'""') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `""sformat'""'
  local sformat `"sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`""pformat'""') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `""pformat'""'
  local pformat `"pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> \_\_\_\_\_ begin \_get\_diopts.Wrapon \_\_\_\_\_

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
}

```

```

        exit 198
    }
    - opts_exclusive "`word' `width'" fwrapon
= opts_exclusive " " fwrapon


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - local fwrapon `word' `width'
    = local fwrapon
    - if `:length local fwrapon' {
    = if 0 {
        c_local fwrapon fwrapon(`fwrapon')
    }


---


> 

---

 end _get_diopts.Wrapon 

---


    - if "`fwrap'" != "" {
    = if "" != "" {
        FVWrap, `fwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fwrap' `fwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options'
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'

```

```

= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts -----
- }
- else {
  GetDiopts DUPS options, `options'
  c_local `name2' ``options'
}
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS' != "" {
= if `"' != "" {
  if (strpos(`DUPS', "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `DUPS'()'
  }
  di as err "duplicate {bf:`DUPS'} option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname'
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname'
= sreturn local k = `0'
- }

```

---

```

> ----- end _get_diopts -----
- if ``markdown' != "" {
= if "" != "" {
  set linesize `lsize'
}
- if ``eqselect' == "" & ``eformnote' != "noeformnote" {
= if "" == "" & "" != "noeformnote" {
- if ``eformdi' != "" & (`e(cmd)' != "gnbreg" | ``e(prefix)' ==
> "svy") {
= if `"' != "" & ("regress" != "gnbreg" | "" == "svy") {
  local k_eform `r(k_eform)'
  if (`k_eform' == "") {
    local k_eform = 1
  }
}

```

```

        local k_eq `e(k_eq)'
        if ("`k_eq'"=="") {
            local k_eq = 1
        }
        if (`k_iform'<`k_eq' & `k_iform') {
            if ("`e(cmd2)'"=="stintreg" | "`e(cmd2)'"=="streg") & ("`e
> (cmd)'"=="weibull" | "`e(cmd)'"=="gompertz") {
            }
            else {
                _eform_multeq_note efnote : `k_iform' ``tr_item'''
                di as txt ``efnote'''
            }
        }
        - if (``eformdi'''!=" & "`e(noconstant)'"=="0" & ``eform_cons
> _ti'''!=" & "`e(consonly)'"=="1") {
            = if (``'''!=" & ""=="0" & ``'''!=" & ""=="1") {
                if ("`e(cmd)'"=="asclogit" | "`e(cmd)'"=="cmclogit" | "`e(cmd)
> '"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
                    local asvar = ``e(ranvars)``e(fixvars)``e(indvars)''' != ""
                    if ``e(casevars)''' != "" & `asvar' {
                        local note1 "Exponentiated coefficients represent odds ratios
> for "
                        local note2 "alternative-specific variables (first equation) a
> nd "
                        local note3 "{help j_cmrisk##|_new:relative-risk ratios} "
                        local note4 "for case-specific variables."
                        di as txt "{p 0 6 2}Note: `note1'`note2'`note3'`note4'{p_end}"
                    }
                }
                if "`eform_cons_ti'"=="Inc. rate" {
                    local eform_cons_note "incidence rate"
                }
                else if "`eform_cons_ti'"=="Rel. risk" {
                    local eform_cons_note "relative risk"
                }
                else if "`eform_cons_ti'"=="Health" {
                    local eform_cons_note "health (probability of no disease)"
                }
                else {
                    local eform_cons_note = strlower("`eform_cons_ti'")
                }
                local eform_cons_note "baseline `eform_cons_note'"
                if ("`e(cmd2)'"=="") {
                    local cmd `e(cmd)'
                }

```

```

else {
  local cmd `e(cmd2)'
}
local f2 = udsustr("`cmd'",1,2)
local is_re = ("`f2'"=="xt" | "`f2'"=="me")
if `is_re' {
  local extranote " (conditional on zero random effects)"
}
di as txt "{p 0 6 2}Note: {bf:_cons} estimates " "`eform_cons_
> note`extranote'" _c
  if ("`e(cmd)'"=="mlogit" | "`e(cmd)'"=="asclogit" | "`e(cmd)'"
> == "cmclgit" | "`e(cmd)'"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
    di as txt " for each outcome" _c
  }
  di as txt ".{p_end}"
}
- }
- return add
- if !`keepmc' {
= if !0 {
- return local mcmethod
- return local mctitle
- }
}

> _____ end _coef_table _____
> _____ end _ereturn_display _____

- mat `b2' = e(b)
= mat __000009 = e(b)
- drop `xb'
= drop __000006
- _predict double `xb' `if'
= _predict double __000006 if __000002
- LikePois `y' `xb' `poff' [`weight' `exp'] `if', n(`nobs')
= LikePois y_timing_max __000006 [] if __000002, n(__000004)

> _____ begin poisson.LikePois _____

- gettoken y 0 : 0
- gettoken xb 0 : 0
- syntax [fw aw pw iw] [if] , Nobs(string)
- if "`weight'"!="" {
= if ""!="" {
  if "`weight'"=="fweight" {
    local wt `[fw`exp']"
  }
  else if "`weight'"=="iweight" {

```

```

        local wt `[iw`exp']"'
    }
    else local wt `[aw`exp']"'
    }
    - tempvar lnf
    - qui gen double `lnf' = `y'*(`xb')-exp(`xb')-lngamma(`y'+1) `if'
    = qui gen double __00000D = y_timing_max*(__000006)-exp(__000006)-
> lngamma(y_timing_max+1) if __000002
    - summarize `lnf' `wt' `if', meanonly
    = summarize __00000D if __000002, meanonly
    - if r(N) != `nobs' { exit }
    = if r(N) != __000004 { exit }
    - if "`weight'"=="aweight" {
    = if ""=="aweight" {
        ret scalar lnf = r(N)*r(mean)
    }
    - else ret scalar lnf = r(sum)
    }

> ----- end poisson.LikePois -----

    - tempname lnf2
    - scalar `lnf2' = r(lnf)
    = scalar __00000D = r(lnf)
    - }
    - if _rc { local lnf2 . }
    - if `lnf1'>=. & `lnf1c'>=. & `lnf2'>=. { exit }
    = if __00000A>=. & __00000C>=. & __00000D>=. { exit }
    - if `lnf2'<.&(`lnf2'>`lnf1'|`lnf1'>=.)&(`lnf2'>`lnf1c'|`lnf1c'>=.)
> {
    = if __00000D<.&(__00000D>__00000A|__00000A>=.)&(__00000D>__00000C|_
> __00000C>=.) {
    - ret matrix b0 `b2'
    = ret matrix b0 __000009
    - exit
    }

> ----- end poisson.Ipois -----

    - if "`r(b0)'"!="" {
    = if "matrix"!="" {
    - tempname from
    - mat `from' = r(b0)
    = mat __000006 = r(b0)
    - }
    - }
    - if "`from'"!="" {
    = if "__000006"!="" {
    - local initopt `init(`from') search(off)'"
    = local initopt `init(__000006) search(off)'"

```

```

- }
- else local initopt "search(on) maxfeas(50)"
- `vv' ml model `mm' `eval' (`yname': `y' = `xvars', `constan' `offopt
> ') [`weight'`exp'] if `touse', collinear missing max nooutput nopreserve `ml
> opts' title(Poisson regression) `scopt' `robust' `clopt' `initopt' `ll0' `ne
> gh' `moptobj'
= version 18, missing: ml model e2 poiss_lf (y_timing_max: y_timing_ma
> x = cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calv
> o ln_neq open, ) [] if __000002, collinear missing max nooutput nopreserve
> title(Poisson regression) robust init(__000006) search(off) lf0(1 -437.38
> 5284075989) negh

```

---

```

> ----- begin ml -----
- if _caller()>=11 {
- mopt `0'
= mopt model e2 poiss_lf (y_timing_max: y_timing_max = cb_authors_ex
> t estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open, )
> [] if __000002, collinear missing max nooutput nopreserve title(Poisson re
> gression) robust init(__000006) search(off) lf0(1 -437.385284075989) negh

```

---

```

> ----- begin mopt -----
- local vv : di "version " string(_caller()) ":"
- version 12
- gettoken subcmd 0 : 0, parse(" ,")
- local lcmd : length local subcmd
- if "`subcmd'" == "clear" {
= if "model" == "clear" {
Clear
exit
}
- if ``"subcmd"' == "hold" {
= if `"model"' == "hold" {
ml_hold `0'
exit
}
- if ``"subcmd"' == "unhold" {
= if `"model"' == "unhold" {
ml_unhold `0'
exit
}
- if "`subcmd'" == "score" {
= if "model" == "score" {
if _caller() > 12 {
`vv' Score `0'
}
else {

```

```

Score `0'
}
exit
}
- if "`subcmd'" == "count" {
= if "model" == "count" {
Count `0'
exit
}
- if "`subcmd'" == "trace" {
= if "model" == "trace" {
Trace `0'
exit
}
- if "`subcmd'" == bsubstr("graph",1,max(2,`lcmd')) {
= if "model" == bsubstr("graph",1,max(2,5)) {
Graph `0'
exit
}
- if "`subcmd'" == "init" {
= if "model" == "init" {
Interactive 0
`vv' Init `0'
exit
}
- if "`subcmd'" == bsubstr("maximize",1,max(3,`lcmd')) {
= if "model" == bsubstr("maximize",1,max(3,5)) {
Interactive 1
`vv' MaxMin maximize `0'
exit
}
- if "`subcmd'" == bsubstr("minimize",1,max(3,`lcmd')) {
= if "model" == bsubstr("minimize",1,max(3,5)) {
Interactive 1
`vv' MaxMin minimize `0'
exit
}
- if "`subcmd'" == bsubstr("display",1,max(2,`lcmd')) {
= if "model" == bsubstr("display",1,max(2,5)) {
Display `0'
exit
}
- if "`subcmd'" == bsubstr("footnote",1,max(4,`lcmd')) {
= if "model" == bsubstr("footnote",1,max(4,5)) {
ml_footnote `0'
exit
}

```



```

    }
    - if "`subcmd'" == "," {
= if "model" == "," {
    Display, `0'
    exit
    }
    - if "`subcmd'" == bsubstr("model",1,max(3,`lcmd')) {
= if "model" == bsubstr("model",1,max(3,5)) {
    - `vv' capture noi Model `0'
    = version 18: capture noi Model e2 poiss_lf (y_timing_max: y_timi
> ng_max = cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr
> _calvo ln_neq open, ) [] if __000002, collinear missing max nooutput nopres
> erve title(Poisson regression) robust init(__000006) search(off) lf0(1 -4
> 37.385284075989) negh

> ----- begin mopt.Model -----
    - local version = string(_caller())
    - local vv : di "version `version':"
    = local vv : di "version 18:"
    - version 12
    - syntax anything(id="model" name=spec) [if] [in] [fw aw pw iw]
> [, BRACKET COLLinear MISSing CONTinue lf0(string) INIT(string) CONSTraints(s
> tring) noCNSNOTEs VCE(passthru) NEGh NOPREserve TECHnique(string) ITERID(str
> ing) CRITtype(string) OBS(numlist max=1 >0) TITle(string) WALDtest(integer -
> 1) SVY SUBpop(string) SRSubpop noSVYadjust SCore(passthru) noOUTput noSCVAR
> S GROUP(varname) ITERPROLOG(string) DERIVPROLOG(string) DERIVH(string) DERIV
> SCALE(string) DERIV(string) nrderiv GNWMATrix(string) KAUXiliary(numlist int
> >=0) MAXimize MINimize USERinfo(namelist) PUPDATED Robust CLuster(passthru)
> moptobj(string) USEVIEWS * ]
    - if ``moptobj' == "" {
    = if `""' == "" {
    - Clear

> ----- begin mopt.Clear -----
    - syntax [, NOTANOPTION]
    - capture drop $ML_w
    = capture drop
    - capture drop $ML_samp
    = capture drop
    - capture drop $ML_sample
    = capture drop
    - capture drop $ML_subv
    = capture drop
    - capture drop $ML_grp
    = capture drop
    - mata: Mopt_drop_external()

```

```

- macro drop ML_*


---


> 

---

 end mopt.Clear 

---


- }
- local maxmin `minimize' `maximize'
= local maxmin maximize
- opts_exclusive ``maxmin''
= opts_exclusive "maximize"


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts maximize
- local n 0
- while ``opts'' != "" {
= while `"maximize"' != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'' != "" {
= while `"' != "" {
- local ++n
- gettoken item`n' opts : opts, bind
- }
- if `n' < 2 {
= if 1 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- if `:length local maxmin' {
= if 8 {
- local interactive 0
- }
- else local interactive 1
- _get_diparmopts, diparmopts(`options') soptions syntaxonly
= _get_diparmopts, diparmopts(search(off)) soptions syntaxonly


---


> 

---

 begin _get_diparmopts 

---


- if _caller() < 8.2 {
- _get_diparmopts_8 `0'
- exit
- }
- version 9
- local version : di "version " string(_caller()) ":"

```

```

- syntax [, diparmopts(string asis) EXECute soptions bottom pl
> us SYNTAXonly * ]
- local goptions `options'
= local goptions
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk ,

> ----- begin _get_diparmopts.SpliceGlobals -----
- syntax name(id="macro name" name=c_diparm) [, diparm(strin
> g asis) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof''' == "" & ``e(df_r)''' != "" {
= if `''' == "" & `212''' != "" {
- capture confirm integer number `e(df_r)''
= capture confirm integer number 212
- if !c(rc) {
- local gdof dof(`e(df_r)')
= local gdof dof(212)
- }
- capture
- }
- else local gdof ``dof'''
= else local gdof `'''
- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)
- if (``diparm''' == "") exit
= if (``''' == "") exit

> ----- end _get_diparmopts.SpliceGlobals -----
- local 0 `", `diparmopts'''
= local 0 `", search(off)'''
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while ``diparm''' != "" {
= while `''' != "" {
local ++k
local single 1
local after
capture _on_colon_parse `diparm'
if !c(rc) {
local single 0
local diparm ``s(before)'''
local after ``s(after)'''
}

```

```

capture
SpliceGlobals diparm , diparm(`diparm') `syntaxonly' `goptio

> ns'

local diparm`k' ```diparm'"""
local diparmopts ```diparmopts' : `diparm'"""
if `single' | ```:list retok after'""" == "" {
local 0 `", `options'"""
syntax [, diparm(string asis) * ]
}
else {
local diparm ```after'"""
}
}
- if ```soptions'""" == "" {
= if ``soptions'""" == "" {
syntax [, diparm(string asis) ]
}
- if ("`execute'"" == "") {
= if ("" == "") {
- sreturn clear
- sreturn local options ```options'"""
= sreturn local options ``search(off)"""
- sreturn local diparm ```diparmopts'"""
= sreturn local diparm ``"""
- forval i = 1/`k' {
= forval i = 1/0 {
sreturn local diparm`i' ```diparm`i'"""
}
- sreturn local k `k'
= sreturn local k 0
- exit
}



---


> 

---

 end _get_diparmopts 

---



- forval k = 1/`s(k)' {
= forval k = 1/0 {
local diparm`k' ```s(diparm`k')'"""
}
- local ndiparm `s(k)'
= local ndiparm 0
- local options ```s(options)'"""
= local options ``search(off)"""
- if `interactive' & `:length local options' {
= if 0 & 11 {
local 0 , `options'
syntax [, NOTANOPTION]
}
}

```

```

- if `:length local nopreserve' | `:length local score' {
= if 10 | 0 {
- global ML_preserve no
- }
- else global ML_preserve yes
- gettoken evaltype spec : spec
- confirm name `evaltype'
= confirm name e2
- gettoken evaluator spec : spec
- global ML_evaltype : copy local evaltype
- if `:length local srssubpop' {
= if 0 {
  local 0 , srssubpop
  syntax [, NONOPTION]
  exit 198
}
- if `:length local subpop' {
= if 0 {
  local svy svy
}
- if `:length local svy' {
= if 0 {
  opts_exclusive "`vce' `svy'" vce
  if `:length local weight' {
    di as err "svy option not allowed with weights"
    exit 198
  }
  global ML_noadj : copy local svyadjust
}
- global ML_svy : copy local svy
- if _caller() >= 15 & c(userversion) >= 15 {
- local fparm ", freeparm"
- }
- local eqspec : copy local spec
- local ieq 0
- while `:length local eqspec' {
= while 123 {
- local ++ieq
- gettoken eq eqspec : eqspec, bind
- if bsubstr(trim("`eq'"),1,1) == "/" {
= if bsubstr(trim("(y_timing_max: y_timing_max = cb_authors_ext
> estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )")
> ,1,1) == "/" {
  local eq = bsubstr(trim("`eq'"),2,.)
  local spec : substr local spec "/"`eq' " ("`eq'":`fparm')"' ,
> word

```

```

    }
    - else {
    - gettoken p1 p2 : eq, parse("()")
    - local p2 = trim("`p2'")
    = local p2 = trim("`y_timing_max: y_timing_max = cb_authors_ext
> estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )"
> )
    - if bsubstr("`p2'",1,1) == "," {
    = if bsubstr("y_timing_max: y_timing_max = cb_authors_ext estima
> ted wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )",1,1) =
> = "," {
        gettoken comma p2 : p2, parse(",")
        local p2 = trim("`p2'")
    }
    - if "`p1'`p2'" == "()" {
    = if `(y_timing_max: y_timing_max = cb_authors_ext estimated wg
> _ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )" == "()" {
        local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
    }
    - }
    - }
    - while `:length local eqspec' {
    = while 3 {
    - local ++ieq
    - gettoken eq eqspec : eqspec, bind
    - if bsubstr(trim("`eq'"),1,1) == "/" {
    = if bsubstr(trim("[]"),1,1) == "/" {
        local eq = bsubstr(trim("`eq'"),2,.)
        local spec : subinstr local spec "/"`eq'" `("`eq':`fparm')"'
> word
    }
    - else {
    - gettoken p1 p2 : eq, parse("()")
    - local p2 = trim("`p2'")
    = local p2 = trim("`")
    - if bsubstr("`p2'",1,1) == "," {
    = if bsubstr("",1,1) == "," {
        gettoken comma p2 : p2, parse(",")
        local p2 = trim("`p2'")
    }
    - if "`p1'`p2'" == "()" {
    = if `[]" == "()" {
        local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
    }
    - }
    - }

```

```

- while `:length local eqspec' {
= while 0 {
  local ++ieq
  gettoken eq eqspec : eqspec, bind
  if bsubstr(trim("`eq'"),1,1) == "/" {
    local eq = bsubstr(trim("`eq'"),2,.)
    local spec : subinstr local spec "/"`eq' " ("`eq':`fparm')"' ,
> word
  }
  else {
    gettoken p1 p2 : eq, parse("(")
    local p2 = trim("`p2'")
    if bsubstr("`p2'",1,1) == "," {
      gettoken comma p2 : p2, parse(",")
      local p2 = trim("`p2'")
    }
    if "`p1'`p2'" == "(" {
      local spec : subinstr local spec "`eq' "(eq`ieq:)", word
    }
  }
}
- if `interactive' {
= if 0 {
  SetGlobalVar ML_sample : _MLtua
}
- else {
- global ML_sample : tempvar
- if ``moptobj' != "" {
= if `"' != "" {
  global ML_sample `moptobj'$ML_sample
}
- }
- if ``moptobj' != "" {
= if `"' != "" {
  cap drop $ML_sample
}
- `vv' mark $ML_sample `if' `in' [`weight'`exp'], `zeroweight'
= version 18: mark __000007 if __000002 [],
- tempname eqlist
- `.eqlist' = _eqlist.new, eqopts(NOCONSTANT FREEParam) eqargopt
> s(OFFset EXPosure) noneedvarlist numdepvars(0) needequal
  = __000008 = _eqlist.new, eqopts(NOCONSTANT FREEParam) eqargopt
> s(OFFset EXPosure) noneedvarlist numdepvars(0) needequal
  -class {
  -instance:
  -      array          n_depvars

```

```

-      string      needequal
-      string      noneedvarlist
-      string      wtypes
-      string      collinear
-      string      nocons
-      string      nocons_ignore
-      double      n_eq
-      string      gl_if
-      string      gl_in
-      string      gl_wgt
-      string      gl_options
-      _optlist     eq_opts_allowed
-class {
-instance:
-      array options = {}
-      array names   = {}
-}
-      array      eq_list
-      _optlist    mark_opts
-      array      mark_list
-      array      mark_name
-      _optlist    smark_opts
-      array      smark_list
-      array      smark_name
-      _optlist    rmcoll_opts
-      array      rmcoll_list
-      array      rmcoll_name
-      _optlist    rmdcoll_opts
-      array      rmdcoll_list
-      array      rmdcoll_name
-}

```

```

> ----- begin _eqlist.new -----
- version 9.2
- ._init
-----
> ----- begin _eqlist._init -----
- version 9.2
- .n_eq = 0
- .gl_if = ""
- .gl_in = ""
- .gl_wgt = ""
- .gl_options = ""
- .eq_list = {}
-----
> ----- end _eqlist._init -----

```



```

- ._setup `0'
= ._setup , eqopts(NOCONSTant FREEParam) eqargopts(OFFset EXPos
> ure) noneedvarlist numdepvars(0) needequal

> ----- begin _eqlist._setup -----
- version 9.2
- .mark_list = {}
- .mark_name = {}
- .mark_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .smark_list = {}
- .smark_name = {}
- .smark_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .rmcoll_list = {}
- .rmcoll_name = {}
- .rmcoll_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .rmdcoll_list = {}
- .rmdcoll_name = {}
- .rmdcoll_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----

```

```

- syntax [, EQopts(namelist) EQARGopts(namelist) noNEEDVARLI
> ST COMMONopts(string asis) MARKopts(namelist) SMARKopts(namelist) RMCOLLOpts
> (namelist) RMDCOLLOpts(namelist) NUMDEPvars(numlist integer >=0) NEEDEQUAL W
> TYPES(string) IGNORENOCONS ]
- .eq_opts_allowed.reset
_____
> _____ begin _optlist.reset _____
- version 9
- .options = {}
- .names = {}
_____
> _____ end _optlist.reset _____
- foreach opt of local eqopts {
- .eq_opts_allowed.addopt `opt'
= .eq_opts_allowed.addopt NOCONStant
_____
> _____ begin _optlist.addopt _____
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("NOCONStant")
- if "`passthru'" != "" {
= if "" != "" {
local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "NOCONStant"
- .names.Arrpush "`mname'"
= .names.Arrpush "noconstant"
_____
> _____ end _optlist.addopt _____
- }
- .eq_opts_allowed.addopt `opt'
= .eq_opts_allowed.addopt FREEParm
_____
> _____ begin _optlist.addopt _____
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("FREEParm")
- if "`passthru'" != "" {
= if "" != "" {
local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "FREEParm"

```

```

- .names.Arrpush "`mname'"
= .names.Arrpush "freeparm"
> ----- end _optlist.addopt -----
- }
- foreach opt of local eqargopts {
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt OFFset, passthru
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OFFset")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec OFFset(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "OFFset(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "offset"
> ----- end _optlist.addopt -----
- }
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt EXPosure, passthru
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("EXPosure")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec EXPosure(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "EXPosure(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "exposure"
> ----- end _optlist.addopt -----
- }

```

```

- local i 0
- foreach opt of local markopts {
  .mark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local smarkopts {
  .smark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmcollopts {
  .rmcoll_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmdcollopts {
  .rmdcoll_opts.addopt `opt', passthru
}
- .n_depvars.Arrdropall
- if "`numdepvars'" != "" {
= if "0" != "" {
- foreach ndep of local numdepvars {
- .n_depvars.Arrpush `ndep'
= .n_depvars.Arrpush 0
- }
- }
- else .n_depvars.Arrpush 1
- .needequal = "`needequal'"
= .needequal = "needequal"
- .wtypes = "`wtypes'"
= .wtypes = ""
- .noneedvarlist = "`needvarlist'"
= .noneedvarlist = "noneedvarlist"
- .nocons_ignore = "`ignorenocons'"
= .nocons_ignore = ""
}

> _____ end _eqlist._setup _____
> _____ end _eqlist.new _____
- .`eqlist'.parse `spec'
= .__000008.parse (y_timing_max: y_timing_max = cb_authors_ext
> estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open, ) [
> ]

> _____ begin _eqlist.parse _____
- version 9.2
- .__init

```

```

> ----- begin _eqlist._init -----
    - version 9.2
    - .n_eq = 0
    - .gl_if = ""
    - .gl_in = ""
    - .gl_wgt = ""
    - .gl_options = ""
    - .eq_list = {}
    -----
> ----- end _eqlist._init -----
    - local MARKOPTLIST ```.mark_opts.dumpoptions'''
    -----
> ----- begin _optlist.dumpoptions -----
    - version 9
    - forval i = 1/`.options.arnels' {
    = forval i = 1/0 {
        local optlist `optlist' `.options['i']'
    }
    - local optlist : list uniq optlist
    - sreturn local optlist ```.optlist'''
    = sreturn local optlist `'''
    - class exit ```.optlist'''
    = class exit `'''
    -----
> ----- end _optlist.dumpoptions -----
    = local MARKOPTLIST `'''
    - local SMARKOPTLIST ```.smark_opts.dumpoptions'''
    -----
> ----- begin _optlist.dumpoptions -----
    - version 9
    - forval i = 1/`.options.arnels' {
    = forval i = 1/0 {
        local optlist `optlist' `.options['i']'
    }
    - local optlist : list uniq optlist
    - sreturn local optlist ```.optlist'''
    = sreturn local optlist `'''
    - class exit ```.optlist'''
    = class exit `'''
    -----
> ----- end _optlist.dumpoptions -----
    = local SMARKOPTLIST `'''
    - local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''
    -----
> ----- begin _optlist.dumpoptions -----
    - version 9

```

```

- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']'
    }
- local optlist : list uniq optlist
- sreturn local optlist `"'optlist'"
= sreturn local optlist `""'
- class exit `"'optlist'"
= class exit `""'

> ----- end _optlist.dumpoptions -----
= local RMCOLLOPTLIST `""'
- local RMDCOLLOPTLIST `"'`rmcoll_opts.dumpoptions'"

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']'
    }
- local optlist : list uniq optlist
- sreturn local optlist `"'optlist'"
= sreturn local optlist `""'
- class exit `"'optlist'"
= class exit `""'

> ----- end _optlist.dumpoptions -----
= local RMDCOLLOPTLIST `""'
- _parse expand EQ GL : 0, gweight common( `MARKOPTLIST' `SMAR
> KOPTLIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
= _parse expand EQ GL : 0, gweight common( COLLlinear )
- if `EQ_n' == 0 & "`noneedvarlist'" == "" {
= if 1 == 0 & "noneedvarlist" == "" {
    syntax varlist
    }
- local rebind = `EQ_n' == 1
= local rebind = 1 == 1
- if `rebind' {
= if 1 {
- tokenize `"'0'"', parse(",()[]")
= tokenize `"(y_timing_max: y_timing_max = cb_authors_ext esti
> mated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open, ) []"',
> parse(",()[]")
- local rebind = `"'1'"' != "(" & `"'2'"' == "("
= local rebind = `"' != "(" & `y_timing_max: y_timing_max =
> cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo l

```

```

> n_neq open''' == "("
- }
- if `rebind' {
= if 0 {
  while `:length local 0' {
    gettoken tok 0 : 0, parse(" ,") bind match(par)
    if "`par'" == "(" {
      gettoken nxt : 0
      local nxt = substr("`nxt'",1,1)
      if inlist("`nxt'", "#", "|") {
        local ZERO ``ZERO' (`tok)''
      }
    }
    else {
      local ZERO ``(`ZERO') (`tok)' `0'''
      local 0
    }
  }
  else if inlist("`tok'", "if", "in", ",") {
    local ZERO ``(`ZERO') `tok' `0'''
    local 0
  }
  else if inlist(substr("`tok'",1,1), "#", "|") {
    local ZERO ``ZERO'`tok'''
  }
  else {
    local ZERO ``ZERO' `tok'''
  }
}
_parse expand EQ GL : ZERO, gweight common( `MARKOPTLIST' `S
> MARKOPTLIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
}
- forval i = 1/`EQ_n' {
= forval i = 1/1 {
- if `i' <= `.n_depvars.arnels' {
= if 1 <= 1 {
- local ndep = `.n_depvars[`i]''
= local ndep = 0
- }
- .eq_list[`i'] = .eqspec.new, numdepvars(`ndep') `.needequal
> ' `.noneedvarlist' wtypes(`.wtypes') `.nocons_ignore'
= .eq_list[1] = .eqspec.new, numdepvars(0) needequal noneedva
> rlist wtypes()
-class {
-instance:
- double n_depvars
- double needequal

```

```

-      double  needvarlist
-      string  wtypes
-      string  name
-      string  depvars
-      double  hasequal
-      string  indepvars
-      string  other_markvars
-      double  nocons
-      double  nocons_ignore
-      string  offset
-      double  exposure
-      double  freeparm
-      string  if
-      string  in
-      string  wgt
-      string  options
-}

> ----- begin _eqspec.new -----
- version 9.2
- syntax [, NUMDEPvars(numlist integer >=0 max=1) NEEDEQUAL
> noNEEDVARLIST WTYPES(string) IGNORENOCONS ]
- .n_depvars = 1
- if `:list sizeof numdepvars' {
= if 1 {
- .n_depvars = `numdepvars'
= .n_depvars = 0
- }
- .needequal = ("`needequal'" != "")
= .needequal = ("needequal" != "")
- .needvarlist = ("`needvarlist'" == "")
= .needvarlist = ("noneedvarlist" == "")
- .nocons_ignore = ("`ignorenocons'" != "")
= .nocons_ignore = (" " != "")
- .wtypes = "`wtypes'"
= .wtypes = ""

> ----- end _eqspec.new -----
- .eq_list['i'].parse ` .eq_opts_allowed.objkey':`EQ_`i'
= .eq_list[1].parse __key(600003e89400):y_timing_max: y_timing
> _max = cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_c
> alvo ln_neq open

> ----- begin _eqspec.parse -----
- version 9.2
- _on_colon_parse `0'

```



```

= _on_colon_parse __key(600003e89400):y_timing_max: y_timing
> _max = cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_c
> alvo ln_neq open

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) qu
> otes
- if "`par'" != "" {
= if "" != "" {
  local before `(`before')'
}
- if ``before'"" == ":" {
= if `__key(600003e89400)' == ":" {
  sreturn local after ``after'""
  exit
}
- while ``COLON'"" != ":" & ``after'"" != "" {
= while `"" != ":" & `:y_timing_max: y_timing_max = cb_a
> uthors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq
> open"" != "" {
- gettoken COLON after : after, parse(":") bind match(par)
> quotes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``COLON'"" != ":" {
= else if `:"' != ":" {
  local before ``before' `COLON'""
  local COLON
}
- }
- while ``COLON'"" != ":" & ``after'"" != "" {
= while `:"' != ":" & `y_timing_max: y_timing_max = cb_a
> uthors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq
> open"" != "" {
  gettoken COLON after : after, parse(":") bind match(par)
> quotes
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }

```

```

        else if `""`COLON`"" != ":" {
            local before `""`before' `COLON`""
            local COLON
        }
    }
- if `""`COLON`"" != ":" {
= if `":`"" != ":" {
    di as err ""' found where ':' expected"
    exit 198
}
- sreturn local before `""`before`""
= sreturn local before `""__key(600003e89400)`""
- sreturn local after `""`after`""
= sreturn local after `""y_timing_max: y_timing_max = cb_au
> thors_ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq
> open`""

> ----- end _on_colon_parse -----
- local ol `s(before)`
= local ol __key(600003e89400)
- if `:list sizeof ol' {
= if 1 {
- local OPTLIST `""`.ol'.dumpoptions`""

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/4 {
- local optlist `optlist' `.options['i']`
= local optlist NOCONSTant
- }
- local optlist `optlist' `.options['i']`
= local optlist NOCONSTant FREEParam
- }
- local optlist `optlist' `.options['i']`
= local optlist NOCONSTant FREEParam OFFSET(passthru)
- }
- local optlist `optlist' `.options['i']`
= local optlist NOCONSTant FREEParam OFFSET(passthru) EXPos
> ure(passthru)
- }
- local optlist : list uniq optlist
- sreturn local optlist `""`optlist`""
= sreturn local optlist `""NOCONSTant FREEParam OFFSET(passthru) EXPos
> hru) EXPosure(passthru)`""
- class exit `""`optlist`""

```

```

= class exit ``NOCONSTant FREEParam OFFset(passthru) EXPosu
> re(passthru)''
_____
> _____ end _optlist.dumpoptions _____
= local OPTLIST ``NOCONSTant FREEParam OFFset(passthru) EXPos
> ure(passthru)''
- local OPTNAMES ```.ol'.dumpnames'''
_____
> _____ begin _optlist.dumpnames _____
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/4 {
- local namelist `namelist' `.names['i']'
= local namelist noconstant
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset exposure
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist ``noconstant freeparm offset expo
> sure''
- class exit ```namelist'''
= class exit ``noconstant freeparm offset exposure''
_____
> _____ end _optlist.dumpnames _____
= local OPTNAMES ``noconstant freeparm offset exposure''
- local NOCONS constant noconstant
- if ```:list NOCONS & OPTNAMES''' == '' {
= if ``noconstant'' == '' {
local NOCONS noCONSTant
}
- else local NOCONS
- }
- local 0 ``s(after)'''
= local 0 ``y_timing_max: y_timing_max = cb_authors_ext esti
> mated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open''
- if ```.wtypes''' != '' {
= if `''' != '' {
local wgt "[`.wtypes']"

```



```

- if (! c(rc)) exit

```

---

```

> end _eqspec._confirm_eqname
- }
- local NAME : list retok NAME
- gettoken COLON anything : REST, parse(":")
- }
- else {
  local NAME
}
- gettoken DEPVARs REST : anything, parse("=") bind
- if "`DEPVARs'" == "=" {
= if "y_timing_max " == "=" {
  local REST : copy local anything
  local DEPVARs
}
- if (:length local DEPVARs' == 0 & `.n_depvars') {
= if (13 == 0 & 0) {
  di as err "nothing found where depvar expected"
  exit 198
}
- if ``"REST"" == "" {
= if ``= cb_authors_ext estimated wg_ndx_mult stky_pr_rotemb
> erg wlth stky_pr_calvo ln_neq open"" == "" {
  if 0`.needequal' {
  if `.needvarlist' | (:length local DEPVARs' != 0 & `.n_de
> pvars') {
  di as err "equal sign required to separate depvars from in
> depvars"
  exit 198
}
}
local DEPVARs
local n_deps : list sizeof anything
if `n_deps' != 0 | `.needvarlist' {
if `n_deps' < `.n_depvars' {
error 102
}
forval j = 1/`.n_depvars' {
gettoken YY anything : anything
local DEPVARs `DEPVARs' `YY'
}
local INDEPVARs : list retok anything
}
loca hasequal 0
}

```

```

- else {
- gettoken EQ INDEPVARs : REST, parse("=") bind
- local hasequal 1
- }
- if `:list sizeof DEPVARs' {
= if 1 {
- _fv_check_depvar `DEPVARs'
= _fv_check_depvar y_timing_max
- tsunab DEPVARs : `DEPVARs'
= tsunab DEPVARs : y_timing_max
}

> ----- begin tsunab -----
- version 6
- gettoken user 0: 0, parse(" :")
- gettoken colon 0: 0, parse(" :")
- if `"'colon'" != ":" { error 198 }
= if `":'" != ":" { error 198 }
- syntax [varlist(ts default=empty)] [, MIN(integer 1) MAX
> (integer 32000) NAME(string)]
- c_local `user' `varlist'
= c_local DEPVARs y_timing_max
- local n : word count `varlist'
= local n : word count y_timing_max
- if `n'>=`min' & `n'<=`max' { exit }
= if 1>=1 & 1<=32000 { exit }

> ----- end tsunab -----
- }
- if `:list sizeof INDEPVARs' {
= if 8 {
- fvunab INDEPVARs : `INDEPVARs'
= fvunab INDEPVARs : cb_authors_ext estimated wg_ndx_mult s
> tky_pr_rottemberg wlth stky_pr_calvo ln_neq open

> ----- begin fvunab -----
- version 11
- _on_colon_parse `0'
= _on_colon_parse INDEPVARs : cb_authors_ext estimated wg
> _ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par)

> quotes

```

```

- if "`par'" != "" {
= if "" != "" {
  local before `("`before')"'
}
- if ``"before"' == ":" {
= if `"INDEPVARs "' == ":" {
  sreturn local after ``"after"'
  exit
}
- while ``"COLON"' != ":" & ``"after"' != "" {
= while `"' != ":" & `":' cb_authors_ext estimated wg_n
> dx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open"' != "" {
- gettoken COLON after : after, parse(":") bind match(pa
> r) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``"COLON"' != ":" {
= else if `":' != ":" {
  local before ``"before' `COLON"'
  local COLON
}
- }
- while ``"COLON"' != ":" & ``"after"' != "" {
= while `":' != ":" & `":' cb_authors_ext estimated wg_n
> dx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open"' != "" {
  gettoken COLON after : after, parse(":") bind match(pa
> r) quotes
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``"COLON"' != ":" {
    local before ``"before' `COLON"'
    local COLON
  }
}
- if ``"COLON"' != ":" {
= if `":' != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before ``"before"'
= sreturn local before `"INDEPVARs "'

```

```

        - sreturn local after ```after'``
        = sreturn local after ``  cb_authors_ext estimated wg_nd
> x_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open"

> ----- end _on_colon_parse -----
        - local 0 ```s(before)``
        = local 0 ``INDEPVARs ``
        - syntax name(id="macro name" name=user)
        - local 0 ```s(after)``
        = local 0 ``  cb_authors_ext estimated wg_ndx_mult stky_pr
> _rottemberg wlth stky_pr_calvo ln_neq open"
        - syntax [varlist(ts fv default=none)] [, MIN(integer 1) M
> AX(integer 32000) NAME(string) ]
        - c_local `user' `varlist'
        = c_local INDEPVARs cb_authors_ext estimated wg_ndx_mult s
> tky_pr_rottemberg wlth stky_pr_calvo ln_neq open
        - local n : list sizeof varlist
        - if (`min' <= `n' & `n' <= `max') exit
        = if (1 <= 8 & 8 <= 32000) exit

> ----- end fvunab -----
        - }
        - if inlist("noconstant", ``noconstant``, ``constant``) {
        = if inlist("noconstant", "", "") {
            if ``INDEPVARs`` == "" & !`.nocons_ignore' {
                di as err "independent variables are required when option
> noconstant is specified"
                exit 102
            }
            if ``NOCONS`` != "" {
                local options `options' noconstant
            }
            local NOCONS 1
        }
        - else local NOCONS 0
        - if ```offset'`exposure'`` != "" {
        = if ```` != "" {
            . _check_offset, `offset' `exposure'
            local OFFSET ``s(offset)``
            local EXPOSURE = (`exposure' != "")
            local OTHER `s(offset)'
        }
        - else local EXPOSURE 0
        - local FREEARM = ```freeparm'`` != ""
        = local FREEARM = ```` != ""
        - local OPTIONS

```



```

- local SPACE
- foreach OPTNAME of local OPTNAMES {
- if `""`OPTNAME`""` != "" {
= if `""` != "" {
    local OPTIONS `""`OPTIONS`""`SPACE`""`OPTNAME`""`
    local SPACE " "
    }
- }
- if `""`OPTNAME`""` != "" {
= if `""` != "" {
    local OPTIONS `""`OPTIONS`""`SPACE`""`OPTNAME`""`
    local SPACE " "
    }
- }
- if `""`OPTNAME`""` != "" {
= if `""` != "" {
    local OPTIONS `""`OPTIONS`""`SPACE`""`OPTNAME`""`
    local SPACE " "
    }
- }
- if `""`OPTNAME`""` != "" {
= if `""` != "" {
    local OPTIONS `""`OPTIONS`""`SPACE`""`OPTNAME`""`
    local SPACE " "
    }
- }
- if `""`options`""` != "" {
= if `""` != "" {
    local OPTIONS `""`OPTIONS`""`SPACE`""`options`""`
    }
- .name = `""`NAME`""`
= .name = `""`y_timing_max`""`
- .depvars = `""`DEPVAR`""`
= .depvars = `""`y_timing_max`""`
- .hasequal = `hasequal`
= .hasequal = 1
- .indepvars = `""`:list retok INDEPVAR`""`
= .indepvars = `""`cb_authors_ext estimated wg_ndx_mult stky_p
> r_rotemberg wlth stky_pr_calvo ln_neq open`""`
- .other_markvars = `""`:list retok OTHER`""`
= .other_markvars = `""`
- .nocons = `NOCONS`
= .nocons = 0
- .if = `""`HOLDIF`""`
= .if = `""`
- .in = `""`HOLDIN`""`

```

```

= .in = `""`
- .wgt = `""`HOLDWGT`""`
= .wgt = `""`
- .offset = `""`OFFSET`""`
= .offset = `""`
- .exposure = `EXPOSURE`
= .exposure = 0
- .freeparm = `FREEPARM`
= .freeparm = 0
- .options = `""`OPTIONS`""`
= .options = `""`

> _____ end _eqspec.parse _____

- }
- local MARKOPTLIST `""`.mark_opts.dumpoptions`""`

> _____ begin _optlist.dumpoptions _____

- version 9
- forval i = 1/`.options.arrnels' {
= forval i = 1/0 {
    local optlist `optlist' `'.options['i']`
    }
- local optlist : list uniq optlist
- sreturn local optlist `""`optlist`""`
= sreturn local optlist `""`
- class exit `""`optlist`""`
= class exit `""`

> _____ end _optlist.dumpoptions _____

= local MARKOPTLIST `""`
- local MARKOPTNAMES `""`.mark_opts.dumpnames`""`

> _____ begin _optlist.dumpnames _____

- version 9
- forval i = 1/`.names.arrnels' {
= forval i = 1/0 {
    local namelist `namelist' `'.names['i']`
    }
- local namelist : list uniq namelist
- sreturn local namelist `""`namelist`""`
= sreturn local namelist `""`
- class exit `""`namelist`""`
= class exit `""`

> _____ end _optlist.dumpnames _____

= local MARKOPTNAMES `""`

```

```

- local OPTLIST1 : substr local MARKOPTLIST "(passthru)" "(s
> tring asis)", all
- local SMARKOPTLIST ```.smark_opts.dumpoptions'''
_____
> _____ begin _optlist.dumpoptions _____
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']'
    }
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist `'''
- class exit ```optlist'''
= class exit `'''
_____
> _____ end _optlist.dumpoptions _____
= local SMARKOPTLIST `'''
- local SMARKOPTNAMES ```.smark_opts.dumpnames'''
_____
> _____ begin _optlist.dumpnames _____
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/0 {
    local namelist `namelist' `.names['i']'
    }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist `'''
- class exit ```namelist'''
= class exit `'''
_____
> _____ end _optlist.dumpnames _____
= local SMARKOPTNAMES `'''
- local OPTLIST1s : substr local SMARKOPTLIST "(passthru)" "
> (string asis)", all
- local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''
_____
> _____ begin _optlist.dumpoptions _____
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']'
    }
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''

```

```

        = sreturn local optlist `""`
        - class exit ``optlist``
        = class exit `""`
    _____
> _____ end _optlist.dumpoptions _____
        = local RMCOLLOPTLIST `""`
        - local RMCOLLOPTNAMES ```.rmcoll_opts.dumpnames``
    _____
> _____ begin _optlist.dumpnames _____
        - version 9
        - forval i = 1/`.names.arnels' {
        = forval i = 1/0 {
            local namelist `namelist' `names['i']`
        }
        - local namelist : list uniq namelist
        - sreturn local namelist ``namelist``
        = sreturn local namelist `""`
        - class exit ``namelist``
        = class exit `""`
    _____
> _____ end _optlist.dumpnames _____
        = local RMCOLLOPTNAMES `""`
        - local OPTLIST2 : substr local RMCOLLOPTLIST "(passthru)" "
> (string asis)", all
        - local RMDCOLLOPTLIST ```.rmdcoll_opts.dumpoptions``
    _____
> _____ begin _optlist.dumpoptions _____
        - version 9
        - forval i = 1/`.options.arnels' {
        = forval i = 1/0 {
            local optlist `optlist' `options['i']`
        }
        - local optlist : list uniq optlist
        - sreturn local optlist ``optlist``
        = sreturn local optlist `""`
        - class exit ``optlist``
        = class exit `""`
    _____
> _____ end _optlist.dumpoptions _____
        = local RMDCOLLOPTLIST `""`
        - local RMDCOLLOPTNAMES ```.rmdcoll_opts.dumpnames``
    _____
> _____ begin _optlist.dumpnames _____
        - version 9
        - forval i = 1/`.names.arnels' {
        = forval i = 1/0 {

```

```

        local namelist `namelist' `.names['i']'
    }
    - local namelist : list uniq namelist
    - sreturn local namelist ``namelist'''
    = sreturn local namelist `'''
    - class exit ``namelist'''
    = class exit `'''

```

---

```

> ----- end _optlist.dumpnames -----
    = local RMDCOLLOPTNAMES `'''
    - local OPTLIST3 : substr local RMDCOLLOPTLIST "(passthru)"
> "(string asis)", all
    - local 0 `", `GL_op'''
    = local 0 `", ""
    - syntax [, `OPTLIST1' `OPTLIST1s' `OPTLIST2' `OPTLIST3' COLli
> near noCONSTant * ]
    = syntax [, COLLlinear noCONSTant * ]
    - local GL_op ``options'''
    = local GL_op `'''
    - if ``collinear''' != "" {
    = if `''' != "" {
        local GL_op ``GL_op' `collinear'''
        .collinear = ``collinear'''
    }
    - if ``constant''' != "" {
    = if `''' != "" {
        local GL_op ``GL_op' `constant'''
        .nocons = ``constant'''
    }
    - local I 0
    - foreach OPT of local MARKOPTNAMES {
        if `:list sizeof `OPT'' {
            .mark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes'
> )
            .mark_list[`I'].parse "" : ``OPT''
            .mark_name.Arrpush ``OPT'''
        }
    }
    - local I 0
    - foreach OPT of local SMARKOPTNAMES {
        if `:list sizeof `OPT'' {
            .smark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes'
> ')
            .smark_list[`I'].parse "" : ``OPT''
            .smark_name.Arrpush ``OPT'''
        }
    }

```

```

    }
- local I 0
- foreach OPT of local RMCOLLOPTNAMES {
  if `:list sizeof `OPT'' {
    .rmcoll_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtype

> s')

    .rmcoll_list[`I'].parse "" : ``OPT''
    .rmcoll_name.Arrpush ``OPT''
  }
}
- local I 0
- foreach OPT of local RMDCOLLOPTNAMES {
  if `:list sizeof `OPT'' {
    .rmdcoll_list[`++I'] = .eqspec.new, wtypes(`.wtypes')
    .rmdcoll_list[`I'].parse "" : ``OPT''
    .rmdcoll_name.Arrpush ``OPT''
  }
}
- .n_eq = `EQ_n'
= .n_eq = 1
- .gl_if = ``GL_if'''
= .gl_if = `'''
- .gl_in = ``GL_in'''
= .gl_in = `'''
- .gl_wgt = ``GL_wt'''
= .gl_wgt = ``[]'''
- .gl_options = ``:list retok GL_op'''
= .gl_options = `'''

> ----- end_eqlist.parse -----
- if ``subpop''' != "" & ``missing''' == "" {
= if "" != "" & "missing" == "" {
  if `interactive' {
    SetGlobalVar ML_samp : _MLtu
  }
  else {
    global ML_samp : tempvar
    if ``moptobj''' != "" {
      global ML_samp `moptobj'$ML_samp
    }
  }
  tempvar markuse
  _svy_setup $ML_sample $ML_samp, svy subpop(`subpop')
  quietly gen byte `markuse' = $ML_samp
}
- else global ML_samp $ML_sample

```

```

= else global ML_samp __000007
- if "`collinear'" == "" {
= if "collinear" == "" {
    .`eqlist'.rmcoll $ML_samp, replace `missing'
}
- else if "`missing'" == "" {
= else if "missing" == "" {
    .`eqlist'.markout $ML_samp, replace
}
- if "$ML_samp" != "$ML_sample" {
= if "__000007" != "__000007" {
    `vv' quietly replace $ML_sample = 0 if `markuse' & !$ML_samp
    drop `markuse'
}
- local eq_n = `.`eqlist'.eq count'

```

---

```

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if "`item'" == "count" {
    = if "count" == "count" {
    - sreturn local value ```.n_eq'''
    = sreturn local value ``1'''
    - class exit ```s(value)'''
    = class exit ``1'''

```

---

```

> ----- end _eqlist.eq -----
    = local eq_n = 1
    - global ML_n : copy local eq_n
    - global ML_dim = `.`eqlist'.dim'

```

---

```

> ----- begin _eqlist.dim -----
    - version 9.2
    - syntax
    - local neq = `.eq count'

```

---

```

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if "`item'" == "count" {
    = if "count" == "count" {
    - sreturn local value ```.n_eq'''
    = sreturn local value ``1'''
    - class exit ```s(value)'''
    = class exit ``1'''

```

---

```

> ----- end _eqlist.eq -----
    = local neq = 1
    - forval i = 1/`neq' {
    = forval i = 1/1 {
    - local xvarsi ```.eq indepvars `i'```

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item' == "count" {
    = if "indepvars" == "count" {
      sreturn local value ```.n_eq'''
      class exit ```s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 1 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item indepvars

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, indepvars

> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEP
> vars IF IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
      local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt
> ' `other' `nocons' `offset' `exposure' `freeparm' `options'
    = local item indepvars
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
      error 198
    }
    - c_local `c_item' `item'
    = c_local item indepvars

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```

```



```

        = local value `"cb_authors_ext estimated wg_ndx_mult stky_
> pr_rottemberg wlth stky_pr_calvo ln_neq open"
        - sreturn local value `""value""
        = sreturn local value `"cb_authors_ext estimated wg_ndx_mu
> lt stky_pr_rottemberg wlth stky_pr_calvo ln_neq open"
        - class exit `""value""
        = class exit `"cb_authors_ext estimated wg_ndx_mult stky_p
> r_rottemberg wlth stky_pr_calvo ln_neq open"

> ----- end _eqspec.item -----
        - class exit `""s(value)""
        = class exit `"cb_authors_ext estimated wg_ndx_mult stky_pr_
> rottemberg wlth stky_pr_calvo ln_neq open"

> ----- end _eqlist.eq -----
        = local xvarsi `"cb_authors_ext estimated wg_ndx_mult stky_pr_
> rottemberg wlth stky_pr_calvo ln_neq open"
        - local noconsi `""eq nocons `i""

> ----- begin _eqlist.eq -----
        - version 9.2
        - args item idx
        - if ""item"" == "count" {
        = if "nocons" == "count" {
            sreturn local value `""n_eq""
            class exit `""s(value)""
        }
        - else if 0 < `idx' & `idx' <= `n_eq' {
        = else if 0 < 1 & 1 <= 1 {
            - .eq_list[`idx'].item `item'
            = .eq_list[1].item nocons

> ----- begin _eqspec.item -----
        - version 9.2
        - args item
        - ._check_eq_item_name item, `item'
        = ._check_eq_item_name item, nocons

> ----- begin _eqspec._check_eq_item_name -----
        - version 9.2
        - capture syntax name(name=c_item) [, NAME DEPvars INDEP
> vars IF IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTIONS ]
        - if c(rc) error 198
        - if `:list sizeof other' {
        = if 0 {
            local other other_markvars

```

```

        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt
> ' `other' `nocons' `offset' `exposure' `freeparm' `options'
        = local item      nocons
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item nocons
    }

> ----- end _eqspec._check_eq_item_name -----
        - local value ```.`item'"""
        = local value `0""
        - sreturn local value ``value'"""
        = sreturn local value `0""
        - class exit ``value'"""
        = class exit `0""
    }

> ----- end _eqspec.item -----
        - class exit ``s(value)"""
        = class exit `0""
    }

> ----- end _eqlist.eq -----
        = local noconsi `0""
        - local dim = `dim' + `:list sizeof xvarsi'
        = local dim = + 8
        - if ``noconsi'"" == "0" {
        = if "0" == "0" {
            - local ++dim
            - }
            - }
            - sreturn local value ``dim'"""
            = sreturn local value `9""
            - class exit ``s(value)"""
            = class exit `9""
        }

> ----- end _eqlist.dim -----
        = global ML_dim = 9
        - forval i = 1/`eq_n' {
        = forval i = 1/1 {
            - local depvars `.`eqlist'.eq depvars `i'
        }

> ----- begin _eqlist.eq -----
        - version 9.2
        - args item idx

```

```

- if "`item'" == "count" {
= if "depvars" == "count" {
  sreturn local value ```.n_eq'''
  class exit ```s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item depvars
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, depvars
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPosure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freeparm' `options'
= local item depvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item depvars
}

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item''''
= local value `y_timing_max'''
- sreturn local value ```.value'''
= sreturn local value `y_timing_max'''
- class exit ```.value'''
= class exit `y_timing_max'''
}

> ----- end _eqspec.item -----
- class exit ```s(value)'''
= class exit `y_timing_max'''

```

```

> ----- end _eqlist.eq -----
= local depvars y_timing_max
- local eq ``.eqlist'.eq name `i''

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "name" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item name

> ----- begin _eqspec.item -----
- version 9.2
- args item
- ._check_eq_item_name item, `item'
= ._check_eq_item_name item, name

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freeparm' `options'
= local item name
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item name

> ----- end _eqspec._check_eq_item_name -----
- local value ``.item'''
= local value `y_timing_max'

```

```

- sreturn local value ``value'''
= sreturn local value ``y_timing_max'''
- class exit ``value'''
= class exit ``y_timing_max'''


---


> 

---

 end _eqspec.item 

---


- class exit ``s(value)'''
= class exit ``y_timing_max'''


---


> 

---

 end _eqlist.eq 

---


= local eq y_timing_max
- if !':length local eq' {
= if !12 {
  local eq eq`i'
}
- if `:list eq in eqnames' {
= if 0 {
  di as err "equation ``eq'' multiply defined"
  exit 110
}
- local eqnames `eqnames' `eq'
= local eqnames y_timing_max
- fvexpand `.`_eqlist'.eq indepvars `i'' if $ML_samp


---


> 

---

 begin _eqlist.eq 

---


- version 9.2
- args item idx
- if ``item'' == "count" {
= if "indepvars" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item indepvars


---


> 

---

 begin _eqspec.item 

---


- version 9.2
- args item
- ._check_eq_item_name item, `item'
= ._check_eq_item_name item, indepvars


---


> 

---

 begin _eqspec._check_eq_item_name 

---


- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva

```

```

> rs IF IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
        local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freeparm' `options'
    = local item indepvars
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
        error 198
    }
    - c_local `c_item' `item'
    = c_local item indepvars


---


> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```
    = local value `cb_authors_ext estimated wg_ndx_mult stky_pr
> _rotemberg wlth stky_pr_calvo ln_neq open"
    - sreturn local value ``value'```
    = sreturn local value `cb_authors_ext estimated wg_ndx_mult
> stky_pr_rotemberg wlth stky_pr_calvo ln_neq open"
    - class exit ``value'```
    = class exit `cb_authors_ext estimated wg_ndx_mult stky_pr_
> rotemberg wlth stky_pr_calvo ln_neq open"


---


> ----- end _eqspec.item -----
    - class exit ``s(value)```
    = class exit `cb_authors_ext estimated wg_ndx_mult stky_pr_ro
> temberg wlth stky_pr_calvo ln_neq open"


---


> ----- end _eqlist.eq -----
    = fvexpand cb_authors_ext estimated wg_ndx_mult stky_pr_rotember
> g wlth stky_pr_calvo ln_neq open if __000007
    - if "`r(tsops)'" == "true" {
    = if "" == "true" {
        global ML_preserve no
    }
    - local eq_rhs`i' ``r(varlist)```
    = local eq_rhs1 `cb_authors_ext estimated wg_ndx_mult stky_pr_r
> otemberg wlth stky_pr_calvo ln_neq open"
    - local eq_nocons`i' `.`eqlist'.eq nocons `i'


---


> ----- begin _eqlist.eq -----
    - version 9.2

```

```

- args item idx
- if "`item'" == "count" {
= if "nocons" == "count" {
  sreturn local value ```.n_eq'''
  class exit ```s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item nocons
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, nocons
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPosure FREEArm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freearm' `options'
= local item nocons
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item nocons
}

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item''''
= local value ``0'''
- sreturn local value ```value'''
= sreturn local value ``0'''
- class exit ```value'''
= class exit ``0'''
}

> ----- end _eqspec.item -----
- class exit ```s(value)'''

```

```

= class exit ``0''
> ----- end _eqlist.eq -----
= local eq_nocons1 0
- local eq_offset`i' ``.eqlist'.eq offset `i''
> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item'' == "count" {
= if "offset" == "count" {
sreturn local value ``.n_eq''
class exit ``s(value)''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item offset
> ----- begin _eqspec.item -----
- version 9.2
- args item
- ._check_eq_item_name item, `item'
= ._check_eq_item_name item, offset
> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freeparm' `options'
= local item offset
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item offset
> ----- end _eqspec._check_eq_item_name -----
- local value ``.item''

```



```

= local value `""`
- sreturn local value `""`value`""`
= sreturn local value `""`
- class exit `""`value`""`
= class exit `""`

> ----- end _eqspec.item -----

- class exit `""`s(value)`""`
= class exit `""`

> ----- end _eqlist.eq -----

= local eq_offset1
- local eq_exposure`i' `.`eqlist'.eq exposure `i'

> ----- begin _eqlist.eq -----

- version 9.2
- args item idx
- if ""`item`"" == "count" {
= if "exposure" == "count" {
  sreturn local value `""`.n_eq`""`
  class exit `""`s(value)`""`
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 1 {
- .eq_list[`idx'].item `item`
= .eq_list[1].item exposure

> ----- begin _eqspec.item -----

- version 9.2
- args item
- .check_eq_item_name item, `item`
= .check_eq_item_name item, exposure

> ----- begin _eqspec._check_eq_item_name -----

- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt`
> `other' `nocons' `offset' `exposure' `freeparm' `options'
= local item exposure
- if `:list sizeof item' != 1 {

```

```

        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item exposure
    _____
> _____ end _eqspec._check_eq_item_name _____
        - local value ```.`item'"""
        = local value ``0""
        - sreturn local value ```value'"""
        = sreturn local value ``0""
        - class exit ```value'"""
        = class exit ``0""
    _____

> _____ end _eqspec.item _____
        - class exit ```s(value)"""
        = class exit ``0""
    _____

> _____ end _eqlist.eq _____
        = local eq_exposure1 0
        - local eq_freeparm`i' ``.`eqlist'.eq freeparm `i'
    _____

> _____ begin _eqlist.eq _____
        - version 9.2
        - args item idx
        - if ``item' == "count" {
        = if "freeparm" == "count" {
            sreturn local value ```.n_eq'"""
            class exit ```s(value)"""
        }
        - else if 0 < `idx' & `idx' <= `.n_eq' {
        = else if 0 < 1 & 1 <= 1 {
            - .eq_list[`idx'].item `item'
            = .eq_list[1].item freeparm
    _____

> _____ begin _eqspec.item _____
        - version 9.2
        - args item
        - ._check_eq_item_name item, `item'
        = ._check_eq_item_name item, freeparm
    _____

> _____ begin _eqspec._check_eq_item_name _____
        - version 9.2
        - capture syntax name(name=c_item) [, NAME DEPvars INDEPva
> rs IF IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
        - if c(rc) error 198

```

```

        - if `:list sizeof other' {
        = if 0 {
            local other other_markvars
        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt'
> `other' `nocons' `offset' `exposure' `freeparm' `options'
        = local item      freeparm
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item freeparm
    }
> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```
    = local value ``0```
    - sreturn local value ```.value```
    = sreturn local value ``0```
    - class exit ```.value```
    = class exit ``0```
> ----- end _eqspec.item -----
    - class exit ```.s(value)```
    = class exit ``0```
> ----- end _eqlist.eq -----
    = local eq_freeparm1 0
    - local lhs `lhs' `depvars'
    = local lhs  y_timing_max
    - local eq_names `eq_names' `eq'
    = local eq_names  y_timing_max
    - local nxvars :list sizeof eq_rhs`i'
    = local nxvars :list sizeof eq_rhs1
    - local TMP : copy local eq_rhs`i'
    = local TMP : copy local eq_rhs1
    - forval j = 1/`nxvars' {
    = forval j = 1/8 {
        - gettoken xvar TMP : TMP
        - local colna ```.colna' `eq':`xvar```
    = local colna `` y_timing_max:cb_authors_ext```
        - }
        - gettoken xvar TMP : TMP
        - local colna ```.colna' `eq':`xvar```
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted```

```

```

- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult''
- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg''
- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlt
> h''
- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlt
> h y_timing_max:stky_pr_calvo''
- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlt
> h y_timing_max:stky_pr_calvo y_timing_max:ln_neq''
- }
- gettoken xvar TMP : TMP
- local colna ``colna' `eq':`xvar''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlt
> h y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open''
- }
- if `eq_freeparm`i' {
= if 0 {
if `nxvars' {
di as err "{it:xvars} not allowed with option {bf:freeparm}"
exit 198
}
if strpos(`eq'', ":") {
local colna ``colna' /\`eq''
}
else {
local colna ``colna' /\`eq''
}
}

```

```

    }
    }
    - else if !`eq_nocons`i'' {
    = else if !0 {
    - local colna ``colna' `eq':_cons''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estima
> ted y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_max:wlt
> h y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_
> max:_cons''
    - }
    - if ``moptobj'' != "" {
    = if "" != "" {
    if ``eq_offset`i''' != "" {
    if usubstr("eq_offset`i'",1,2)=="__" {
    capture clonevar `moptobj'`eq_offset`i'' = `eq_offset`i''
    local dropmetoo `dropmetoo' `moptobj'`eq_offset`i''
    }
    }
    }
    - }
    - if ``moptobj'' != "" {
    = if "" != "" {
    foreach y of local lhs {
    if usubstr("`y'",1,2)=="__" {
    capture clonevar `moptobj'`y' = `y'
    local tmp `tmp' `moptobj'`y'
    local dropmetoo `dropmetoo' `moptobj'`y'
    }
    else {
    local tmp `tmp' `y'
    }
    }
    local lhs `tmp'
    }
    - if `waldtest' > 0 & `:length local lf0' {
    = if -1 > 0 & 19 {
    opts_exclusive "waldtest() lf0()"
    }
    - if `:length local continue' {
    = if 0 {
    if !`:length local init' {
    tempname init
    local ub0 `init'
    }
    else tempname ub0
    mat `ub0' = e(b)

```

```

tempname v0 iv0
mat `v0' = e(V)
mat `iv0' = syminv(`v0')
global ML_rank0 = colsof(`v0') - diag0cnt(`iv0')
if !':length local lf0' & !missing(e(ll)) {
    local lf0 `=colsof(`ub0')' `e(ll)'
}
local ub0
}
- if `:length local init' {
= if 8 {
- tempname b0
- local dim : list sizeof colna
- matrix `b0' = J(1,`dim',0)
= matrix __000009 = J(1,9,0)
- matrix colna `b0' = `colna'
= matrix colna __000009 = y_timing_max:cb_authors_ext y_timing_
> max:estimated y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timi
> ng_max:wlth y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open
> y_timing_max:_cons
- `vv' _mkvec `b0', from(`init') update first error("initial vec
> tor")
= version 18: _mkvec __000009, from(__000006) update first error
> ("initial vector")
> ----- begin _mkvec -----
- local vv : display "version " string(_caller()) ":"
- version 6
- gettoken matout 0 : 0, parse(", ")
- if "`matout'"==" | "`matout'"=="," {
= if "__000009"==" | "__000009"=="," {
    error 198
}
- syntax [, FROM(str) UPdate COLnames(str) ERRor(str) FIRST ]
- if "`update'"!="" {
= if "update"!="" {
- if "`colname'"!="" {
= if ""!="" {
    di in red "update and colnames() cannot both be " "specified
> "
    exit 198
}
- capture local colname : colfullnames `matout'
= capture local colname : colfullnames __000009
- if _rc {
    di in red "matrix `matout' not found"

```

```

        exit 111
    }
    - if rowsof(matrix(`matout')) != 1 {
= if rowsof(matrix(__000009)) != 1 {
        di in red ``matout' matrix to be updated must be " "a row ve
> ctor"

        exit 503
    }
    - local matin `matout'
= local matin __000009
    - }
    - if ``error'!='' {
= if ``initial vector'!='' {
    - local error ``error': ''
= local error ``initial vector: ''
    - }
    - tempname mat
    - if ``from'!='' {
= if ``__000006'!='' {
    - From `from'
= From __000006

> ----- begin _mkvec.From -----
    - gettoken from 0 : 0, parse(",") bind
    - sret clear
    - sret local from `from'
= sret local from __000006
    - if ``0'!='' {
= if ``'!='' {
        syntax [, COPY SKIP ]
    }
    - sret local copy `copy'
= sret local copy
    - sret local skip `skip'
= sret local skip

> ----- end _mkvec.From -----
    - local copy `s(copy)'
= local copy
    - local skip `s(skip)'
= local skip
    - if ``copy'!='' {
= if ``'!='' {
        CopyMat `mat' ``error' `s(from)'
        if ``colname'!='' {
            local dim : list sizeof colname

```

```

        if `dim' != colsof(matrix(`mat')) {
        di in red `"'error'matrix must "'`be dimension "' `dim'
        exit 503
        }
        `vv' mat colnames `mat' = `colname'
        }
    }
- else {
- `vv' BuildMat `mat' `"'error'"' "`skip'" `s(from)'
= version 18: BuildMat __00000A `initial vector: "' "" __0000
> 06

> ----- begin _mkvec.BuildMat -----
- local vv : display "version " string(_caller()) ":"
- version 6
- args matall error skip
- macro shift 3
- tokenize `"'*'"', parse(" =")
= tokenize `__000006"', parse(" =")
- tempname matadd
- local i 1
- while `"'i'"'!="' {
= while "__000006"!="' {
- local k = `i' + 1
= local k = 1 + 1
- if `"'k'"'=="' {
= if ""=="' {
    local j = `i' + 2
    `vv' MkMat `matadd' `i' `j' `skip'
    local i = `i' + 3
  }
- else {
- ChkMat `matadd' `i' `"'error'"'
= ChkMat __00000B __000006 `initial vector: "'
> ----- begin _mkvec.ChkMat -----
- args matout matin error
- capture di matrix(`matin'[1,1])
= capture di matrix(__000006[1,1])
- if _rc {
    capture confirm number `matin'
    if _rc {
        di in red `"'error'matrix `matin' not found"'
        exit 111
    }
    di in red `"'error'list of numbers requires copy option"

```



```

> '
    exit 198
    }
    - if matrix(rowsof(`matin')!=1 & colsof(`matin')!=1) {
= if matrix(rowsof(__000006)!=1 & colsof(__000006)!=1) {
    di in red "`matin' " matrix(rowsof(`matin')) " x " matri
> x(colsof(`matin')) " is not a vector"
    exit 503
    }
    - if matrix(rowsof(`matin')) == 1 {
= if matrix(rowsof(__000006)) == 1 {
    - mat `matout' = `matin'
= mat __00000B = __000006
    - }
    - else mat `matout' = `matin''
= else mat __00000B = __000006'
}

> ----- end _mkvec.ChkMat -----

    - local i = `i' + 1
= local i = 1 + 1
    - }
    - mat `matall' = nullmat(`matall') , `matadd'
= mat __00000A = nullmat(__00000A) , __00000B
    - }
    - while "`i'"!=" " {
= while ""!=" " {
    local k = `i' + 1
    if "`k'"==" " {
    local j = `i' + 2
    `vv' MkMat `matadd' ``i'' ``j'' `skip'
    local i = `i' + 3
    }
    else {
    ChkMat `matadd' ``i'' ``error''
    local i = `i' + 1
    }
    mat `matall' = nullmat(`matall') , `matadd'
    }
}

> ----- end _mkvec.BuildMat -----

    - if ``colname'"!=" " {
= if ``y_timing_max:cb_authors_ext y_timing_max:estimated y_ti
> ming_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlth y_timi
> ng_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_max:con
> s'"!=" " {
    - `vv' ChkNames `mat' "matin'" "colname'" "error'" "sk

```

```

> ip'" "`first'"
= version 18: ChkNames __00000A "__000009" `"y_timing_max:cb_a
> uthors_ext y_timing_max:estimated y_timing_max:wg_ndx_mult y_timing_max:stky
> _pr_rottemberg y_timing_max:wlth y_timing_max:stky_pr_calvo y_timing_max:ln_n
> eq y_timing_max:open y_timing_max:_cons"' `"initial vector: "' "" "first"

> ----- begin _mkvec.ChkNames -----
- local vv : display "version " string(_caller()) ":"
- version 6
- args mat matin names error skip first
- local dim = colsof(matrix(`mat'))
= local dim = colsof(matrix(__00000A))
- if "`first'"!="" {
= if "first"!="" {
- gettoken name : names
- local eq1 = bsubstr(`"name"',1,index(`"name"',":"))
= local eq1 = bsubstr(`"y_timing_max:cb_authors_ext"',1,inde
> x(`"y_timing_max:cb_authors_ext"',":"))
- }
- tempname new fill x
- local k : list sizeof names
- if "`matin'"!="" {
= if "__000009"!="" {
- mat `new' = `matin'
= mat __00000B = __000009
- }
- else {
mat `new' = J(1,`k',0)
`vv' mat colnames `new' = `names'
}
- mat `fill' = J(1,`k',0)
= mat __00000C = J(1,9,0)
- local i 1
- while `i' <= `dim' {
= while 1 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,1..1]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if "`eq'" == "/" {
= if "_" == "/" {
local spec "/"`name'"
}
- else {

```

```

- local spec "`eq':`name'"
= local spec "_:cb_authors_ext"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "/_:cb_authors_ext"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse cb_authors_ext, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)''
    local ospec "`eq':`oname'"
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000B,`":cb_authors_ext"

> ')

- if `j' == . & "`oname'" != "" {
= if 1 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & "`sspec'" != "" {
= if 1 == . & "/_:cb_authors_ext" != "" {
  `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
= if 1==. & ``_""=="_" & ``y_timing_max:'""!="" {
  `vv' local j = colnumb(`new',``eq1``name'')
  if `j' == . & "`oname'" != "" {
    `vv' local j = colnumb(`new',``eq1``oname'')
  }
}
- if `j'!=. {
= if 1!=. {
- if `fill'[1,`j'] {
= if __00000C[1,1] {
  _ms_parse_parts `name'
  if !r(omit) {
    if "`eq'""=="_" {
      di in red ``error'duplicate "" ``entries for `name' found

> ""
}

```

```

else di in red ``error'' ``duplicate entries for `eq':`n
> ame' found''
    exit 507
}
else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,1] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,1] = __00000A[1,1]
- }
- else if ``skip'')=="" {
= else if ""==" {
    if ``eq'')== "_" {
        local spec : copy local name
    }
    di in red ``error'extra parameter "' ``spec' found'' _n
> "specify skip option if necessary"
    exit 111
}
- local i = `i' + 1
= local i = 1 + 1
- }
- while `i' <= `dim' {
= while 2 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,2..2]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if ``eq'')== "/" {
= if "_" == "/" {
    local spec "/`name'"
}
- else {
- local spec ``eq':`name'"
= local spec "_:estimated"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "/_:estimated"
- }
- }
}

```

```

- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse estimated, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)'''
    local ospec ``eq':`name'''
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000B,``_:estimated'')
- if `j' == . & ``oname''' != "" {
= if 2 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & ``sspec''' != "" {
= if 2 == . & "/_:estimated" != "" {
  `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'''=="_" & ``eq1'''!="" {
= if 2==. & ``_'=="_" & ``y_timing_max:'!="" {
  `vv' local j = colnumb(`new',``eq1``name'')
  if `j' == . & ``oname''' != "" {
    `vv' local j = colnumb(`new',``eq1``oname'')
  }
}
- if `j'!=. {
= if 2!=. {
- if `fill'[1,`j'] {
= if __00000C[1,2] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ``eq'''=="_" {
      di in red ``error'duplicate "'`"entries for `name' found
> "'
    }
    else di in red ``error''''`"duplicate entries for `eq':`n
> ame' found'''
  }
  exit 507
}
else if !missing(`new'[1,`j']) {
  matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1

```

```

= else mat __00000C[1,2] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,2] = __00000A[1,2]
- }
- else if "`skip'"==" {
= else if ""==" {
  if "`eq'"=="_" {
    local spec : copy local name
  }
  di in red ``error'extra parameter "' ``spec' found"' _n
> "specify skip option if necessary"
  exit 111
}
- local i = `i' + 1
= local i = 2 + 1
- }
- while `i' <= `dim' {
= while 3 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,3..3]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if "`eq'" == "/" {
= if "_" == "/" {
  local spec "/`name'"
}
- else {
- local spec "`eq':`name'"
= local spec "._wg_ndx_mult"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "._wg_ndx_mult"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse wg_ndx_mult, noomit
- if c(rc) == 0 {
- if r(omit) {
  local oname ``r(stripe)'"
  local ospec "`eq':`oname'"
}
}

```

```

- }
- `vv' local j = colnumb(`new',`""`spec'')
= version 18: local j = colnumb(__00000B,`":wg_ndx_mult'')
- if `j' == . & ""`oname' != "" {
= if 3 == . & "" != "" {
  `vv' local j = colnumb(`new',`""`ospec'')
  }
- if `j' == . & ""`sspec' != "" {
= if 3 == . & "/_wg_ndx_mult" != "" {
  `vv' local j = colnumb(`new',`""`sspec'')
  }
- if `j'==. & ""`eq'=="_" & ""`eq1'!="" {
= if 3==. & ""`_"=="_" & ""`y_timing_max:'!="" {
  `vv' local j = colnumb(`new',`""`eq1'`name'')
  if `j' == . & ""`oname' != "" {
    `vv' local j = colnumb(`new',`""`eq1'`oname'')
  }
  }
- if `j'!=. {
= if 3!=. {
- if `fill'[1,`j'] {
= if __00000C[1,3] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ""`eq'=="_" {
      di in red ""`error'duplicate ""`"entries for `name' found
> ""
    }
    else di in red ""`error'""`"duplicate entries for `eq':`n
> ame' found""
  }
  exit 507
}
else if !missing(`new'[1,`j']) {
  matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,3] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,3] = __00000A[1,3]
- }
- else if ""`skip'=="" {
= else if ""=="" {
  if ""`eq'=="_" {
    local spec : copy local name
  }
}

```

```

        di in red ``error'extra parameter "' ``spec' found"' _n
> "specify skip option if necessary"
    exit 111
    }
    - local i = `i' + 1
    = local i = 3 + 1
    - }
    - while `i' <= `dim' {
    = while 4 <= 9 {
    - mat `x' = `mat'[1,`i'..`i']
    = mat __00000D = __00000A[1,4..4]
    - local eq : coleq `x'
    = local eq : coleq __00000D
    - local name : colname `x'
    = local name : colname __00000D
    - if "`eq'" == "/" {
    = if "_" == "/" {
        local spec "/"`name'"
    }
    - else {
    - local spec "`eq':`name'"
    = local spec "_:stky_pr_rotemberg"
    - if (usubstr("`eq'",1,1) != "/") {
    = if (usubstr("_",1,1) != "/") {
    - local sspec "/"`eq':`name'"
    = local sspec "/_:stky_pr_rotemberg"
    - }
    - }
    - local oname
    - local ospec
    - capture _msparse `name', noomit
    = capture _msparse stky_pr_rotemberg, noomit
    - if c(rc) == 0 {
    - if r(omit) {
        local oname ``r(stripe)'"
        local ospec "`eq':`oname'"
    }
    - }
    - `vv' local j = colnumb(`new',``spec'')
    = version 18: local j = colnumb(__00000B,`_:stky_pr_rotembe
> rg'')

    - if `j' == . & "`oname'" != "" {
    = if 4 == . & "" != "" {
        `vv' local j = colnumb(`new',``ospec'')
    }
    - if `j' == . & "sspec'" != "" {

```



```

= if 4 == . & "/_:stky_pr_rotemberg" != "" {
  `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'""=="_ & ``eq1'""!=" {
= if 4==. & ``_'""=="_ & ``y_timing_max:'""!=" {
  `vv' local j = colnumb(`new',``eq1'`name'')
  if `j' == . & ``oname'"" != "" {
    `vv' local j = colnumb(`new',``eq1'`oname'')
  }
}
- if `j'!=. {
= if 4!=. {
- if `fill'[1,`j'] {
= if __00000C[1,4] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ``eq'""=="_ {
      di in red ``error'duplicate "' ``entries for `name' found
> "'

    }
    else di in red ``error'"" ``duplicate entries for `eq':`n
> ame' found"'

    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,4] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,4] = __00000A[1,4]
- }
- else if ``skip'""=="" {
= else if ""=="" {
  if ``eq'""=="_ {
    local spec : copy local name
  }
  di in red ``error'extra parameter "' ``spec' found"' _n
> "specify skip option if necessary"
  exit 111
}
- local i = `i' + 1
= local i = 4 + 1
- }
- while `i' <= `dim' {

```

```

= while 5 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,5..5]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if "`eq'" == "/" {
= if "_" == "/" {
    local spec "/`name'"
  }
- else {
- local spec "`eq':`name'"
= local spec "_:wlth"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "/_:wlth"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse wlth, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)''
    local ospec "`eq':`oname'"
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000B,":wlth")
- if `j' == . & "`oname'" != "" {
= if 5 == . & "" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
- if `j' == . & "`sspec'" != "" {
= if 5 == . & "/_:wlth" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
- if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
= if 5==. & ``_'""=="_" & ``y_timing_max:'""!="" {
    `vv' local j = colnumb(`new',``eq1``name'')
    if `j' == . & "`oname'" != "" {
      `vv' local j = colnumb(`new',``eq1``oname'')
    }
  }
}

```

```

    }
- if `j'!=. {
= if 5!=. {
- if `fill'[1,`j'] {
= if __00000C[1,5] {
    _ms_parse_parts `name'
    if !r(omit) {
    if "`eq'"=="_" {
di in red `"'error'duplicate "' `"'entries for `name' found
> "'

    }
else di in red `"'error'"'"' `"'duplicate entries for `eq':`n
> ame' found"'

    exit 507
    }
    else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
    }
    }
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,5] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,5] = __00000A[1,5]
- }
- else if "`skip'"==" " {
= else if ""==" " {
    if "`eq'"=="_" {
    local spec : copy local name
    }
di in red `"'error'extra parameter "' `"'`spec' found"' _n
> "specify skip option if necessary"
    exit 111
    }
- local i = `i' + 1
= local i = 5 + 1
- }
- while `i' <= `dim' {
= while 6 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,6..6]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if "`eq'" == "/" {
= if "_" == "/" {

```

```

        local spec "/"`name'"
    }
- else {
- local spec "`eq':`name'"
= local spec "_:stky_pr_calvo"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/"`eq':`name'"
= local sspec "/_:stky_pr_calvo"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse stky_pr_calvo, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)''
    local ospec "`eq':`oname'"
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000B,`_:stky_pr_calvo')

> )

- if `j' == . & "`oname'" != "" {
= if 6 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & "`sspec'" != "" {
= if 6 == . & "/_:stky_pr_calvo" != "" {
  `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
= if 6==. & ``_'""=="_" & ``y_timing_max:'""!="" {
  `vv' local j = colnumb(`new',``eq1``name'')
  if `j' == . & "`oname'" != "" {
    `vv' local j = colnumb(`new',``eq1``oname'')
  }
}
- if `j'!=. {
= if 6!=. {
- if `fill'[1,`j'] {
= if __00000C[1,6] {
  _ms_parse_parts `name'
  if !r(omit) {
    if "`eq'""=="_" {

```

```

di in red ``error'duplicate "' ``entries for `name' found
> "'
}
else di in red ``error'''' ``duplicate entries for `eq':`n
> ame' found'''

exit 507
}
else if !missing(`new'[1,`j']) {
matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,6] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,6] = __00000A[1,6]
- }
- else if ``skip'""=="" {
= else if ""=="" {
if ``eq'""=="_" {
local spec : copy local name
}
di in red ``error'extra parameter "' ``spec' found"' _n
> "specify skip option if necessary"
exit 111
}
- local i = `i' + 1
= local i = 6 + 1
- }
- while `i' <= `dim' {
= while 7 <= 9 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,7..7]
- local eq : coleq `x'
= local eq : coleq __00000D
- local name : colname `x'
= local name : colname __00000D
- if ``eq'"" == "/" {
= if "_" == "/" {
local spec "/`name'"
}
- else {
- local spec ``eq':`name'"
= local spec "_:ln_neq"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
- local sspec "/`eq':`name'"

```

```

= local sspec "/_:ln_neq"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse ln_neq, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)'''
    local ospec ``eq':`oname'''
  }
- }
- `vv' local j = colnumb(`new',``spec''')
= version 18: local j = colnumb(__00000B,``_:ln_neq'')
- if `j' == . & ``oname''' != "" {
= if 7 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec''')
}
- if `j' == . & ``sspec''' != "" {
= if 7 == . & "/_:ln_neq" != "" {
  `vv' local j = colnumb(`new',``sspec''')
}
- if `j'==. & ``eq''''=="_" & ``eq1''''!="" {
= if 7==. & ``_''=="_" & ``y_timing_max:'''!="" {
  `vv' local j = colnumb(`new',``eq1``name''')
  if `j' == . & ``oname''' != "" {
    `vv' local j = colnumb(`new',``eq1``oname''')
  }
}
- if `j'!=. {
= if 7!=. {
- if `fill'[1,`j'] {
= if __00000C[1,7] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ``eq''''=="_" {
      di in red ``error'duplicate "" ``entries for `name' found
> ""
    }
    else di in red ``error'''' ``duplicate entries for `eq':`n
> ame' found'''
  }
  exit 507
}
else if !missing(`new'[1,`j']) {
matrix `mat'[1,`i'] = `new'[1,`j']

```

```

    }
  }
  - else mat `fill'[1,`j'] = 1
= else mat __00000C[1,7] = 1
  - mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,7] = __00000A[1,7]
  - }
  - else if "`skip'"==" {
= else if ""==" {
    if "`eq'"=="_" {
      local spec : copy local name
    }
    di in red ``error'extra parameter "' `"'spec' found"' _n
> "specify skip option if necessary"
    exit 111
  }
  - local i = `i' + 1
= local i = 7 + 1
  - }
  - while `i' <= `dim' {
= while 8 <= 9 {
  - mat `x' = `mat'[1,`i'..`i']
= mat __00000D = __00000A[1,8..8]
  - local eq : coleq `x'
= local eq : coleq __00000D
  - local name : colname `x'
= local name : colname __00000D
  - if "`eq'" == "/" {
= if "_" == "/" {
    local spec "/"`name'
  }
  - else {
  - local spec "`eq':`name'"
= local spec "_:open"
  - if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("_",1,1) != "/") {
  - local sspec "/"`eq':`name'"
= local sspec "/_:open"
  - }
  - }
  - local oname
  - local ospec
  - capture _msparse `name', noomit
= capture _msparse open, noomit
  - if c(rc) == 0 {
  - if r(omit) {

```

```

        local oname ``r(stripe)''
        local ospec ``eq':`oname''
    }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000B,``_:open'')
- if `j' == . & ``oname'' != "" {
= if 8 == . & "" != "" {
    `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & ``sspec'' != "" {
= if 8 == . & "/_:open" != "" {
    `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq''=="_" & ``eq1''!="" {
= if 8==. & ``_'=="_" & ``y_timing_max:''!="" {
    `vv' local j = colnumb(`new',``eq1``name'')
    if `j' == . & ``oname'' != "" {
        `vv' local j = colnumb(`new',``eq1``oname'')
    }
}
- if `j'!=. {
= if 8!=. {
- if `fill'[1,`j'] {
= if __00000C[1,8] {
    _ms_parse_parts `name'
    if !r(omit) {
        if ``eq''=="_" {
            di in red ``error'duplicate '' ``entries for `name' found
> ""

        }
        else di in red ``error'' ``duplicate entries for `eq':`n
> ame' found''

    exit 507
}
else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,8] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,8] = __00000A[1,8]
- }
- else if ``skip''="" {
= else if ""="" {

```



```

        if "`eq'"=="_" {
            local spec : copy local name
        }
        di in red `"'error'extra parameter "' `"'spec' found"' _n
> "specify skip option if necessary"
        exit 111
    }
    - local i = `i' + 1
    = local i = 8 + 1
    - }
    - while `i' <= `dim' {
    = while 9 <= 9 {
    - mat `x' = `mat'[1,`i'..`i']
    = mat __00000D = __00000A[1,9..9]
    - local eq : coleq `x'
    = local eq : coleq __00000D
    - local name : colname `x'
    = local name : colname __00000D
    - if "`eq'" == "/" {
    = if "_" == "/" {
        local spec "/"`name'
    }
    - else {
    - local spec "`eq':`name'"
    = local spec ":_cons"
    - if (usubstr("`eq'",1,1) != "/") {
    = if (usubstr("_",1,1) != "/") {
    - local sspec "/"`eq':`name'"
    = local sspec "/_:_cons"
    - }
    - }
    - local oname
    - local ospec
    - capture _msparse `name', noomit
    = capture _msparse _cons, noomit
    - if c(rc) == 0 {
    - if r(omit) {
        local oname ``r(stripe)'''
        local ospec "`eq':`oname'"
    }
    - }
    - `vv' local j = colnumb(`new',``"spec'")
    = version 18: local j = colnumb(__00000B,`":_cons'")
    - if `j' == . & "`oname'" != "" {
    = if 9 == . & "" != "" {
        `vv' local j = colnumb(`new',``"ospec'")

```

```

    }
- if `j' == . & "`sspec'" != "" {
= if 9 == . & "/_:_cons" != "" {
  `vv' local j = colnumb(`new', "`sspec'")
}
- if `j'==. & "`eq'"=="_" & "`eq1'"!="" {
= if 9==. & "`_"=="_" & "`y_timing_max:'"!="" {
  `vv' local j = colnumb(`new', "`eq1`name'")
  if `j' == . & "oname'" != "" {
    `vv' local j = colnumb(`new', "`eq1`oname'")
  }
}
- if `j'!=. {
= if 9!=. {
- if `fill'[1,`j'] {
= if __00000C[1,9] {
  _ms_parse_parts `name'
  if !r(omit) {
    if "`eq'"=="_" {
      di in red "`error'duplicate "' `"'entries for `name' found
> ""
    }
    else di in red "`error'" "duplicate entries for `eq':`n
> ame' found"

    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000C[1,9] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000B[1,9] = __00000A[1,9]
- }
- else if "`skip'"=="" {
= else if ""=="" {
  if "`eq'"=="_" {
    local spec : copy local name
  }
  di in red "`error'extra parameter "' `"'spec' found" _n
> "specify skip option if necessary"
  exit 111
}
- local i = `i' + 1
= local i = 9 + 1

```

```

- }
- while `i' <= `dim' {
= while 10 <= 9 {
  mat `x' = `mat'[1,`i'..'i']
  local eq : coleq `x'
  local name : colname `x'
  if "`eq'" == "/" {
    local spec "/"`name'
  }
  else {
    local spec "`eq':`name'
    if (usubstr("`eq'",1,1) != "/") {
      local sspec "/"`eq':`name'
    }
  }
  local oname
  local ospec
  capture _msparse `name', noomit
  if c(rc) == 0 {
    if r(omit) {
      local oname ``r(stripe)''
      local ospec "`eq':`oname'
    }
  }
  `vv' local j = colnumb(`new',``spec'')
  if `j' == . & "`oname'" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
  if `j' == . & "`sspec'" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
  if `j'==. & ``eq'=="_" & ``eq1'!='' {
    `vv' local j = colnumb(`new',``eq1``name'')
  }
  if `j' == . & "`oname'" != "" {
    `vv' local j = colnumb(`new',``eq1``oname'')
  }
}
if `j'!=. {
  if `fill'[1,`j'] {
    _ms_parse_parts `name'
    if !r(omit) {
      if "`eq'=="_" {
        di in red ``error'duplicate "'`"entries for `name' found
> ""
      }
    }
  }
  else di in red ``error'' ``duplicate entries for `eq':`n

```

```

> ame' found''
    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
  }
  else mat `fill'[1,`j'] = 1
  mat `new'[1,`j'] = `mat'[1,`i']
  }
  else if ""skip""=="" {
    if ""eq""=="_" {
      local spec : copy local name
    }
    di in red ""error'extra parameter "" ``spec' found"" _n
> "specify skip option if necessary"
    exit 111
  }
  local i = `i' + 1
  }
  - mat `fill' = `fill'*`fill''
  = mat __00000C = __00000C*__00000C'
  - sret clear
  - sret local k_fill = `fill'[1,1]
  = sret local k_fill = __00000C[1,1]
  - mat `mat' = `new'
  = mat __00000A = __00000B



---


> 

---

 end _mkvec.ChkNames 

---


  - local fill `s(k_fill)'
  = local fill 9
  - }
  - }
  - }
  - else {
    local fill 0
    if ""update""!="" {
      mat `mat' = `matin'
    }
    else if ""colname""!="" {
      local dim : list sizeof colname
      mat `mat' = J(1,`dim',0)
      `vv' mat colnames `mat' = `colname'
    }
    else {
      di in red "from() must be specified"

```

```

        exit 198
    }
}
- sret clear
- sret local copy `copy'
= sret local copy
- sret local skip `skip'
= sret local skip
- sret local k = colsof(matrix(`mat'))
= sret local k = colsof(matrix(__00000A))
- if "`fill'"!="" {
= if "9"!=" " {
- sret local k_fill `fill'
= sret local k_fill 9
- }
- else sret local k_fill = colsof(matrix(`mat'))
= else sret local k_fill = colsof(matrix(__00000A))
- matrix rename `mat' `matout', replace
= matrix rename __00000A __000009, replace


---


> 

---

 end _mkvec 

---


- }
- tempname bomit
- matrix `bomit' = J(1,`:list sizeof colna',0)
= matrix __00000A = J(1,9,0)
- matrix colna `bomit' = `colna'
= matrix colna __00000A = y_timing_max:cb_authors_ext y_timing_
> max:estimated y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timi
> ng_max:wlth y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open
> y_timing_max:_cons
- _ms_omit_info `bomit'
= _ms_omit_info __00000A
- if r(k_omit) | `:length local constraints' {
= if r(k_omit) | 0 {
nobreak {
capture noisily break {
_b_post0 `colna'
local clist : subinstr local constraints ", " " ", all
`vv' makecns `clist', `cnsnotes'
local constraints ``r(clist)'''
local k_autoCns = e(k_autoCns)
tempname C
capture matrix `C' = e(Cns)
local rcc = c(rc)
}
local rc = c(rc)

```

```

        ereturn clear
    }
    if (`rc') exit `rc'
    if (`rcc') {
        local C
        local k_autoCns
    }
}

- if `:length local constraints' {
= if 0 {
    local lf0
}
- local vce `vce' `robust' `cluster'
= local vce robust
- if `:length local vce' {
= if 6 {
- _vce_parse, argoptlist(CLister) optlist(OIM OPG Robust) old :
> `wt', `vce'
= _vce_parse, argoptlist(CLister) optlist(OIM OPG Robust) old :
> , robust

> ----- begin _vce_parse -----
- version 10
- _on_colon_parse `0'
= _on_colon_parse , argoptlist(CLister) optlist(OIM OPG Robust
> ) old : , robust

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quot
> es
- if "`par'" != "" {
= if "" != "" {
    local before `("`before')"'
}
- if ``"before"' == ":" {
= if `", argoptlist(CLister) optlist(OIM OPG Robust) old ""
> == ":" {
    sreturn local after ``"after"'
    exit
}
- while ``COLON"' != ":" & ``"after"' != "" {
= while `"' != ":" & `": , robust"' != "" {
- gettoken COLON after : after, parse(":") bind match(par) q

```

```

> uotes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``"COLON"' != ":" {
= else if `":' != ":" {
  local before ``"before' `COLON'
  local COLON
}
- }
- while ``"COLON"' != ":" & ``"after"' != "" {
= while `":' != ":" & `", robust"' != "" {
  gettoken COLON after : after, parse(":") bind match(par) q

> uotes

  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``"COLON"' != ":" {
    local before ``"before' `COLON'
    local COLON
  }
}
- if ``"COLON"' != ":" {
= if `":' != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before ``"before'
= sreturn local before `", argoptlist(CLister) optlist(OIM 0

> PG Robust) old ""
- sreturn local after ``"after'
= sreturn local after `", robust"

-----
> ----- end _on_colon_parse -----
- local 0 ``"s(before)'
= local 0 `", argoptlist(CLister) optlist(OIM OPG Robust) old

> ""
- local user ``"s(after)'
= local user `", robust"
- syntax [varname(default=none)], [ OPTlist(namelist) ARGOPTli
> st(namelist) PWallowed(namelist) OLD_is_ok NOARGOPT_is_ok ]
- local touse `varlist'
= local touse

```

```

- tempname o a
- .`o' = ._optlist.new
= .__00000B = ._optlist.new
- local optlist : list uniq optlist
- foreach name of local optlist {
- .`o'.addopt `name'
= .__00000B.addopt OIM

```

---

```

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OIM")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
  }
- .options.Arrpush "`optspec'"
= .options.Arrpush "OIM"
- .names.Arrpush "`mname'"
= .names.Arrpush "oim"

```

---

```

> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= .__00000B.addopt OPG

```

---

```

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OPG")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
  }
- .options.Arrpush "`optspec'"
= .options.Arrpush "OPG"
- .names.Arrpush "`mname'"
= .names.Arrpush "opg"

```

---

```

> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= .__00000B.addopt Robust

```

---



```

> ----- begin _optlist.adopt -----
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("Robust")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "Robust"
    - .names.Arrpush "`mname'"
    = .names.Arrpush "robust"
-----

> ----- end _optlist.adopt -----
    - }
    - .`a' = ._optlist.new
    = .__00000C = ._optlist.new
    - local argoptlist : list uniq argoptlist
    - foreach name of local argoptlist {
    - .`a'.adopt `name'
    = .__00000C.adopt CLuster
-----

> ----- begin _optlist.adopt -----
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("CLuster")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "CLuster"
    - .names.Arrpush "`mname'"
    = .names.Arrpush "cluster"
-----

> ----- end _optlist.adopt -----
    - }
    - local OPTNAMES ```.`o'.dumpnames'''
-----

> ----- begin _optlist.dumpnames -----
    - version 9
    - forval i = 1/`.names.arnels' {
    = forval i = 1/3 {
    - local namelist `namelist' `.names[`i']'

```

```

= local namelist oim
- }
- local namelist `namelist' `.names['i']'
= local namelist oim opg
- }
- local namelist `namelist' `.names['i']'
= local namelist oim opg robust
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `oim opg robust''
- class exit ``namelist'''
= class exit `oim opg robust''

> ----- end _optlist.dumpnames -----
= local OPTNAMES `oim opg robust''
- local ARGOPTNAMES ```.a'.dumpnames'''

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arrnels' {
= forval i = 1/1 {
- local namelist `namelist' `.names['i']'
= local namelist cluster
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `cluster''
- class exit ``namelist'''
= class exit `cluster''

> ----- end _optlist.dumpnames -----
= local ARGOPTNAMES `cluster''
- local AND : list OPTNAMES & ARGOPTNAMES
- if `:list sizeof AND' {
= if 0 {
di as err "vctype `:word 1 of `AND'' specified twice"
exit 198
}
- local pwallowed : list uniq pwallowed
- local pwlist robust cluster
- foreach n of local pwallowed {
local lc = lower("`n'")
local pwlist `pwlist' `lc'
}
- if "`old_is_ok'" != "" {

```

```

= if "old_is_ok" != "" {
- local ROBUST robust
- if `:list ROBUST in OPTNAMES' {
= if 1 {
- local robust Robust
- }
- local CLUSTER cluster
- if `:list CLUSTER in ARGOPTNAMES' {
= if 1 {
- local cluster CLuster(varname)
- }
- }
- local 0 ``user''
= local 0 `` , robust''
- syntax [fw aw pw iw] [, VCE(string asis) `robust' `cluster'

> ]

= syntax [fw aw pw iw] [, VCE(string asis) Robust CLuster(varn

> ame) ]

- if ``vce'' != "" {
= if `'' != "" {
gettoken name args : vce, parse(" ,")
local args : list retok args
capture ParseVCE `o' `a' , `name'
if c(rc) {
di as err "vcetype {bf:`name'} not allowed"
exit 198
}
local name `r(vce)'
if ``weight' == "pweight" & !`:list name in pwlist' {
di as err "pweights are not allowed with vce(`name')"
exit 198
}
return add
if ``robust'' != "" & !inlist("`name'", "", "robust", "eim", "

> oim", "cluster") {
opts_exclusive ``robust' vce(`name')"
}
if ``name' == "cluster" {
if ``args' == "" {
di as err "invalid vce(cluster) option"
exit 198
}
confirm var `args'
unab args : `args'
local markout `args'
if ``cluster' != "" {

```

```

if !`:list args == cluster' {
di as err "options cluster() and vce(cluster) are in conflic
> t"

exit 100
}
}
}
else if "`cluster'" != "" & !inlist("`name'", "eim", "oim") {
opts_exclusive "cluster() vce(`name')"
}
if ```args'"" != "" {
if !`:list name in ARGOPTNAMES' {
di as err "invalid vce() option"
exit 198
}
if "`name'" == "hc2" {
confirm var `args'
unab args : `args'
local markout `args'
}
return local vceargs ```:list retok args'""
}
else if `:list name in ARGOPTNAMES' {
if "`noargopt_is_ok'" == "" {
di as err "invalid vce() option"
exit 198
}
}
}
- else {
- if "`weight'" == "pweight" | "`robust'" != "" {
= if "" == "pweight" | "robust" != "" {
- local vce robust
- }
- if "`cluster'" != "" {
= if "" != "" {
local vce cluster
local args `cluster'
local markout `cluster'
}
- return local vce ```:vce'""
= return local vce ``robust'
- return local vceargs ```:args'""
= return local vceargs ``""
- }
- if "`return(vce)'" != "" {

```

```

= if "robust" != "" {
- if ``return(vceargs)'' != "" {
= if `''' != "" {
  local ss " "
}
- return local vceopt ``vce(`return(vce)''ss``return(vceargs)''

> )''

= return local vceopt ``vce(robust)''
- }
- if ``return(vce)'' == "cluster" {
= if "robust" == "cluster" {
  return local cluster `return(vceargs)'
}
- if inlist("`return(vce)'", "robust", "cluster") {
= if inlist("robust", "robust", "cluster") {
- return local robust robust
- }
- if ``touse'' != "" {
= if "" != "" {
  if ``markout'' != "" {
    markout `touse' `markout', strok
  }
}
}



---


> 

---

 end _vce_parse 

---



- local vce `r(vce)'
= local vce robust
- }
- else if ``weight'' == "pweight" {
= else if "" == "pweight" {
  local vce robust
}
- if `:length local svy' {
= if 0 {
  local vce svy
}
- if ``vce'' == "cluster" {
= if ``robust'' == "cluster" {
  local cluster `r(cluster)'
  markout $ML_sample `cluster', strok
}
- if `:length local group' {
= if 0 {
  markout $ML_sample `group', strok
  global ML_group : copy local group
  if bsubstr("`type `group'",1,3) == "str" {

```

```

if `interactive' {
  SetGlobalVar ML_grp : _MLgr
}
else {
  global ML_grp : tempvar
  if ``moptobj' != "" {
    global ML_grp `moptobj'$ML_grp
  }
}
quietly egen $ML_grp = group(`group')
if ! `:length local cluster' {
  local cluster : copy global ML_grp
}
}
else {
  if ! `:length local cluster' {
    local cluster : copy global ML_group
  }
}
}
- if "$ML_samp" != "$ML_sample" {
= if "__000007" != "__000007" {
  quietly replace $ML_samp = 0 if $ML_samp & $ML_sample == 0
}
- if `:length local vce' == 0 {
= if 6 == 0 {
  if ``technique' == "bhhh" {
    local vce opg
  }
  else local vce oim
}
- if inlist("`vce'", "robust", "cluster") & bsubstr("`evaltype'"
> ,1,1) == "d" {
  = if inlist("robust", "robust", "cluster") & bsubstr("e2",1,1) =
> = "d" {
    local optname = cond("`svy'"=="", "vce(`vce')", "svy")
    di as err "option `optname' is not allowed with evaltype `eval
> type'"
    exit 198
  }
- if `:length local crittype' == 0 {
= if 0 == 0 {
- if inlist("`vce'", "cluster", "robust", "svy") {
= if inlist("robust", "cluster", "robust", "svy") {
- local crittype "log pseudolikelihood"
- }

```

```

- else local crittype "log likelihood"
- }
- global ML_waldtest `waldtest'
= global ML_waldtest -1
- if `:length local svy' | `:length local weight' {
= if 0 | 0 {
  if `interactive' {
    SetGlobalVar ML_w : _MLw
  }
  else {
    global ML_w : tempvar
    if ``moptobj' != "" {
      global ML_w `moptobj'$ML_w
    }
  }
}
- if `:length local weight' {
= if 0 {
  if ``moptobj' != "" {
    cap drop $ML_w
  }
  quietly gen double $ML_w `exp' if $ML_samp
  local wtype ``weight'
  local wexp ``exp'
  global ML_wtyp : copy local wtype
  global ML_wexp : copy local wexp
  if "`wtype'" == "iweight" | "`svy'" != "" {
    local zeroweight zeroweight
  }
}
- if inlist("`vce'", "svy", "cluster") {
= if inlist("robust", "svy", "cluster") {
  if `interactive' {
    SetGlobalVar ML_subv : _MLsu
  }
  else {
    global ML_subv : tempvar
    if ``moptobj' != "" {
      global ML_subv `moptobj'$ML_subv
    }
  }
}
- quietly count if $ML_sample
= quietly count if __000007
- if r(N) == 0 {
  error 2000

```

```

}
- if ``weight'' != "" {
= if ``'' != "" {
  quietly count if $ML_sample & $ML_w != 0
  if r(N) == 0 {
    di in gr "all observation have zero weights"
    exit 2000
  }
}
- if `:length local lf0' {
= if 19 {
- if `:list sizeof lf0' != 2 {
= if 2 != 2 {
  di as err "lf0(`lf0') invalid"
  exit 198
}
- global ML_k0 : word 1 of `lf0'
= global ML_k0 : word 1 of 1 -437.385284075989
- global ML_ll_0 : word 2 of `lf0'
= global ML_ll_0 : word 2 of 1 -437.385284075989
- confirm integer number $ML_k0
= confirm integer number 1
- confirm number $ML_ll_0
= confirm number -437.385284075989
- if $ML_k0 < 0 {
= if 1 < 0 {
  di as err "lf0(`lf0') invalid"
  exit 198
}
- }
- if ``obs'' == "" {
= if "" == "" {
- if ``wtype'' != "fweight" {
= if "" != "fweight" {
- quietly count if $ML_sample
= quietly count if __000007
- local obs = r(N)
- }
- else {
  sum $ML_w if $ML_sample, mean
  local obs = r(sum)
}
- }
- if `:length local iterid' {
= if 0 {
  local iterid = trim(bsubstr(```iterid'',1,32))

```



```

    }
    - if `:length local crittype' {
= if 20 {
    - local crittype = trim(bsubstr("`crittype'",1,32))
= local crittype = trim(bsubstr("`log pseudolikelihood'",1,32))
    - }
    - if `:length local derivscale' {
= if 0 {
        confirm matrix `derivscale'
    }
    - if `:length local derivh' {
= if 0 {
        confirm matrix `derivh'
    }
    - local l : strlen local deriv
    - ParseUsemin, `deriv' len(`l')
= ParseUsemin, len(0)
}

> ----- begin mopt.ParseUsemin -----
    - capture syntax [, LEN(integer 0), MIN1(numlist min=2 max=2 m
> issingokay) NOMIN MIN]
    - if (_rc) {
        syntax [, LEN(integer 0) MIN1(string) NOMIN MIN]
    }
    - if ``min1' != "" & ``nomin' != "" {
= if `'' != "" & `'' != "" {
        di as err "options {bf:min()} and {bf:nomin} are mutually ex
> clusive"

        exit 198
    }
    - if ``min1' != "" & ``nomin' != "" {
= if `'' != "" & `'' != "" {
        di as err "options {bf:min} and {bf:nomin} are mutually excl
> usive"

        exit 198
    }
    - if (``min1' != "") {
= if (``' != "") {
        if (_rc) {
            capture confirm matrix `min1'
            if (_rc) {
                di as err "option {bf:min()} requires that either two number
> s or a 1X2 matrix be specified"
                exit 198
            }
        }
    }
}

```

```

        local deriv `min1'
    }
    - else if (```min'' != "") {
= else if (`'' != "") {
        if (`len' > length(```min''')) {
            di as err "option {bf:min()} requires that either two number
> s or a 1X2 matrix be specified"
            exit 198
        }
        local deriv 1
    }
    - else {
    - if (`len' > 0) {
= if (0 > 0) {
        if (`nomin' == "") {
            di as err "option {bf:min()} requires that either two number
> s or a 1X2 matrix be specified"
            exit 198
        }
        else if (`len' > length(```nomin''')) {
            di as err "options {bf:min()} and {bf:nomin} are mutually ex
> clusive"

            exit 198
        }
        else {
            local deriv 0
        }
    }
    - else {
    - local deriv 0
    - }
    - }
    - sreturn local deriv `deriv'
= sreturn local deriv 0
}

> ----- end mopt.ParseUsemin -----

    - local deriv = s(deriv)
    - if `:length local gnwmatrix' {
= if 0 {
        confirm matrix `gnwmatrix'
    }
    - ``v' mata: Mopt_model()
= version 18: mata: Mopt_model()
}

> ----- begin mopt_check_program -----

    - version 10

```

```

- args name
- capture confirm name `name'
= capture confirm name poiss_lf
- if c(rc) {
  di as err "unrecognized command:  `name' invalid command na
> me"

  exit 199
}
- capture which `name'
= capture which poiss_lf
- if !c(rc) {
- exit

-----
> ----- end mopt_check_program -----
- global ML_stat model
- if !`interactive' {
= if !0 {
- `vv' MaxMin `maxmin', nooutput `score' `options' moptobj(`mopt
> obj')
= version 18: MaxMin maximize, nooutput  search(off) moptobj()

-----
> ----- begin mopt.MaxMin -----
- local vv : di "version " string(_caller()) ":"
- version 12
- ChkMacros

-----
> ----- begin mopt.ChkMacros -----
- if "$ML_stat" != "model" {
= if "model" != "model" {
  di as err "you must issue -ml model- first"
  exit 198
}
- capture confirm var $ML_samp $ML_w $ML_subv
= capture confirm var __000007 __000007
- if c(rc) {
  di as err "Since issuing the -ml model- statement, you hav
> e done something to drop" _n "temporary variables ml added to your data. Yo
> u must start again."
  Clear
  exit 111
}

-----
> ----- end mopt.ChkMacros -----
- syntax name(name=maxmin) [, NEGH TOLerance(numlist max=1 >=0
> ) LTOLerance(numlist max=1 >=0) NONRTOLerance NRTOLerance(numlist max=1 >=0)
> QTOLerance(numlist max=1 >=0) GTOLerance(numlist max=1 >=0) ITERate(numlist

```

```

> max=1 >=0) NOTCONCAVE(numlist max=1 >=0) ndami DIFFicult HALFSTEPonly D0OPT
> SEArch(name) Repeat(passthru) Bounds(string) SCore(string) noWARNing noCLEA
> R NOLOg LOg DOTs SHOWTOLerance SHOWNRtolerance SHOWSTEP TRace COEFDiffs GRAD
> ient HESSian noSKIPline noOUTput Level(cilevel) moptobj(string) * ]
    - if `:length local nrtolerance' {
    = if 0 {
      opts_exclusive "nrtolerance() `nonrtolerance'"
    }
    - if `:length local qtolerance' {
    = if 0 {
      opts_exclusive "qtolerance() `nonrtolerance'"
    }
    - opts_exclusive "`nonrtolerance' `shownrtolerance'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
      local ++n
      gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - if `:length local shownrtolerance' {
    = if 0 {
      local showtolerance showtolerance
    }
    - if "$ML_preserve" != "no" {
    = if "no" != "no" {
      if `:length local score' {
        di as smcl as err "May not specify {bf:score()} option unless
> s" _n "      a) estimation subsample is the entire data in memory, or" _n "
>      b) you specify option {bf:nopreserve} on {bf:ml model} statement (meaning
> " _n "      your evaluation program explicitly restricts itself to obs" _
> n "      for which {bf:$}" "{bf:ML_samp==1}."
        exit 198
      }
      preserve

```

```

        local N = c(N)
        quietly keep if $ML_sample
        if (`N' != c(N)) {
            mata: Mopt_init_regetviews()
        }
    }
- if `:length local score' {
= if 0 {
    SetupScore ``$ML_evaltype'' ``score'' $ML_n $ML_dim
    local scores ``s(scores)''
}
- _parse_iterlog, `nolog' `log'
= _parse_iterlog,

```

---

```

> ----- begin _parse_iterlog -----
    - version 16
    - syntax [, NOLOG LOG]
    - if ``log'`nolog' != "" {
= if "" != "" {
    opts_exclusive ``log' `nolog'
    }
    - else if ``c(iterlog)' == "on" {
= else if "on" == "on" {
    - local log log
    - }
    - else {
        local nolog nolog
    }
    - sreturn local log `log'
= sreturn local log log
    - sreturn local nolog `nolog'
= sreturn local nolog

```

---

```

> ----- end _parse_iterlog -----
    - local log ``s(nolog)''
= local log ""
    - local tropts `log' `dots' `showtolerance' `showstep' `trace'
> `coefdiffs'
= local tropts
    - if ``tropts' != "nolog" {
= if "" != "nolog" {
    - local tropts `tropts' `gradient' `hessian'
= local tropts
    - if ``skipline' == "" {
= if "" == "" {
    - if ``iterate' != "0" | `:length local tropts' {

```

```

= if "" != "0" | 0 {
- di

- }
- }
- }
- else {
    local gradient
    local hessian
    local nolog nolog
    }
- if "`search'" != "off" & "`iterate'" != "0" {
= if "off" != "off" & "" != "0" {
    if "`search'" == "quietly" {
        local sopts nolog
    }
    else {
        if `:length local tropts' {
            if "`tropts'" != "nolog" {
                local sopts trace
            }
            else local sopts nolog
        }
        if "`search'" == "norescale" {
            local sopts `sopts' norescale
        }
        Search `bounds', nopreserve `sopts' `repeat' `maxmin'
    }
- tempname value b V iV
- `vv' mata: Mopt_maxmin()
= version 18: mata: Mopt_maxmin()

-----
> ----- begin poiss_lf -----
    - version 6
    - args todo b lnf g H score
    - tempvar xb
    - mlevel `xb' = `b'
    = mlevel __00000Q = __00000K
    - mlsum `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
    = mlsum __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
    - if `todo' == 0 | `lnf'==. {
    = if 2 == 0 | __00000L==. {
        exit
    }

```

```

- qui replace `score' = $ML_y1 - exp(`xb') if $ML_samp
= qui replace __000000 = y_timing_max - exp(__00000Q) if __0
> 00007

- $ML_ec mlvecsum `lnf' `g' = `score'
= * mlvecsum __00000L __00000M = __000000
- if `todo' == 1 | `lnf'==. {
= if 2 == 1 | __00000L==. {
    exit
  }
- mlmatsum `lnf' `H' = exp(`xb')
= mlmatsum __00000L __00000N = exp(__00000Q)

> _____ end poiss_lf _____
Iteration 0:  Log pseudolikelihood =  -364.7159

> _____ begin poiss_lf _____

- version 6
- args todo b lnf g H score
- tempvar xb
- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsum `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsum __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 0 == 0 | __00000L==. {
- exit

> _____ end poiss_lf _____

> _____ begin poiss_lf _____

- version 6
- args todo b lnf g H score
- tempvar xb
- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsum `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsum __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 0 == 0 | __00000L==. {
- exit

> _____ end poiss_lf _____

> _____ begin poiss_lf _____

```

```

- version 6
- args todo b lnf g H score
- tempvar xb
- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsun `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsun __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 2 == 0 | __00000L==. {
  exit
}
- qui replace `score' = $ML_y1 - exp(`xb') if $ML_samp
= qui replace __000000 = y_timing_max - exp(__00000Q) if __0
> 00007

- $ML_ec mlevelsum `lnf' `g' = `score'
= * mlevelsum __00000L __00000M = __000000
- if `todo' == 1 | `lnf'==. {
= if 2 == 1 | __00000L==. {
  exit
}
- mlevelsum `lnf' `H' = exp(`xb')
= mlevelsum __00000L __00000N = exp(__00000Q)
_____
> _____ end poiss_lf _____
Iteration 1: Log pseudolikelihood = -364.69076
_____
> _____ begin poiss_lf _____

- version 6
- args todo b lnf g H score
- tempvar xb
- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsun `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsun __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 0 == 0 | __00000L==. {
- exit
_____

> _____ end poiss_lf _____
_____
> _____ begin poiss_lf _____

- version 6
- args todo b lnf g H score
- tempvar xb

```



```

- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsun `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsun __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 0 == 0 | __00000L==. {
- exit
}
}

> _____ end poiss_lf _____
> _____ begin poiss_lf _____
- version 6
- args todo b lnf g H score
- tempvar xb
- mlevel `xb' = `b'
= mlevel __00000Q = __00000K
- mlsun `lnf' = -exp(`xb') + `xb'*$ML_y1 - lngamma($ML_y1+1)
= mlsun __00000L = -exp(__00000Q) + __00000Q*y_timing_max -
> lngamma(y_timing_max+1)
- if `todo' == 0 | `lnf'==. {
= if 2 == 0 | __00000L==. {
exit
}
- qui replace `score' = $ML_y1 - exp(`xb') if $ML_samp
= qui replace __000000 = y_timing_max - exp(__00000Q) if __0
> 00007

- $ML_ec mlvecsum `lnf' `g' = `score'
= * mlvecsum __00000L __00000M = __000000
- if `todo' == 1 | `lnf'==. {
= if 2 == 1 | __00000L==. {
exit
}
- mlmatsum `lnf' `H' = exp(`xb')
= mlmatsum __00000L __00000N = exp(__00000Q)
}

> _____ end poiss_lf _____
Iteration 2: Log pseudolikelihood = -364.69076
> _____ begin mopt_post _____
- args b V C wgt touse depvar
- if `:length local touse' {
= if 8 {
- tempname esamp
- quietly gen byte `esamp' = `touse'
= quietly gen byte __00000S = __000007

```

```

- }
- ereturn post `b' `V' `C' `wgt', esample(`esamp') depname(`
> depvar') buildfvinfo
= ereturn post __00000Q __00000R [], esample(__00000S) depn
> ame(y_timing_max) buildfvinfo


---


> 

---

 end mopt_post 

---


- ereturn scalar k_eq_model = abs($ML_waldtest)
= ereturn scalar k_eq_model = abs(-1)
- if `:length global ML_ll_0' {
= if 17 {
- ereturn scalar ll_0 = $ML_ll_0
= ereturn scalar ll_0 = -437.385284075989
- if `:length global ML_rank0' {
= if 0 {
ereturn scalar rank0 = $ML_rank0
ereturn scalar df_m = e(rank) - $ML_rank0
}
- else ereturn scalar df_m = e(rank) - $ML_k0
= else ereturn scalar df_m = e(rank) - 1
- }
- if !`:length global ML_ll_0' | inlist(e(vce), "robust", "clu
> ster", "linearized") {
= if !17 | inlist(e(vce), "robust", "cluster", "linearized") {
- if "$ML_svy" == "" {
= if "" == "" {
- _prefix_model_test, minimum


---


> 

---

 begin _prefix_model_test 

---


- version 9
- syntax [name(name=cmd)] [, svy noADJust noSVYadjust MINimu
> m ]
- if (inlist("`cmd'", "asmixlogit", "cmmixlogit", "cmxtmixlogit
> ")) {
= if (inlist("", "asmixlogit", "cmmixlogit", "cmxtmixlogit")) {
_asmixtest, `svy' `adjust' `svyadjust'
exit
}
- local K = 1
- if "`e(k_eq_model)'" != "" {
= if "1" != "" {
- capture confirm integer number `e(k_eq_model)'
= capture confirm integer number 1
- if ! c(rc) {
- local K = e(k_eq_model)
- }

```

```

- }
- if "`e(k_eq_model_skip)'" != "" {
= if "" != "" {
    capture confirm integer number `e(k_eq_model_skip)'
    if ! c(rc) {
        local skip = e(k_eq_model_skip)
    }
}
- ereturn local df_m
- ereturn local F
- ereturn local chi2
- ereturn local chi2type
- ereturn local p
- if (`K' == 0) {
= if (1 == 0) {
    exit
}
- forval i = 1/`K' {
= forval i = 1/1 {
- if !`:list i in skip' {
= if !0 {
- local spec "`spec' ([#`i'])"
= local spec " ([#1])"
- }
- }
- tempname cns omit
- capture matrix `cns' = get(Cns)
= capture matrix __00000S = get(Cns)
- if (c(rc)) local cns
- if "`cns'" != "" {
= if "" != "" {
    local k_autoCns = e(k_autoCns)
    if missing(`k_autoCns') {
        local k_autoCns 0
    }
    if rowsof(`cns') == `k_autoCns' {
        local cns
    }
}
- if "`svy'" != "" & "`adjust'`svyadjust'" != "" {
= if "" != "" & "" != "" {
    local adjust nosvyadjust
}
- else local adjust
- _ms_omit_info e(b), nocons
- matrix `omit' = r(omit)

```

```

= matrix __00000T = r(omit)
- capture test `spec', `adjust'
= capture test ([#1]),

> ----- begin test -----
- if "`e(cmd)'" == "anova" | "`e(cmd)'" == "manova" {
= if "mopt" == "anova" | "mopt" == "manova" {
  AnovaTestChooser `0'
  return add
  exit
}
- if "`e(cmd)'" == "mixed" {
= if "mopt" == "mixed" {
  MixedTestChooser `0'
  return add
  exit
}
- if _caller() < 8 {
  _test `0'
  return add
  exit
}
- is_svy

> ----- begin is_svy -----
- version 8
- syntax [, Regression]
- if ``e(prefix)'" == "svy" {
= if `""' == "svy" {
  return scalar is_svy = 1
}
- else if "`regression'" == "" {
= else if "" == "" {
- return scalar is_svy = ("`e(N_psu)'" != "")
= return scalar is_svy = (" != ")
- }
- else {
  capt mat list e(b)
  local rc = _rc
  return scalar is_svy = ("`e(N_psu)'" != "") & (`rc' ==
> 0)

}

> ----- end is_svy -----
- local is_svy = r(is_svy)
- if replay() & `is_svy' == 0 {

```

```

= if replay() & 0 == 0 {
  _test `0'
  return add
  exit
}
- version 8
- WaldTest `is_svy' `0'
= WaldTest 0 ([#1]),

> ----- begin test.WaldTest -----
- gettoken is_svy 0 : 0
- if ``is_svy' == "1" {
= if "0" == "1" {
  if ``e(complete)' == "available" {
    di as err ``must run svy command with "complete" "' "o
> ption before using this command"
    exit 301
  }
}
- gettoken tok : 0, parse(" (,") bind
- local single single
- local zero ``0'
= local zero `` ([#1]), "'
- if ``tok' == "(" {
= if ``('' == "(" {
- local single
- while ``tok' == "(" {
= while ``('' == "(" {
- gettoken tok 0 : 0, parse(" (,") match(paren) bind
- local eqns ``eqns' ``tok'
= local eqns `` [#1]'
- gettoken tok : 0, parse(" (,") bind
- }
- while ``tok' == "(" {
= while ``",'' == "(" {
  gettoken tok 0 : 0, parse(" (,") match(paren) bind
  local eqns ``eqns' ``tok'
  gettoken tok : 0, parse(" (,") bind
}
- if ``tok' != "," & ``tok' != "" {
= if ``",'' != "," & ``",'' != "" {
  local single single
}
- }
- if ``single'=="single" {
= if ""=="single" {

```

```

        local eqns
        local 0 ``zero'''
        while ``tok''' != "" & ``tok''' != "," {
        gettoken tok 0 : 0, parse(" ,") bind
        local eqns ``eqns' `tok'''
        gettoken tok : 0, parse(" ,") bind
        }
        local eqns ``''eqns'''''
        }
        - syntax [, Accumulate noSVYadjust common CONStant MINim
> um Mtest Mtest2(passthru) noTest COEF matvlc(passthru) DF(numlist max=1 >0)
> ]

        - if ``df''' != "" {
        = if "" != "" {
            local dof = `df'
        }
        - else if ``e(df_r)''' != "" {
        = else if "" != "" {
            local dof = e(df_r)
        }
        - else {
        - local dof = .
        - }
        - if ``mtest'`mtest2''' != "" {
        = if ``''' != "" {
            if inlist("`e(cmd)'", "svymean", "svytotal", "svyratio
> ") {

            di as err ``mtest option not allowed after `e(cmd)'''
            exit 301
        }
        _mtest syntax, `mtest' `mtest2'
        local mtmethod `r(method)'
        }
        - if ``mtmethod''' != "" {
        = if "" != "" {
            if ``test''' != "" {
            di as err "options mtest and notest may not be combine
> d"

            exit 198
        }
        if ``accumulate''' != "" {
            di as err "options mtest and accumulate may not be com
> bined"

            exit 198
        }
        }
    }

```

```

- if `is_svy' {
= if 0 {
  if ``e(cmd)''' == "margins" {
    local vce ``e(model_vce)'''
  }
  else local vce ``e(vce)'''
  if inlist(```vce'''', "bootstrap", "sdr") {
    if missing(`dof') {
      local svyadjust nosvyadjust
    }
  }
}
- if !`is_svy' & ``svyadjust' != "" {
= if !0 & "" != "" {
  di as err "option nosvyadjust is allowed only with svy

> data"

  exit 198
}
- local svy_adjust = `is_svy' & ``svyadjust' == ""
= local svy_adjust = 0 & "" == ""
- if `svy_adjust' & missing(`dof') & (missing(e(N_psu))
> | missing(e(N_strata))) {
= if 0 & missing(.) & (missing(e(N_psu)) | missing(e(N_s
> trata))) {

  di as err "impossible to adjust p-values"
  exit 198
}
- if `is_svy' & ``test''' == "" {
= if 0 & `''' == "" {
  if ``svyadjust' == "" {
    di _n as txt "Adjusted Wald test"
  }
  else di _n as txt "Unadjusted Wald test"
}
- local tmp : subinstr local eqns "==" "", all count(lo

> cal nch)

- if `nch' > 0 {
= if 0 > 0 {
  di as err "==" invalid operator"
  exit 198
}
- local eqns : subinstr local eqns "==" "=", all
- ExpandExpns ``eqns''' ``common' ``constant'
= ExpandExpns ``#[#1]''' "" ""

> _____ begin test.ExpandExpns _____

```

```

- args expns common constant
- gettoken exp expns : expns
- while ``exp'' != "" {
= while ``[#1]'' != "" {
- Expand ``exp'' ``common'' ``constant''
= Expand ``[#1]'' "" ""

> ----- begin test.Expand -----
- args exp common constant
- local junk : substr local exp "=" "=", count(loc
> al nch) all
- if `nch' <= 1 {
= if 0 <= 1 {
- return local explist ``exp''
= return local explist ``[#1]''
- return local nexp 1
- exit

> ----- end test.Expand -----
- local explist ``explist' `r(explist)''
= local explist ``[#1]''
- local nexp = `nexp' + `r(nexp)'
= local nexp = + 1
- gettoken exp expns : expns
- }
- while ``exp'' != "" {
= while ``'' != "" {
Expand ``exp'' ``common'' ``constant''
local explist ``explist' `r(explist)''
local nexp = `nexp' + `r(nexp)'
gettoken exp expns : expns
}
- return local explist ``explist''
= return local explist ``[#1]''
- return local nexp `nexp'
= return local nexp 1

> ----- end test.ExpandExpns -----
- local explist ``r(explist)''
= local explist ``[#1]''
- local nexp `r(nexp)'
= local nexp 1
- local acc `accumulate'
= local acc
- foreach exp of local explist {
- quietly _test `exp' , `acc' `common' `constant' notest

```



```

= quietly _test [#1] ,      notest
- local acc acc
- }
- if "`test'" == "notest" {
= if "" == "notest" {
    _test , notest
    exit
    }
- quietly _test , `matvlc'
= quietly _test ,
- if `dof' < . {
= if . < . {
    local testtype F
    tempname F df1 df2 p drop
    if "`r(F)'" != "" {
        scalar `F' = r(F)
    }
    else {
        scalar `F' = r(chi2)/r(df)
    }
    scalar `df1' = r(df)
    scalar `df2' = `dof'
    scalar `p' = Ftail(`df1', `df2', `F')
    scalar `drop' = r(drop)
    }
- else {
- local testtype chi2
- tempname chi2 df1 p drop
- scalar `chi2' = r(chi2)
= scalar __00000U = r(chi2)
- scalar `df1' = r(df)
= scalar __00000V = r(df)
- scalar `p' = r(p)
= scalar __00000W = r(p)
- scalar `drop' = r(drop)
= scalar __00000X = r(drop)
- }
- local i 1
- while "`r(dropped_`i')'" != "" {
= while "" != "" {
    local dropped_`i' `r(dropped_`i')'
    local ++i
    }
- if `svy_adjust' {
= if 0 {
    if "`df'" != "" {

```

```

local svy_d = `df'
}
else if !missing(e(df_r)) {
local svy_d = e(df_r)
}
else local svy_d = e(N_psu)-e(N_strata)
if missing(`svy_d') {
di as err "internal error in adjustment for svy"
exit 10001
}
if "`testtype'" == "F" {
scalar `F' = ((`svy_d'-`df1'+1)/`svy_d') * `F'
}
else {
tempname F df2
scalar `F' = ((`svy_d'-`df1'+1)/(`svy_d'*`df1')) * `ch
> i2'

}
scalar `df2' = `svy_d' - `df1' + 1
if `df2' < 0 {
scalar `F' = .
}
scalar `p' = Ftail(`df1', `df2', `F')
local testtype F
}
- if "`mtmethod'" != "" {
= if "" != "" {
tempname R r Rr V b VR tm Wald
mat `V' = e(V)
mat `b' = e(b)
mat `Rr' = get(Rr)
local nb = rowsof(`V')
local nr = rowsof(`Rr')
mat `R' = `Rr'[1...,1..`nb']
mat `VR' = `R'*`V'*`R''
mat `r' = `R'*`b'' - `Rr'[1...,`= `nb'+1']
mat `Wald' = `r'*syminv(`VR')*`r'
mat `tm' = J(`nr', 3, .)
mat colnames `tm' = `testtype' df p
forvalues it = 1 / `nr' {
mat `Wald' = `r'[`it',1]^2 / `VR'[`it',`it']
mat `tm'[`it',1] = `Wald'[1,1]
mat `tm'[`it',2] = 1
if `svy_adjust' {
mat `tm'[`it',3] = Ftail(1,`svy_d',`tm'[`it',1])
}
}

```

```

else if "`testtype'" == "chi2" {
mat `tm'[`it',3] = chi2tail(1,`tm'[`it',1])
}
else {
mat `tm'[`it',3] = Ftail(1,`df2',`tm'[`it',1])
}
}
if "`mtmethod'" != "noadjust" {
_mtest adjust `tm', mtest(`mtmethod') pindex(3) append
mat `tm' = r(result)
local pindex 4
}
else local pindex 3
}
- _test, notest
- local i 1
- while "`dropped_`i'" != "" {
= while "" != "" {
di as txt "          Constraint `dropped_`i' dropped"
local ++i
}
- if "`mtmethod'" == "" {
= if "" == "" {
- di
- if "`testtype'" == "chi2" {
= if "chi2" == "chi2" {
- di as txt _col(12) "chi2(" %3.0f `df1' ") =" as res %8
= di as txt _col(12) "chi2(" %3.0f __00000V ") =" as res
- di as txt _col(10) "Prob > chi2 = " as res %8.4f `p'
= di as txt _col(10) "Prob > chi2 = " as res %8.4f __00
> 000W
- }
- else {
di as txt "          F(" %3.0f `df1' ", " %6.0f `df2' ") =
> " as res %8.2f `F'
di as txt _col(13) "Prob > F =" as res %10.4f `p'
}
- }
- else {
local nexp = rowsof(`tm')
local nc = colsof(`tm')
if "`testtype'" == "F" {
local d "F(df,`=df2'')"
local dip "p > F"

```

```

    }
    else {
        local d chi2
        local dip "p > chi2"
    }
    di
    di as txt "{hline 7}{c TT}{hline 30}"
    di as txt "          {c |}{ralign 12:`d'}`df{ralign 1
> 1:`dip'}"

    di as txt "{hline 7}{c +}{hline 30}"
    forvalues it = 1 / `nexp' {
        di as txt "    (`it') {c |}" as res _col(12) %9.2f `tm'
> [`it',1] _col(22) %6.0f `tm' [`it',2] _col(33) %6.4f `tm' [`it',`pindex'] as t
> xt "*"

    }
    di as txt "{hline 7}{c +}{hline 30}"
    di as txt "    All {c |}" as res _col(12) %9.2f ``testtt
> ype'' _col(22) %6.0f `df1' _col(33) %6.4f `p'
    di as txt "{hline 7}{c BT}{hline 30}"
    _mtest footer 1 "`mtmethod'" "*"
    }
    - if "`coef'" != "" {
= if "" != "" {
    Table
    }
    - return scalar p = `p'
= return scalar p = __00000W
    - return scalar df = `df1'
= return scalar df = __00000V
    - if "`testtype'" == "F" {
= if "chi2" == "F" {
        return scalar F = `F'
        return scalar df_r = `df2'
    }
    - else return scalar chi2 = `chi2'
= else return scalar chi2 = __00000U
    - return scalar drop = `drop'
= return scalar drop = __00000X
    - local i 1
    - while "`dropped_`i'" != "" {
= while "" != "" {
        return scalar dropped_`i' = `dropped_`i'
        local ++i
    }
    - if "`mtmethod'" != "" {
= if "" != "" {

```

```

        return matrix mtest `tm'
        return local mtmethod `mtmethod'
    }
}

> ----- end test.WaldTest -----
- return add

> ----- end test -----
- if !c(rc) {
- if `:length local minimum' {
= if 7 {
- quietly test, min

> ----- begin test -----
- if "`e(cmd)'" == "anova" | "`e(cmd)'" == "manova" {
= if "mopt" == "anova" | "mopt" == "manova" {
    AnovaTestChooser `0'
    return add
    exit
}
- if "`e(cmd)'" == "mixed" {
= if "mopt" == "mixed" {
    MixedTestChooser `0'
    return add
    exit
}
- if _caller() < 8 {
    _test `0'
    return add
    exit
}
- is_svy

> ----- begin is_svy -----
- version 8
- syntax [, Regression]
- if ``e(prefix)'" == "svy" {
= if `"' == "svy" {
    return scalar is_svy = 1
}
- else if "`regression'" == "" {
= else if "" == "" {
- return scalar is_svy = ("`e(N_psu)'" != "")
= return scalar is_svy = (" != ")
- }
- else {

```

```

        capt mat list e(b)
        local rc = _rc
        return scalar is_svy = ("`e(N_psu)'" != "") & (`rc' ==
> 0)
    }
}
----- end is_svy -----
>
- local is_svy = r(is_svy)
- if replay() & `is_svy' == 0 {
= if replay() & 0 == 0 {
- _test `0'
= _test , min
- return add
- exit
}
----- end test -----
>
- }
- ereturn scalar df_m = r(df)
- local drop = r(drop)
- if `drop' {
= if 0 {
    local drop 0
    local i 1
    local idrop ``r(dropped_`i')'''
    while ``idrop'" != "" {
        if !`omit'[1,`idrop'] {
            local ++drop
        }
        local ++i
        local idrop ``r(dropped_`i')'''
    }
}
- if `drop' == 0 | "`cns'" != "" {
= if 0 == 0 | "" != "" {
- if !missing(e(df_r)) {
    ereturn scalar F = r(F)
}
- else {
- ereturn scalar chi2 = r(chi2)
- ereturn local chi2type Wald
- }
- ereturn scalar p = r(p)
- }
- else {
    if !missing(e(df_r)) {
        ereturn scalar F = .
    }
}

```

```

    }
    else {
        ereturn scalar chi2 = .
        ereturn local chi2type Wald
    }
    ereturn scalar p = .
}
- }
- else {
    if missing(e(df_r)) {
        ereturn local chi2type Wald
    }
    ereturn scalar df_m = 0
}
}


---


> 

---

 end _prefix_model_test 

---


- }
- else {
    _prefix_model_test, $ML_svy $ML_noadj
}
- }
- else if `:length global ML_ll_0' {
= else if 17 {
    if ``"maxmin" == "minimize" {
        ereturn scalar chi2 = -2*(e(ll)-e(ll_0))
    }
    else {
        ereturn scalar chi2 = 2*(e(ll)-e(ll_0))
    }
    ereturn scalar p = chi2tail(e(df_m), e(chi2))
    ereturn local chi2type "LR"
}
- if `:length global ML_wexp' {
= if 0 {
    ereturn local wtype ``$ML_wtyp"
    ereturn local wexp ``$ML_wexp"
}
- if `:length global ML_group' {
= if 0 {
    ereturn local group ``$ML_group"
    if ``"e(clustvar)" == "$ML_grp" & "$ML_grp" != "" {
        ereturn local clustvar ``$ML_group"
    }
}
- ereturn local cmd "ml"
- if `:length local score' {

```

```

= if 0 {
    mata: Mopt_score(0)
    ereturn local scorevars `scores'
}
- if "$ML_preserve" != "no" {
= if "no" != "no" {
    restore
    if "`e(prefix)'" == "svy" & e(N_strata_omit) {
        tempname v
        quietly _svy2 $ML_sample if $ML_sample, type(total) svy v(`v
> ') subpop(`e(subpop)') touse($ML_sample)
    }
    ereturn repost, esample($ML_sample)
}
- if ``moptobj' != "" {
= if `""' != "" {
    local clear noclear
}
- if !':length local clear' {
= if !0 {
- Clear
}

> ----- begin mopt.Clear -----
- syntax [, NOTANOPTION]
- capture drop $ML_w
= capture drop __000007
- capture drop $ML_samp
= capture drop __000007
- capture drop $ML_sample
= capture drop __000007
- capture drop $ML_subv
= capture drop
- capture drop $ML_grp
= capture drop
- mata: Mopt_drop_external()
- macro drop ML_*

> ----- end mopt.Clear -----
- }
- else {
    capture confirm var $ML_sample
    if c(rc) {
        quietly gen byte $ML_sample = e(sample)
    }
}
- if `':length local output' == 0 {

```



```

        = if 8 == 0 {
            Display, level(`level') `options'
        }
    }
> ----- end mopt.MaxMin -----
    - }
> ----- end mopt.Model -----
    - if c(rc) {
        local rc = c(rc)
        Clear
        exit `rc'
    }
    - exit
> ----- end mopt -----
    - exit
> ----- end ml -----
    - if "`score'" != "" {
= if "" != "" {
    label var `scvar' "Score index from poisson"
    rename `scvar' `score'
    est local scorevars `score'
}
    - est scalar r2_p = 1 - e(ll)/e(ll_0)
    - est hidden local gof "poiss_g"
    - est local offset "`offvar'"
= est local offset ""
    - est local offset1
    - est local estat_cmd "poisson_estat"
    - est hidden local marginsderiv default N IR
    - est local predict "poisso_p"
    - est local cmd "poisson"
    - global S_E_ll = e(ll)
    - global S_E_ll0 `e(ll_0)'
= global S_E_ll0 -437.385284075989
    - global S_E_chi2 = e(chi2)
    - global S_E_mdf = e(df_m)
    - global S_E_pr2 `e(r2_p)'
= global S_E_pr2 .1662024978318156
    - global S_E_nobs = e(N)
    - global S_E_tdf = e(N)
    - global S_E_off `e(offset)'
= global S_E_off
    - global S_E_depv `e(depvar)'

```

```

= global S_E_depv y_timing_max
- global S_E_cmd `e(cmd)'
= global S_E_cmd poisson
- if "`display'"==" " {
= if "nodisplay"==" " {
    Display, `irr' level(`level') `diopts'
}
- error `e(rc)'
= error 0

```

---

```

> ----- end poisson.Estimate -----
- version 10: ereturn local cmdline `"poisson `0'"
= version 10: ereturn local cmdline `"poisson y_timing_max cb_authors_ex
> t estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open []
> if __000000, nodisplay robust"

```

---

```

> ----- end poisson -----
- tempname bp
- mat `bp' = e(b)
= mat __000002 = e(b)
- if "`lrtest'"==" " {
= if "nolrtest"==" " {
    tempname llp
    scalar `llp' = e(ll)
}
- }
- else if "`from'"==" " {
= else if `""'==" " {
    `vvpoi' capture poisson `y' `xvars' [`weight'`exp'] if `touse', `offopt'
> `constan' iter(1)
    tempname bp
    mat `bp' = e(b)
}
- if "`constan'"==" " & "`from'"==" " {
= if ""==" " & `""'==" " {
- if "`offset'"!=" " {
= if ""!=" " {
    SolveC `y' `offset' [`weight'`exp'] if `touse', mean(`mean')
    local c = r(_cons)
}
- else local c = ln(`mean')
= else local c = ln(__000001)
- tempname b0 ll0
- if `c'<. {
= if .8283739677697126<. {
- mat `b0' = (`c', 0)

```

```

= mat __000003 = (.8283739677697126, 0)
- }
- else mat `b0' = (0, 0)
= else mat __000003 = (0, 0)
- mat colnames `b0' = `y':_cons ln`parm':_cons
= mat colnames __000003 = y_timing_max:_cons lnalpha:_cons
- if "`weight'"=="aweight" {
= if ""=="aweight" {
    local wt `[aw`exp']"
  }
- else if "`weight'"!="" {
= else if ""!="" {
    local wt `[iw`exp']"
  }
- `nohead' di in gr _n "Fitting constant-only model:"
= di in gr _n "Fitting constant-only model:"

```

Fitting constant-only model:

```

- `vv' ml model `mm' `prog' (`yname': `y'=, `constan' `offopt') /ln`parm'
> if `touse' `wt', collinear missing max nooutput nopreserve wald(0) init(`b0'
> , copy) search(off) `mlopts' `robust' nocnsnotes `negh'
= version 18: ml model e2 nbreg_lf (y_timing_max: y_timing_max=, ) /lnalp
> ha if __000000 , collinear missing max nooutput nopreserve wald(0) init(__00
> 0003, copy) search(off) robust nocnsnotes negh

```

---

```

> ----- begin ml -----
- if _caller()>=11 {
- mopt `0'
= mopt model e2 nbreg_lf (y_timing_max: y_timing_max=, ) /lnalpha if __
> 000000 , collinear missing max nooutput nopreserve wald(0) init(__000003, co
> py) search(off) robust nocnsnotes negh

```

---

```

> ----- begin mopt -----
- local vv : di "version " string(_caller()) ":"
- version 12
- gettoken subcmd 0 : 0, parse(" ,")
- local lcmd : length local subcmd
- if "`subcmd'" == "clear" {
= if "model" == "clear" {
    Clear
    exit
  }
- if ``"subcmd"' == "hold" {
= if ``model"' == "hold" {
    ml_hold `0'
    exit

```

```

}
- if ``subcmd''' == "unhold" {
= if "model" == "unhold" {
  ml_unhold `0'
  exit
}
- if "subcmd" == "score" {
= if "model" == "score" {
  if _caller() > 12 {
    `vv' Score `0'
  }
  else {
    Score `0'
  }
  exit
}
- if "subcmd" == "count" {
= if "model" == "count" {
  Count `0'
  exit
}
- if "subcmd" == "trace" {
= if "model" == "trace" {
  Trace `0'
  exit
}
- if "subcmd" == bsubstr("graph",1,max(2,`lcmd')) {
= if "model" == bsubstr("graph",1,max(2,5)) {
  Graph `0'
  exit
}
- if "subcmd" == "init" {
= if "model" == "init" {
  Interactive 0
  `vv' Init `0'
  exit
}
- if "subcmd" == bsubstr("maximize",1,max(3,`lcmd')) {
= if "model" == bsubstr("maximize",1,max(3,5)) {
  Interactive 1
  `vv' MaxMin maximize `0'
  exit
}
- if "subcmd" == bsubstr("minimize",1,max(3,`lcmd')) {
= if "model" == bsubstr("minimize",1,max(3,5)) {
  Interactive 1

```

```

`vv' MaxMin minimize `0'
exit
}
- if "`subcmd'" == bsubstr("display",1,max(2,`lcmd')) {
= if "model" == bsubstr("display",1,max(2,5)) {
  Display `0'
  exit
}
- if "`subcmd'" == bsubstr("footnote",1,max(4,`lcmd')) {
= if "model" == bsubstr("footnote",1,max(4,5)) {
  ml_footnote `0'
  exit
}
- if "`subcmd'" == "," {
= if "model" == "," {
  Display, `0'
  exit
}
- if "`subcmd'" == bsubstr("model",1,max(3,`lcmd')) {
= if "model" == bsubstr("model",1,max(3,5)) {
- `vv' capture noi Model `0'
= version 18: capture noi Model e2 nbreg_lf (y_timing_max: y_timing_m
> ax=, ) /lnalpha if __000000 , collinear missing max nooutput nopreserve wal
> d(0) init(__000003, copy) search(off) robust nocnsnotes negh

```

---

```

> ----- begin mopt.Model -----
- local version = string(_caller())
- local vv : di "version `version':"
= local vv : di "version 18:"
- version 12
- syntax anything(id="model" name=spec) [if] [in] [fw aw pw iw] [, B
> RACKET COLLinear MISSing CONTinue lf0(string) INIT(string) CONSTraints(strin
> g) noCNSNOTes VCE(passthru) NEGh NOPREserve TECHnique(string) ITERID(string)
> CRITtype(string) OBS(numlist max=1 >0) TITle(string) WALDtest(integer -1) S
> VY SUBpop(string) SRSsubpop noSVYadjust SCore(passthru) noOUTput noSCVARS GR
> OUP(varname) ITERPROLOG(string) DERIVPROLOG(string) DERIVH(string) DERIVSCAL
> E(string) DERIV(string) nrderiv GNWMATrix(string) KAUXiliary(numlist int >=0
> ) MAXimize MINimize USERinfo(namelist) PUPDATED Robust CLuster(passthru) mop
> tobj(string) USEVIEWS * ]
- if ``moptobj''' == "" {
= if `""' == "" {
- Clear

```

---

```

> ----- begin mopt.Clear -----
- syntax [, NOTANOPTION]
- capture drop $ML_w

```

```

= capture drop
- capture drop $ML_samp
= capture drop
- capture drop $ML_sample
= capture drop
- capture drop $ML_subv
= capture drop
- capture drop $ML_grp
= capture drop
- mata: Mopt_drop_external()
- macro drop ML_*

```

---

```

> ----- end mopt.Clear -----
- }
- local maxmin `minimize' `maximize'
= local maxmin maximize
- opts_exclusive "`maxmin'"
= opts_exclusive "maximize"

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts maximize
- local n 0
- while ``opts' != "" {
= while "maximize" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts' != "" {
= while `"' != "" {
- local ++n
- gettoken item`n' opts : opts, bind
- }
- if `n' < 2 {
= if 1 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- if `:length local maxmin' {
= if 8 {
- local interactive 0
- }
- else local interactive 1

```

```

- _get_diparmopts, diparmopts(`options') soptions syntaxonly
= _get_diparmopts, diparmopts(search(off)) soptions syntaxonly

```

---

```

> ----- begin _get_diparmopts -----
- if _caller() < 8.2 {
  _get_diparmopts_8 `0'
  exit
}
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom plus S
> YNTAXonly * ]
- local goptions `options'
= local goptions
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk ,

```

---

```

> ----- begin _get_diparmopts.SpliceGlobals -----
- syntax name(id="macro name" name=c_diparm) [, diparm(string as
> is) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof''' == "" & ``e(df_r)''' != "" {
= if `''' == "" & `''' != "" {
  capture confirm integer number `e(df_r)'
  if !c(rc) {
    local gdof dof(`e(df_r)')
  }
  capture
}
- else local gdof ``dof'''
= else local gdof `'''
- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)
- if (``diparm''' == "") exit
= if (``''' == "") exit

```

---

```

> ----- end _get_diparmopts.SpliceGlobals -----
- local 0 `", `diparmopts'''
= local 0 `", search(off)'''
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while ``diparm''' != "" {
= while `''' != "" {
  local ++k

```

```

local single 1
local after
capture _on_colon_parse `diparm'
if !c(rc) {
    local single 0
    local diparm ``s(before)''
    local after ``s(after)''
}
capture
SpliceGlobals diparm , diparm(`diparm') `syntaxonly' `goptions'
local diparm`k' ``diparm''
local diparmopts ``diparmopts' : `diparm''
if `single' | ``:list retok after'' == "" {
    local 0 `", `options''
    syntax [, diparm(string asis) * ]
}
else {
    local diparm ``after''
}
}
- if ``soptions'' == "" {
= if ``soptions'' == "" {
    syntax [, diparm(string asis) ]
}
- if ("`execute" == "") {
= if ("" == "") {
- sreturn clear
- sreturn local options ``options''
= sreturn local options ``search(off)''
- sreturn local diparm ``diparmopts''
= sreturn local diparm ``''
- forval i = 1/`k' {
= forval i = 1/0 {
    sreturn local diparm`i' ``diparm`i''
}
- sreturn local k `k'
= sreturn local k 0
- exit

```

---

```

> _____ end _get_diparmopts _____
- forval k = 1/`s(k)' {
= forval k = 1/0 {
    local diparm`k' ``s(diparm`k)''
}
- local ndiparm `s(k)'
= local ndiparm 0

```



```

- local options ``s(options)''
= local options ``search(off)''
- if `interactive' & `:length local options' {
= if 0 & 11 {
    local 0 , `options'
    syntax [, NOTANOPTION]
}
- if `:length local nopreserve' | `:length local score' {
= if 10 | 0 {
- global ML_preserve no
- }
- else global ML_preserve yes
- gettoken evaltype spec : spec
- confirm name `evaltype'
= confirm name e2
- gettoken evaluator spec : spec
- global ML_evaltype : copy local evaltype
- if `:length local srssubpop' {
= if 0 {
    local 0 , srssubpop
    syntax [, NONOPTION]
    exit 198
}
- if `:length local subpop' {
= if 0 {
    local svy svy
}
- if `:length local svy' {
= if 0 {
    opts_exclusive ``vce' `svy' vce
    if `:length local weight' {
        di as err "svy option not allowed with weights"
        exit 198
    }
    global ML_noadj : copy local svyadjust
}
- global ML_svy : copy local svy
- if _caller() >= 15 & c(userversion) >= 15 {
- local fparm "", freeparm"
- }
- local eqspec : copy local spec
- local ieq 0
- while `:length local eqspec' {
= while 42 {
- local ++ieq
- gettoken eq eqspec : eqspec, bind

```

```

- if bsubstr(trim("`eq'"),1,1) == "/" {
= if bsubstr(trim("(y_timing_max: y_timing_max=, )"),1,1) == "/" {
  local eq = bsubstr(trim("`eq'"),2,.)
  local spec : subinstr local spec "/"`eq'" "("("`eq'":`fparm')"'', wor
> d
  }
- else {
- gettoken p1 p2 : eq, parse("(")
- local p2 = trim("`p2'")
= local p2 = trim("(y_timing_max: y_timing_max=, )")
- if bsubstr("`p2'",1,1) == "," {
= if bsubstr("y_timing_max: y_timing_max=, )",1,1) == "," {
  gettoken comma p2 : p2, parse(",")
  local p2 = trim("`p2'")
  }
- if "`p1'`p2'" == "(" {
= if "(y_timing_max: y_timing_max=, )" == "(" {
  local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
  }
- }
- }
- while `:length local eqspec' {
= while 9 {
- local ++ieq
- gettoken eq eqspec : eqspec, bind
- if bsubstr(trim("`eq'"),1,1) == "/" {
= if bsubstr(trim("/lnalpha"),1,1) == "/" {
- local eq = bsubstr(trim("`eq'"),2,.)
= local eq = bsubstr(trim("/lnalpha"),2,.)
- local spec : subinstr local spec "/"`eq'" "("("`eq'":`fparm')"'', wor
> d
= local spec : subinstr local spec "/lnalpha" "("("lnalpha":, freepar
> m)"', word
- }
- else {
  gettoken p1 p2 : eq, parse("(")
  local p2 = trim("`p2'")
  if bsubstr("`p2'",1,1) == "," {
    gettoken comma p2 : p2, parse(",")
    local p2 = trim("`p2'")
  }
  if "`p1'`p2'" == "(" {
    local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
  }
- }
- }

```

```

- while `:length local eqspec' {
= while 0 {
  local ++ieq
  gettoken eq eqspec : eqspec, bind
  if bsubstr(trim("`eq'"),1,1) == "/" {
    local eq = bsubstr(trim("`eq'"),2,.)
    local spec : subinstr local spec "/"`eq' " "("("`eq':`fparm')", wor
> d
  }
  else {
    gettoken p1 p2 : eq, parse("(")
    local p2 = trim("`p2'")
    if bsubstr("`p2'",1,1) == "," {
      gettoken comma p2 : p2, parse(",")
      local p2 = trim("`p2'")
    }
    if "`p1'`p2'" == "(" {
      local spec : subinstr local spec "`eq'" "(eq`ieq:)", word
    }
  }
}
- if `interactive' {
= if 0 {
  SetGlobalVar ML_sample : _MLtua
}
- else {
- global ML_sample : tempvar
- if ``moptobj' != "" {
= if `"' != "" {
  global ML_sample `moptobj'$ML_sample
}
- }
- if ``moptobj' != "" {
= if `"' != "" {
  cap drop $ML_sample
}
- `vv' mark $ML_sample `if' `in' [`weight'`exp'], `zeroweight'
= version 18: mark __000005 if __000000 [],
- tempname eqlist
- `eqlist' = _eqlist.new, eqopts(NOCONSTant FREEParM) eqargopts(OF
> Fset EXPosure) noneedvarlist numdepvars(0) needequal
= __000006 = _eqlist.new, eqopts(NOCONSTant FREEParM) eqargopts(OF
> Fset EXPosure) noneedvarlist numdepvars(0) needequal
- class {
- instance:
- array n_depvars

```

```

-      string      needequal
-      string      noneedvarlist
-      string      wtypes
-      string      collinear
-      string      nocons
-      string      nocons_ignore
-      double      n_eq
-      string      gl_if
-      string      gl_in
-      string      gl_wgt
-      string      gl_options
-      _optlist     eq_opts_allowed
-class {
-instance:
-      array options = {}
-      array names   = {}
-}
-      array      eq_list
-      _optlist    mark_opts
-      array      mark_list
-      array      mark_name
-      _optlist    smark_opts
-      array      smark_list
-      array      smark_name
-      _optlist    rmcoll_opts
-      array      rmcoll_list
-      array      rmcoll_name
-      _optlist    rmdcoll_opts
-      array      rmdcoll_list
-      array      rmdcoll_name
-}

```

```

> ----- begin _eqlist.new -----
- version 9.2
- ._init
-----
> ----- begin _eqlist._init -----
- version 9.2
- .n_eq = 0
- .gl_if = ""
- .gl_in = ""
- .gl_wgt = ""
- .gl_options = ""
- .eq_list = {}
-----
> ----- end _eqlist._init -----

```

```

- ._setup `0'
= ._setup , eqopts(NOCONSTant FREEParam) eqargopts(OFFset EXPosure)
> noneedvarlist numdepvars(0) needequal

> ----- begin _eqlist._setup -----
- version 9.2
- .mark_list = {}
- .mark_name = {}
- .mark_opts.reset

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}

> ----- end _optlist.reset -----
- .smark_list = {}
- .smark_name = {}
- .smark_opts.reset

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}

> ----- end _optlist.reset -----
- .rmcoll_list = {}
- .rmcoll_name = {}
- .rmcoll_opts.reset

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}

> ----- end _optlist.reset -----
- .rmdcoll_list = {}
- .rmdcoll_name = {}
- .rmdcoll_opts.reset

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}

> ----- end _optlist.reset -----

```

```

- syntax [, EQopts(namelist) EQARGopts(namelist) noNEEDVARLIST C
> OMMONopts(string asis) MARKopts(namelist) SMARKopts(namelist) RMCOLLOpts(nam
> elist) RMDCOLLOpts(namelist) NUMDEPvars(numlist integer >=0) NEEDEQUAL WTYPE
> S(string) IGNORENOCONS ]
- .eq_opts_allowed.reset

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}

> ----- end _optlist.reset -----
- foreach opt of local eqopts {
- .eq_opts_allowed.addopt `opt'
= .eq_opts_allowed.addopt NOCONSTant

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("NOCONSTant")
- if "`passthru'" != "" {
= if "" != "" {
  local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "NOCONSTant"
- .names.Arrpush "`mname'"
= .names.Arrpush "noconstant"

> ----- end _optlist.addopt -----
- }
- .eq_opts_allowed.addopt `opt'
= .eq_opts_allowed.addopt FREEParm

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("FREEParm")
- if "`passthru'" != "" {
= if "" != "" {
  local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "FREEParm"

```

```

- .names.Arrpush "`mname'"
= .names.Arrpush "freeparm"


---


> 

---

 end _optlist.addopt 

---


- }
- foreach opt of local eqargopts {
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt OFFset, passthru


---


> 

---

 begin _optlist.addopt 

---


- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OFFset")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec OFFset(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "OFFset(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "offset"


---


> 

---

 end _optlist.addopt 

---


- }
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt EXPosure, passthru


---


> 

---

 begin _optlist.addopt 

---


- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("EXPosure")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec EXPosure(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "EXPosure(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "exposure"


---


> 

---

 end _optlist.addopt 

---


- }

```

```

- local i 0
- foreach opt of local markopts {
  .mark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local smarkopts {
  .smark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmcollopts {
  .rmcoll_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmdcollopts {
  .rmdcoll_opts.addopt `opt', passthru
}
- .n_depvars.Arrdropall
- if "`numdepvars'" != "" {
= if "0" != "" {
- foreach ndep of local numdepvars {
- .n_depvars.Arrpush `ndep'
= .n_depvars.Arrpush 0
- }
- }
- else .n_depvars.Arrpush 1
- .needequal = "`needequal'"
= .needequal = "needequal"
- .wtypes = "`wtypes'"
= .wtypes = ""
- .noneedvarlist = "`needvarlist'"
= .noneedvarlist = "noneedvarlist"
- .nocons_ignore = "`ignorenocons'"
= .nocons_ignore = ""

> _____ end _eqlist._setup _____
> _____ end _eqlist.new _____
- .`eqlist'.parse `spec'
= .__000006.parse (y_timing_max: y_timing_max=, ) ("lnalpha":, fre
> eparm)

> _____ begin _eqlist.parse _____
- version 9.2
- .__init

> _____ begin _eqlist._init _____

```



```

- version 9.2
- .n_eq = 0
- .gl_if = ""
- .gl_in = ""
- .gl_wgt = ""
- .gl_options = ""
- .eq_list = {}

> ----- end _eqlist._init -----
- local MARKOPTLIST ```.mark_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```.optlist'''
= sreturn local optlist `'''
- class exit ```.optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----
= local MARKOPTLIST `'''
- local SMARKOPTLIST ```.smark_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```.optlist'''
= sreturn local optlist `'''
- class exit ```.optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----
= local SMARKOPTLIST `'''
- local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {

```

```

= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist `'''
- class exit ```optlist'''
= class exit `'''

> _____ end _optlist.dumpoptions _____

= local RMCOLLOPTLIST `'''
- local RMDCOLLOPTLIST ```.rmdcoll_opts.dumpoptions'''

> _____ begin _optlist.dumpoptions _____

- version 9
- forval i = 1/`.options.arrnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist `'''
- class exit ```optlist'''
= class exit `'''

> _____ end _optlist.dumpoptions _____

= local RMDCOLLOPTLIST `'''
- _parse expand EQ GL : 0, gweight common( `MARKOPTLIST' `SMARKOPT
> LIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
= _parse expand EQ GL : 0, gweight common( COLLlinear )
- if `EQ_n' == 0 & ``.noneedvarlist' == "" {
= if 2 == 0 & "noneedvarlist" == "" {
  syntax varlist
}
- local rebind = `EQ_n' == 1
= local rebind = 2 == 1
- if `rebind' {
= if 0 {
  tokenize ```0''', parse(",()[]")
  local rebind = ```1''' != "(" & ```2''' == "("
}
- if `rebind' {
= if 0 {
  while `:length local 0' {
    gettoken tok 0 : 0, parse(" ,") bind match(par)
    if "`par'" == "(" {

```

```

gettoken nxt : 0
local nxt = substr("`nxt'",1,1)
if inlist("`nxt'", "#", "|") {
  local ZERO ``ZERO' (`tok')''
}
else {
  local ZERO ``(`ZERO') (`tok') `0'''
  local 0
}
else if inlist("`tok'", "if", "in", ",",) {
  local ZERO ``(`ZERO') `tok' `0'''
  local 0
}
else if inlist(substr("`tok'",1,1), "#", "|") {
  local ZERO ``ZERO`tok'''
}
else {
  local ZERO ``ZERO' `tok'''
}
}
_parse expand EQ GL : ZERO, gweight common( `MARKOPTLIST' `SMARK
> OPTLIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
}
- forval i = 1/`EQ_n' {
= forval i = 1/2 {
- if `i' <= `.n_depvars.arnels' {
= if 1 <= 1 {
- local ndep = `.n_depvars[`i']'
= local ndep = 0
- }
- .eq_list[`i'] = .eqspec.new, numdepvars(`ndep') `.needequal' `.
> noneedvarlist' wtypes(`.wtypes') `.nocons_ignore'
= .eq_list[1] = .eqspec.new, numdepvars(0) needequal noneedvarlis
> t wtypes()
-class {
-ininstance:
- double n_depvars
- double needequal
- double needvarlist
- string wtypes
- string name
- string depvars
- double hasequal
- string indepvars
- string other_markvars

```

```

-      double  nocons
-      double  nocons_ignore
-      string  offset
-      double  exposure
-      double  freeparm
-      string  if
-      string  in
-      string  wgt
-      string  options
-}

> ----- begin _eqspec.new -----
- version 9.2
- syntax [, NUMDEPvars(numlist integer >=0 max=1) NEEDEQUAL noNE
> EDVARLIST WTYPES(string) IGNORENOCONS ]
- .n_depvars = 1
- if `:list sizeof numdepvars' {
= if 1 {
- .n_depvars = `numdepvars'
= .n_depvars = 0
- }
- .needequal = ("`needequal'" != "")
= .needequal = ("needequal" != "")
- .needvarlist = ("`needvarlist'" == "")
= .needvarlist = ("noneedvarlist" == "")
- .nocons_ignore = ("`ignorenocons'" != "")
= .nocons_ignore = ("" != "")
- .wtypes = "`wtypes'"
= .wtypes = ""

> ----- end _eqspec.new -----
- .eq_list[`i'].parse `.eq_opts_allowed.objkey':`EQ`i''
= .eq_list[1].parse __key(60000be5c1c0):y_timing_max: y_timing_max
> =

> ----- begin _eqspec.parse -----
- version 9.2
- _on_colon_parse `0'
= _on_colon_parse __key(60000be5c1c0):y_timing_max: y_timing_max
> =

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes

```

```

- if "`par'" != "" {
= if "" != "" {
    local before `("`before')"'
}
- if ``"before"' == ":" {
= if `__key(60000be5c1c0)' == ":" {
    sreturn local after ``"after"'
    exit
}
- while ``COLON"' != ":" & ``"after"' != "" {
= while `"' != ":" & `":y_timing_max: y_timing_max=" != "" {
- gettoken COLON after : after, parse(":") bind match(par) quo

> tes

- if "`par'" != "" {
= if "" != "" {
    local before `before' (`COLON')
    local COLON
}
- else if ``COLON"' != ":" {
= else if `":"' != ":" {
    local before ``"before' `COLON"'
    local COLON
}
- }
- while ``COLON"' != ":" & ``"after"' != "" {
= while `":"' != ":" & `":y_timing_max: y_timing_max=" != "" {
    gettoken COLON after : after, parse(":") bind match(par) quo

> tes

    if "`par'" != "" {
        local before `before' (`COLON')
        local COLON
    }
    else if ``COLON"' != ":" {
        local before ``"before' `COLON"'
        local COLON
    }
}
- if ``COLON"' != ":" {
= if `":"' != ":" {
    di as err "" found where ':' expected"
    exit 198
}
- sreturn local before ``"before"'
= sreturn local before `__key(60000be5c1c0)'
- sreturn local after ``"after"'
= sreturn local after `":y_timing_max: y_timing_max="

```

```

> ----- end _on_colon_parse -----
- local ol `s(before)'
= local ol __key(60000be5c1c0)
- if `:list sizeof ol' {
= if 1 {
- local OPTLIST ```.ol'.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arrnels' {
= forval i = 1/4 {
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParam
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParam OFFSET(passthru)
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParam OFFSET(passthru) EXPosure(
> passthru)
- }
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist ``NOCONSTant FREEParam OFFSET(passthru)
> EXPosure(passthru)'''
- class exit ```optlist'''
= class exit ``NOCONSTant FREEParam OFFSET(passthru) EXPosure(p
> assthru)'''

> ----- end _optlist.dumpoptions -----
= local OPTLIST ``NOCONSTant FREEParam OFFSET(passthru) EXPosure(
> passthru)'''
- local OPTNAMES ```.ol'.dumpnames'''

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arrnels' {
= forval i = 1/4 {
- local namelist `namelist' `.names['i']'
= local namelist noconstant
- }
- local namelist `namelist' `.names['i']'

```

```

= local namelist noconstant freeparm
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset exposure
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `noconstant freeparm offset exposure

> ""

- class exit ``namelist'''
= class exit `noconstant freeparm offset exposure'''

> _____ end _optlist.dumpnames _____

= local OPTNAMES `noconstant freeparm offset exposure'''
- local NOCONS constant noconstant
- if ``:list NOCONS & OPTNAMES''' == "" {
= if `noconstant''' == "" {
    local NOCONS noCONSTant
}
- else local NOCONS
- }
- local 0 ``s(after)'''
= local 0 `y_timing_max: y_timing_max='
- if ``.wtypes''' != "" {
= if `''' != "" {
    local wgt `[.wtypes']"
}
- syntax [anything(equalok)] [if] [in] `wgt' [, `OPTLIST' `NOCON

> S' *]

= syntax [anything(equalok)] [if] [in] [, NOCONSTant FREEParm 0
> FFset(passthru) EXPosure(passthru) *]
- local HOLDIF ``if'''
= local HOLDIF `'''
- local HOLDIN ``in'''
= local HOLDIN `'''
- if `wgt''' != "" & `weight''' != "" {
= if "" != "" & "" != "" {
    local HOLDWGT ``[weight`exp]'''
}
- gettoken NAME REST : anything, parse(":")
- if trim("`NAME'") == ":" {
= if trim("y_timing_max") == ":" {
    di as err "nothing found where equation name expected"

```

```

        exit 198
    }
- if `:list sizeof REST' {
= if 2 {
- if strpos(`"NAME"', ":") {
= if strpos(`"y_timing_max"', ":") {
    capture _msparse `NAME'
    if c(rc) {
        di as err "{bf:`NAME'} invalid equation specification"
        exit 198
    }
}
- else {
- . _confirm_eqname `NAME'
= . _confirm_eqname y_timing_max
}

> ----- begin _eqspec._confirm_eqname -----
- version 9.2
- capture confirm number `0'
= capture confirm number y_timing_max
- if (! c(rc)) exit
- local hasdot = strpos(`"0"', ".")
= local hasdot = strpos(`"y_timing_max"', ".")
- if `hasdot' {
= if 0 {
    syntax varname(ts)
    exit
}
- capture syntax name(id="equation name")
- if (! c(rc)) exit
}

> ----- end _eqspec._confirm_eqname -----
- }
- local NAME : list retok NAME
- gettoken COLON anything : REST, parse(":")
- }
- else {
    local NAME
}
- gettoken DEPVARS REST : anything, parse("=") bind
- if "`DEPVARS'" == "=" {
= if "y_timing_max" == "=" {
    local REST : copy local anything
    local DEPVARS
}
- if (`:length local DEPVARS' == 0 & `.n_depvars') {

```



```

= if (12 == 0 & 0) {
  di as err "nothing found where depvar expected"
  exit 198
}
- if ``"REST" == "" {
= if `"=" == "" {
  if 0`.needequal' {
    if `.needvarlist' | (`:length local DEPVARs' != 0 & `.n_depvar
> s') {
    di as err "equal sign required to separate depvars from indepv
> ars"

    exit 198
  }
  }
  local DEPVARs
  local n_deps : list sizeof anything
  if `n_deps' != 0 | `.needvarlist' {
    if `n_deps' < `.n_depvars' {
      error 102
    }
    forval j = 1/`.n_depvars' {
      gettoken YY anything : anything
      local DEPVARs `DEPVARs' `YY'
    }
    local INDEPVARs : list retok anything
  }
  loca hasequal 0
}
- else {
- gettoken EQ INDEPVARs : REST, parse("=") bind
- local hasequal 1
- }
- if `:list sizeof DEPVARs' {
= if 1 {
- _fv_check_depvar `DEPVARs'
= _fv_check_depvar y_timing_max
- tsunab DEPVARs : `DEPVARs'
= tsunab DEPVARs : y_timing_max

}

> ----- begin tsunab -----
- version 6
- gettoken user 0: 0, parse(" :")
- gettoken colon 0: 0, parse(" :")
- if ``"colon" != ":" { error 198 }
= if `":" != ":" { error 198 }
- syntax [varlist(ts default=empty)] [, MIN(integer 1) MAX(int

```

```

> eger 32000) NAME(string)]
    - c_local `user' `varlist'
    = c_local DEPVARS y_timing_max
    - local n : word count `varlist'
    = local n : word count y_timing_max
    - if `n'>=`min' & `n'<=`max' { exit }
    = if 1>=1 & 1<=32000 { exit }

> ----- end tsunab -----
    - }
    - if `:list sizeof INDEPVARS' {
    = if 0 {
        fvnab INDEPVARS : `INDEPVARS'
    }
    - if inlist("noconstant", "`noconstant'", "`constant'") {
    = if inlist("noconstant", "", "") {
        if "`INDEPVARS'" == "" & !`.nocons_ignore' {
            di as err "independent variables are required when option noco
> nstant is specified"
            exit 102
        }
        if "`NOCONS'" != "" {
            local options `options' noconstant
        }
        local NOCONS 1
    }
    - else local NOCONS 0
    - if ``offset'`exposure'"" != "" {
    = if `""' != "" {
        . _check_offset, `offset' `exposure'
        local OFFSET "`s(offset)'"
        local EXPOSURE = ("`exposure'" != "")
        local OTHER `s(offset)'
    }
    - else local EXPOSURE 0
    - local FREEPARM = ``freeparm'"" != ""
    = local FREEPARM = `""' != ""
    - local OPTIONS
    - local SPACE
    - foreach OPTNAME of local OPTNAMES {
    - if ``OPTNAME'"" != "" {
    = if `""' != "" {
        local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'""
        local SPACE " "
    }
    - }

```

```

- if `""`OPTNAME'''' != "" {
= if `""' != "" {
    local OPTIONS `""OPTIONS'`SPACE'`OPTNAME''''
    local SPACE " "
  }
- }
- if `""`OPTNAME'''' != "" {
= if `""' != "" {
    local OPTIONS `""OPTIONS'`SPACE'`OPTNAME''''
    local SPACE " "
  }
- }
- if `""`OPTNAME'''' != "" {
= if `""' != "" {
    local OPTIONS `""OPTIONS'`SPACE'`OPTNAME''''
    local SPACE " "
  }
- }
- if `""options'''' != "" {
= if `""' != "" {
    local OPTIONS `""OPTIONS'`SPACE'`options''''
  }
- .name = `""NAME''''
= .name = `""y_timing_max''
- .depvars = `""DEPVARs''''
= .depvars = `""y_timing_max''
- .hasequal = `hasequal'
= .hasequal = 1
- .indepvars = `"":list retok INDEPVARs''''
= .indepvars = `""'
- .other_markvars = `"":list retok OTHER''''
= .other_markvars = `""'
- .nocons = `NOCONS'
= .nocons = 0
- .if = `""HOLDIF''''
= .if = `""'
- .in = `""HOLDIN''''
= .in = `""'
- .wgt = `""HOLDWGT''''
= .wgt = `""'
- .offset = `""OFFSET''''
= .offset = `""'
- .exposure = `EXPOSURE'
= .exposure = 0
- .freeparm = `FREEPARM'
= .freeparm = 0

```

```

- .options = ```OPTIONS``
= .options = ````

```

---

```

> ----- end _eqspec.parse -----
- }
- if `i' <= `.n_depvars.arnels' {
= if 2 <= 1 {
    local ndep = `.n_depvars[`i']'
    }
- .eq_list[`i'] = ._eqspec.new, numdepvars(`ndep') `.needequal' `.
> noneedvarlist' wtypes(`.wtypes') `.nocons_ignore'
    = .eq_list[2] = ._eqspec.new, numdepvars(0) needequal noneedvarlis
> t wtypes()

```

---

```

> ----- begin _eqspec.new -----
- version 9.2
- syntax [, NUMDEPvars(numlist integer >=0 max=1) NEEDEQUAL noNE
> EDVARLIST WTYPES(string) IGNORENOCONS ]
- .n_depvars = 1
- if `:list sizeof numdepvars' {
= if 1 {
- .n_depvars = `numdepvars'
= .n_depvars = 0
- }
- .needequal = ("`needequal'" != "")
= .needequal = ("needequal" != "")
- .needvarlist = ("`needvarlist'" == "")
= .needvarlist = ("noneedvarlist" == "")
- .nocons_ignore = ("`ignorenocons'" != "")
= .nocons_ignore = ("" != "")
- .wtypes = "`wtypes'"
= .wtypes = ""

```

---

```

> ----- end _eqspec.new -----
- .eq_list[`i'].parse `.eq_opts_allowed.objkey':`EQ_`i''
= .eq_list[2].parse __key(60000be5c1c0):"lnalpha":, freeparm

```

---

```

> ----- begin _eqspec.parse -----
- version 9.2
- _on_colon_parse `0'
= _on_colon_parse __key(60000be5c1c0):"lnalpha":, freeparm

```

---

```

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""

```

```

- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `(`before')'
}
- if ``before' == ":" {
= if `__key(60000be5c1c0)' == ":" {
  sreturn local after ``after'
  exit
}
- while ``COLON' != ":" & ``after' != "" {
= while `"' != ":" & `":'lnalpha":, freeparm' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quo

```

> tes

```

- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``COLON' != ":" {
= else if `":' != ":" {
  local before ``before' `COLON'
  local COLON
}
- }
- while ``COLON' != ":" & ``after' != "" {
= while `":' != ":" & `":'lnalpha":, freeparm' != "" {
  gettoken COLON after : after, parse(":") bind match(par) quo

```

> tes

```

if "`par'" != "" {
  local before `before' (`COLON')
  local COLON
}
else if ``COLON' != ":" {
  local before ``before' `COLON'
  local COLON
}
}
- if ``COLON' != ":" {
= if `":' != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before ``before'
= sreturn local before `__key(60000be5c1c0)'
- sreturn local after ``after'

```

```

= sreturn local after `""lnalpha":, freeparm""


---


> 

---

 end _on_colon_parse 

---


- local ol `s(before)'
= local ol __key(60000be5c1c0)
- if `:list sizeof ol' {
= if 1 {
- local OPTLIST `""`.ol'.dumpoptions'''


---


> 

---

 begin _optlist.dumpoptions 

---


- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/4 {
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm OFFSET(passthru)
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm OFFSET(passthru) EXPosure(
> passthru)
- }
- local optlist : list uniq optlist
- sreturn local optlist `""`optlist'''
= sreturn local optlist `""NOCONSTant FREEParm OFFSET(passthru)
> EXPosure(passthru)'''
- class exit `""`optlist'''
= class exit `""NOCONSTant FREEParm OFFSET(passthru) EXPosure(p
> assthru)'''


---


> 

---

 end _optlist.dumpoptions 

---


= local OPTLIST `""NOCONSTant FREEParm OFFSET(passthru) EXPosure(
> passthru)'''
- local OPTNAMES `""`.ol'.dumpnames'''


---


> 

---

 begin _optlist.dumpnames 

---


- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/4 {
- local namelist `namelist' `.names['i']'
= local namelist noconstant
- }

```

```

- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset exposure
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist ``noconstant freeparm offset exposure

> ""

- class exit ``namelist'''
= class exit ``noconstant freeparm offset exposure'''



---


> ----- end _optlist.dumpnames -----

= local OPTNAMES ``noconstant freeparm offset exposure'''
- local NOCONS constant noconstant
- if ``:list NOCONS & OPTNAMES''' == "" {
= if ``noconstant''' == "" {
  local NOCONS noCONSTant
}
- else local NOCONS
- }
- local 0 ``s(after)'''
= local 0 ``lnalpha":, freeparm'''
- if ``.wtypes''' != "" {
= if ``''' != "" {
  local wgt "[.wtypes']"
}
- syntax [anything(equalok)] [if] [in] `wgt' [, `OPTLIST' `NOCON

> S' *]

= syntax [anything(equalok)] [if] [in] [, NOCONSTant FREEParam 0
> FFset(passthru) EXPosure(passthru) *]
- local HOLDIF ``if'''
= local HOLDIF ``'''
- local HOLDIN ``in'''
= local HOLDIN ``'''
- if ``wgt''' != "" & ``weight''' != "" {
= if "" != "" & "" != "" {
  local HOLDWGT ``[weight'exp]'''
}
- gettoken NAME REST : anything, parse(":")
- if trim("`NAME'") == ":" {
= if trim("lnalpha") == ":" {

```

```

        di as err "nothing found where equation name expected"
        exit 198
    }
- if `:list sizeof REST' {
= if 1 {
- if strpos(`"NAME"', ":") {
= if strpos(`"lnalpha"', ":") {
    capture _msparse `NAME'
    if c(rc) {
        di as err "{bf:`NAME'} invalid equation specification"
        exit 198
    }
}
- else {
- ._confirm_eqname `NAME'
= ._confirm_eqname lnalpha
}

> ----- begin _eqspec._confirm_eqname -----
- version 9.2
- capture confirm number `0'
= capture confirm number lnalpha
- if (! c(rc)) exit
- local hasdot = strpos(`"0"', ".")
= local hasdot = strpos(`"lnalpha"', ".")
- if `hasdot' {
= if 0 {
    syntax varname(ts)
    exit
}
- capture syntax name(id="equation name")
- if (! c(rc)) exit
}

> ----- end _eqspec._confirm_eqname -----
- }
- local NAME : list retok NAME
- gettoken COLON anything : REST, parse(":")
- }
- else {
    local NAME
}
- gettoken DEPVARS REST : anything, parse("=") bind
- if "`DEPVARS'" == "=" {
= if "" == "=" {
    local REST : copy local anything
    local DEPVARS
}
}

```



```

- if (`:length local DEPVARs' == 0 & `.n_depvars') {
= if (0 == 0 & 0) {
    di as err "nothing found where depvar expected"
    exit 198
  }
- if ``REST'' == "" {
= if `'' == "" {
- if 0`.needequal' {
= if 01 {
- if `.needvarlist' | (`:length local DEPVARs' != 0 & `.n_depvar
> s') {
= if 0 | (0 != 0 & 0) {
    di as err "equal sign required to separate depvars from indepv
> ars"
    exit 198
  }
- }
- local DEPVARs
- local n_deps : list sizeof anything
- if `n_deps' != 0 | `.needvarlist' {
= if 0 != 0 | 0 {
    if `n_deps' < `.n_depvars' {
      error 102
    }
    forval j = 1/`.n_depvars' {
      gettoken YY anything : anything
      local DEPVARs `DEPVARs' `YY'
    }
    local INDEPVARs : list retok anything
  }
- local hasequal 0
- }
- else {
  gettoken EQ INDEPVARs : REST, parse("=") bind
  local hasequal 1
}
- if `:list sizeof DEPVARs' {
= if 0 {
  _fv_check_depvar `DEPVARs'
  tsunab DEPVARs : `DEPVARs'
}
- if `:list sizeof INDEPVARs' {
= if 0 {
  fvunab INDEPVARs : `INDEPVARs'
}
- if inlist("noconstant", "`noconstant'", "`constant'") {

```

```

= if inlist("noconstant", "", "") {
  if "`INDEPVARs'" == "" & !`.nocons_ignore' {
    di as err "independent variables are required when option noco
> nstant is specified"
    exit 102
  }
  if "`NOCONS'" != "" {
    local options `options' noconstant
  }
  local NOCONS 1
}
- else local NOCONS 0
- if ``offset'`exposure'`` != "" {
= if `""' != "" {
  . _check_offset, `offset' `exposure'
  local OFFSET ``s(offset)''
  local EXPOSURE = (`exposure' != "")
  local OTHER `s(offset)'
}
- else local EXPOSURE 0
- local FREEARM = ``freeparm'`` != ""
= local FREEARM = `freeparm' != ""
- local OPTIONS
- local SPACE
- foreach OPTNAME of local OPTNAMES {
- if ``OPTNAME'`` != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
  local SPACE " "
}
- }
- if ``OPTNAME'`` != "" {
= if `freeparm' != "" {
- local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
= local OPTIONS `freeparm'
- local SPACE " "
- }
- }
- if ``OPTNAME'`` != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
  local SPACE " "
}
- }
- if ``OPTNAME'`` != "" {
= if `""' != "" {

```

```

        local OPTIONS ```OPTIONS``SPACE``OPTNAME````
        local SPACE " "
    }
- }
- if ```options```` != "" {
= if ```` != "" {
    local OPTIONS ```OPTIONS``SPACE``options````
    }
- .name = ```NAME````
= .name = ``lnalpha``
- .depvars = ```DEPVAR````
= .depvars = ````
- .hasequal = `hasequal`
= .hasequal = 0
- .indepvars = ```:list retok INDEPVAR````
= .indepvars = ````
- .other_markvars = ```:list retok OTHER````
= .other_markvars = ````
- .nocons = `NOCONS`
= .nocons = 0
- .if = ```HOLDIF````
= .if = ````
- .in = ```HOLDIN````
= .in = ````
- .wgt = ```HOLDWGT````
= .wgt = ````
- .offset = ```OFFSET````
= .offset = ````
- .exposure = `EXPOSURE`
= .exposure = 0
- .freeparm = `FREEPARM`
= .freeparm = 1
- .options = ```OPTIONS````
= .options = ``freeparm``

> _____ end _eqspec.parse _____
- }
- local MARKOPLIST ```.mark_opts.dumpoptions````

> _____ begin _optlist.dumpoptions _____
- version 9
- forval i = 1/`.options.arnels` {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']`
    }
- local optlist : list uniq optlist

```

```

- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----

= local MARKOPTLIST `'''
- local MARKOPTNAMES ```.mark_opts.dumpnames'''

> ----- begin _optlist.dumpnames -----

- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/0 {
    local namelist `namelist' `'.names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

> ----- end _optlist.dumpnames -----

= local MARKOPTNAMES `'''
- local OPTLIST1 : substr local MARKOPTLIST "(passthru)" "(string
> g asis)", all
- local SMARKOPTLIST ```.smark_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----

- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `'.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----

= local SMARKOPTLIST `'''
- local SMARKOPTNAMES ```.smark_opts.dumpnames'''

> ----- begin _optlist.dumpnames -----

- version 9
- forval i = 1/`.names.arnels' {

```

```

= forval i = 1/0 {
  local namelist `namelist' `.names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

> ----- end _optlist.dumpnames -----

= local SMARKOPTNAMES `'''
- local OPTLIST1s : substr local SMARKOPTLIST "(passthru)" "(str
> ing asis)", all
- local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----

- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----

= local RMCOLLOPTLIST `'''
- local RMCOLLOPTNAMES ```.rmcoll_opts.dumpnames'''

> ----- begin _optlist.dumpnames -----

- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/0 {
  local namelist `namelist' `.names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

> ----- end _optlist.dumpnames -----

= local RMCOLLOPTNAMES `'''
- local OPTLIST2 : substr local RMCOLLOPTLIST "(passthru)" "(str

```

```

> ring asis)", all
    - local RMDCOLLOPTLIST ```.rmdcoll_opts.dumpoptions'''
> ----- begin _optlist.dumpoptions -----
    - version 9
    - forval i = 1/`.options.arnels' {
    = forval i = 1/0 {
        local optlist `optlist' `.options['i']'
    }
    - local optlist : list uniq optlist
    - sreturn local optlist ```optlist'''
    = sreturn local optlist `'''
    - class exit ```optlist'''
    = class exit `'''
> ----- end _optlist.dumpoptions -----
    = local RMDCOLLOPTLIST `'''
    - local RMDCOLLOPTNAMES ```.rmdcoll_opts.dumpnames'''
> ----- begin _optlist.dumpnames -----
    - version 9
    - forval i = 1/`.names.arnels' {
    = forval i = 1/0 {
        local namelist `namelist' `.names['i']'
    }
    - local namelist : list uniq namelist
    - sreturn local namelist ```namelist'''
    = sreturn local namelist `'''
    - class exit ```namelist'''
    = class exit `'''
> ----- end _optlist.dumpnames -----
    = local RMDCOLLOPTNAMES `'''
    - local OPTLIST3 : substr local RMDCOLLOPTLIST "(passthru)" "(st
> ring asis)", all
    - local 0 `", `GL_op'''
    = local 0 `", ""
    - syntax [, `OPTLIST1' `OPTLIST1s' `OPTLIST2' `OPTLIST3' COLLlinear
> noCONSTant * ]
    = syntax [, COLLlinear noCONSTant * ]
    - local GL_op ```options'''
    = local GL_op `'''
    - if ```collinear''' != "" {
    = if `''' != "" {
        local GL_op ```GL_op' `collinear'''
        .collinear = ``collinear'''

```

```

}
- if ``constant'' != "" {
= if ``'' != "" {
    local GL_op ``GL_op' `constant''
    .nocons = ``constant''
}
- local I 0
- foreach OPT of local MARKOPTNAMES {
    if `:list sizeof `OPT'' {
        .mark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .mark_list[`I'].parse "" : ``OPT''
        .mark_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local SMARKOPTNAMES {
    if `:list sizeof `OPT'' {
        .smark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .smark_list[`I'].parse "" : ``OPT''
        .smark_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local RMCOLLOPTNAMES {
    if `:list sizeof `OPT'' {
        .rmcoll_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .rmcoll_list[`I'].parse "" : ``OPT''
        .rmcoll_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local RMDCOLLOPTNAMES {
    if `:list sizeof `OPT'' {
        .rmdcoll_list[`++I'] = .eqspec.new, wtypes(`.wtypes')
        .rmdcoll_list[`I'].parse "" : ``OPT''
        .rmdcoll_name.Arrpush ``OPT''
    }
}
- .n_eq = `EQ_n'
= .n_eq = 2
- .gl_if = ``GL_if''
= .gl_if = ``''
- .gl_in = ``GL_in''
= .gl_in = ``''
- .gl_wgt = ``GL_wt''
= .gl_wgt = ``''

```

```

- .gl_options = ``:list retok GL_op'''
= .gl_options = `'''

> ----- end _eqlist.parse -----
- if ``subpop''' != "" & ``missing''' == "" {
= if "" != "" & "missing" == "" {
  if `interactive' {
    SetGlobalVar ML_samp : _MLtu
  }
  else {
    global ML_samp : tempvar
    if ``moptobj''' != "" {
      global ML_samp `moptobj'$ML_samp
    }
  }
  tempvar markuse
  _svy_setup $ML_sample $ML_samp, svy subpop(`subpop')
  quietly gen byte `markuse' = $ML_samp
}
- else global ML_samp $ML_sample
= else global ML_samp __000005
- if ``collinear''' == "" {
= if "collinear" == "" {
  .`eqlist'.rmcoll $ML_samp, replace `missing'
}
- else if ``missing''' == "" {
= else if "missing" == "" {
  .`eqlist'.markout $ML_samp, replace
}
- if "$ML_samp" != "$ML_sample" {
= if "__000005" != "__000005" {
  `vv' quietly replace $ML_sample = 0 if `markuse' & !$ML_samp
  drop `markuse'
}
- local eq_n = `.`eqlist'.eq count'

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item''' == "count" {
= if "count" == "count" {
  - sreturn local value ``.n_eq'''
= sreturn local value ``2'''
  - class exit ``s(value)'''
= class exit ``2'''

```



```

> ----- end _eqlist.eq -----
    = local eq_n = 2
    - global ML_n : copy local eq_n
    - global ML_dim = ``.eqlist'.dim'
    -----

> ----- begin _eqlist.dim -----
    - version 9.2
    - syntax
    - local neq = ``.eq count'
    -----

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item'" == "count" {
    = if "count" == "count" {
    - sreturn local value ``.n_eq'"
    = sreturn local value ``2'"
    - class exit ``s(value)'"
    = class exit ``2'"
    -----

> ----- end _eqlist.eq -----
    = local neq = 2
    - forval i = 1/`neq' {
    = forval i = 1/2 {
    - local xvarsi ``.eq indepvars `i'"
    -----

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item'" == "count" {
    = if "indepvars" == "count" {
    sreturn local value ``.n_eq'"
    class exit ``s(value)'"
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item indepvars
    -----

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, indepvars
    -----

> ----- begin _eqspec._check_eq_item_name -----

```

```

- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
    local other other_markvars
    }
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
    error 198
    }
- c_local `c_item' `item'
= c_local item indepvars

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'"""
= local value `""'
- sreturn local value ```.value'"""
= sreturn local value `""'
- class exit ```.value'"""
= class exit `""'

> ----- end _eqspec.item -----
- class exit ```.s(value)"""
= class exit `""'

> ----- end _eqlist.eq -----
= local xvarsi `""'
- local noconsi ```.eq nocons `i'"""

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "nocons" == "count" {
    sreturn local value ```.n_eq'"""
    class exit ```.s(value)"""
    }
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
= .eq_list[1].item nocons

```

```

> ----- begin _eqspec.item -----
- version 9.2
- args item
- ._check_eq_item_name item, `item'
= ._check_eq_item_name item, nocons

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPosure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item nocons
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item nocons

> ----- end _eqspec._check_eq_item_name -----
- local value ```.item'''
= local value ``0''
- sreturn local value ``value'''
= sreturn local value ``0''
- class exit ``value'''
= class exit ``0''

> ----- end _eqspec.item -----
- class exit ``s(value)'''
= class exit ``0''

> ----- end _eqlist.eq -----
= local noconsi ``0''
- local dim = `dim' + `:list sizeof xvarsi'
= local dim = + 0
- if ``noconsi' == "0" {
= if "0" == "0" {
- local ++dim
- }

```

```

- }
- local xvarsi ```.eq indepvars `i'```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "indepvars" == "count" {
sreturn local value ```.n_eq```
class exit ```.s(value)```
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item indepvars

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, indepvars

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPosure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item indepvars

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ````
- sreturn local value ```.value```
= sreturn local value ````

```

```

- class exit ``value'''
= class exit `'''

```

---

```

> ----- end _eqspec.item -----
- class exit ``s(value)'''
= class exit `'''

```

---

```

> ----- end _eqlist.eq -----
= local xvarsi `'''
- local noconsi ``.eq nocons `i'''

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "nocons" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item nocons

```

---

```

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, nocons

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item nocons
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}

```

```

- c_local `c_item' `item'
= c_local item nocons


---


> 

---

 end _eqspec._check_eq_item_name 

---


- local value ```.`item'"""
= local value ``0"
- sreturn local value ```value'"""
= sreturn local value ``0"
- class exit ```value'"""
= class exit ``0"


---


> 

---

 end _eqspec.item 

---


- class exit ```s(value)"""
= class exit ``0"


---


> 

---

 end _eqlist.eq 

---


= local noconsi ``0"
- local dim = `dim' + `:list sizeof xvarsi'
= local dim = 1 + 0
- if ``noconsi' == "0" {
= if "0" == "0" {
- local ++dim
- }
- }
- sreturn local value ```dim'"""
= sreturn local value ``2"
- class exit ```s(value)"""
= class exit ``2"


---


> 

---

 end _eqlist.dim 

---


= global ML_dim = 2
- forval i = 1/`eq_n' {
= forval i = 1/2 {
- local depvars `.`eqlist'.eq depvars `i'


---


> 

---

 begin _eqlist.eq 

---


- version 9.2
- args item idx
- if ``item' == "count" {
= if "depvars" == "count" {
sreturn local value ```.n_eq'"""
class exit ```s(value)"""
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'

```

```

= .eq_list[1].item depvars
> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, depvars
> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item depvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item depvars
> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value `y_timing_max`
- sreturn local value ```.`value'```
= sreturn local value `y_timing_max`
- class exit ```.`value'```
= class exit `y_timing_max`
> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit `y_timing_max`
> ----- end _eqlist.eq -----
= local depvars y_timing_max
- local eq `.`eqlist'.eq name `i'
> ----- begin _eqlist.eq -----
- version 9.2
- args item idx

```

```

- if "`item'" == "count" {
= if "name" == "count" {
  sreturn local value ```.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item name
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, name
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item name
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item name
}

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item''''
= local value `y_timing_max'
- sreturn local value ``value'''
= sreturn local value `y_timing_max'
- class exit ``value'''
= class exit `y_timing_max'
}

> ----- end _eqspec.item -----
- class exit ``s(value)'''
= class exit `y_timing_max'

```



```

> ----- end _eqlist.eq -----
= local eq y_timing_max
- if !':length local eq' {
= if !12 {
  local eq eq`i'
}
- if `:list eq in eqnames' {
= if 0 {
  di as err "equation ``eq'' multiply defined"
  exit 110
}
- local eqnames `eqnames' `eq'
= local eqnames y_timing_max
- fvexpand `.`_eqlist'.eq indepvars `i'' if $ML_samp

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item'" == "count" {
= if "indepvars" == "count" {
  sreturn local value ```.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item indepvars

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, indepvars

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'

```

```

= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item indepvars


---


> 

---

 end _eqspec._check_eq_item_name 

---


- local value ```.`item'``
= local value `'''
- sreturn local value ```.value'''
= sreturn local value `'''
- class exit ```.value'''
= class exit `'''


---


> 

---

 end _eqspec.item 

---


- class exit ```.s(value)'''
= class exit `'''


---


> 

---

 end _eqlist.eq 

---


= fvexpand if __000005
- if "r(tsops)" == "true" {
= if "" == "true" {
global ML_preserve no
}
- local eq_rhs`i' ``r(varlist)'''
= local eq_rhs1 `'''
- local eq_nocons`i' ``.eqlist'.eq nocons `i''


---


> 

---

 begin _eqlist.eq 

---


- version 9.2
- args item idx
- if "item" == "count" {
= if "nocons" == "count" {
sreturn local value ```.n_eq'''
class exit ```.s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item nocons


---


> 

---

 begin _eqspec.item 

---


- version 9.2
- args item

```

```

- .check_eq_item_name item, `item'
= .check_eq_item_name item, nocons

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
    local other other_markvars
    }
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item nocons
- if `:list sizeof item' != 1 {
= if 1 != 1 {
    error 198
    }
- c_local `c_item' `item'
= c_local item nocons

```

---

```

> ----- end _eqspec._check_eq_item_name -----
- local value ```.item'''
= local value ``0'''
- sreturn local value ``value'''
= sreturn local value ``0'''
- class exit ``value'''
= class exit ``0'''

```

---

```

> ----- end _eqspec.item -----
- class exit ``s(value)'''
= class exit ``0'''

```

---

```

> ----- end _eqlist.eq -----
= local eq_nocons1 0
- local eq_offset`i' `.`eqlist'.eq offset `i'

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "offset" == "count" {
    sreturn local value ``.n_eq'''
    class exit ``s(value)'''
    }

```

```

- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item offset

```

---

```

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, offset

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item offset
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item offset

```

---

```

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ````
- sreturn local value ```.value```
= sreturn local value ````
- class exit ```.value```
= class exit ````

```

---

```

> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit ````

```

---

```

> ----- end _eqlist.eq -----
= local eq_offset1
- local eq_exposure`i' `.`eqlist'.eq exposure `i'

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item'' == "count" {
= if "exposure" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item exposure
-----

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, exposure
-----

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item exposure
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item exposure
-----

> ----- end _eqspec._check_eq_item_name -----
- local value ``.item'''
= local value ``0'''
- sreturn local value ``value'''
= sreturn local value ``0'''
- class exit ``value'''
= class exit ``0'''
-----

```

```

> ----- end _eqspec.item -----
    - class exit ``s(value)''
    = class exit ``0''

> ----- end _eqlist.eq -----
    = local eq_exposure1 0
    - local eq_freeparm`i' `.`eqlist'.eq freeparm `i''

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item'' == "count" {
    = if "freeparm" == "count" {
        sreturn local value ``.n_eq''
        class exit ``s(value)''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
        - .eq_list[`idx'].item `item'
    = .eq_list[1].item freeparm

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - ._check_eq_item_name item, `item'
    = ._check_eq_item_name item, freeparm

> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
        local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
    = local item freeparm
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
        error 198
    }
    - c_local `c_item' `item'
    = c_local item freeparm

```

```

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.item'''
    = local value ``0''
    - sreturn local value ``value'''
    = sreturn local value ``0''
    - class exit ``value'''
    = class exit ``0''

> ----- end _eqspec.item -----
    - class exit ``s(value)'''
    = class exit ``0''

> ----- end _eqlist.eq -----
    = local eq_freeparm1 0
    - local lhs `lhs' `depvars'
    = local lhs y_timing_max
    - local eq_names `eq_names' `eq'
    = local eq_names y_timing_max
    - local nxvars :list sizeof eq_rhs`i'
    = local nxvars :list sizeof eq_rhs1
    - local TMP : copy local eq_rhs`i'
    = local TMP : copy local eq_rhs1
    - forval j = 1/`nxvars' {
    = forval j = 1/0 {
        gettoken xvar TMP : TMP
        local colna ``colna' `eq':`xvar'''
    }
    - if `eq_freeparm`i' {
    = if 0 {
        if `nxvars' {
            di as err "{it:xvars} not allowed with option {bf:freeparm}"
            exit 198
        }
        if strpos(``eq'', ":") {
            local colna ``colna' /\`eq'''
        }
        else {
            local colna ``colna' /\`eq'''
        }
    }
    - else if !`eq_nocons`i' {
    = else if !0 {
        - local colna ``colna' `eq':_cons'''
    = local colna `` y_timing_max:_cons'''
    - }
    - if ``moptobj' != "" {

```

```

= if "" != "" {
  if "`eq_offset`i'" != "" {
    if usubstr("`eq_offset`i'",1,2)=="__" {
      capture clonevar `moptobj`eq_offset`i' = `eq_offset`i'
      local dropmetoo `dropmetoo' `moptobj`eq_offset`i'
    }
  }
}
- }
- local depvars `.`eqlist'.eq depvars `i'

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "depvars" == "count" {
  sreturn local value ```.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item depvars

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, depvars

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item depvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}

```



```

- c_local `c_item' `item'
= c_local item depvars

```

---

```

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ````
- sreturn local value ```.value```
= sreturn local value ````
- class exit ```.value```
= class exit ````

```

---

```

> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit ````

```

---

```

> ----- end _eqlist.eq -----
= local depvars
- local eq ``.eqlist'.eq name `i'

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "name" == "count" {
  sreturn local value ```.n_eq```
  class exit ```.s(value)```
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
  - .eq_list[`idx'].item `item'
= .eq_list[2].item name

```

---

```

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, name

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars

```

```

    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
    = local item name
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
        error 198
    }
    - c_local `c_item' `item'
    = c_local item name


---


> 

---

 end _eqspec._check_eq_item_name 

---


    - local value ```.`item'```
    = local value `"\lnalpha"
    - sreturn local value ```.`value'```
    = sreturn local value `"\lnalpha"
    - class exit ```.`value'```
    = class exit `"\lnalpha"


---


> 

---

 end _eqspec.item 

---


    - class exit ```.s(value)```
    = class exit `"\lnalpha"


---


> 

---

 end _eqlist.eq 

---


    = local eq lnalpha
    - if !`:length local eq' {
    = if !7 {
        local eq eq`i'
    }
    - if `:list eq in eqnames' {
    = if 0 {
        di as err "equation ``eq'' multiply defined"
        exit 110
    }
    - local eqnames `eqnames' `eq'
    = local eqnames y_timing_max lnalpha
    - fvexpand `.`eqlist'.eq indepvars `i' if $ML_samp


---


> 

---

 begin _eqlist.eq 

---


    - version 9.2
    - args item idx
    - if ``item' == "count" {
    = if "indepvars" == "count" {
        sreturn local value ```.n_eq```
        class exit ```.s(value)```
    }

```

```

- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item indepvars
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, indepvars
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item indepvars
}

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ````
- sreturn local value ```.value```
= sreturn local value ````
- class exit ```.value```
= class exit ````

> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit ````

> ----- end _eqlist.eq -----
= fvexpand if __000005
- if "`r(tsops)'" == "true" {
= if "" == "true" {

```

```

    global ML_preserve no
    }
    - local eq_rhs`i' ``r(varlist)'''
    = local eq_rhs2 `''''
    - local eq_nocons`i' ``.eqlist'.eq nocons `i''
  }
> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item' == "count" {
    = if "nocons" == "count" {
      sreturn local value ``.n_eq'''
      class exit ``s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 2 & 2 <= 2 {
      - .eq_list[`idx'].item `item'
      = .eq_list[2].item nocons
    }
  }
> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, nocons
  }
> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
      local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
    = local item nocons
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
      error 198
    }
    - c_local `c_item' `item'
    = c_local item nocons
  }
> ----- end _eqspec._check_eq_item_name -----
    - local value ``.item'''
  }

```

```

    = local value ``0''
    - sreturn local value ``value'''
    = sreturn local value ``0''
    - class exit ``value'''
    = class exit ``0''
  _____
> _____ end _eqspec.item _____
    - class exit ``s(value)'''
    = class exit ``0''
  _____

> _____ end _eqlist.eq _____
    = local eq_nocons2 0
    - local eq_offset`i' `.`eqlist'.eq offset `i''
  _____

> _____ begin _eqlist.eq _____
    - version 9.2
    - args item idx
    - if ``item' == "count" {
    = if "offset" == "count" {
      sreturn local value ``.n_eq'''
      class exit ``s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 2 & 2 <= 2 {
      - .eq_list[`idx'].item `item'
    = .eq_list[2].item offset
  _____

> _____ begin _eqspec.item _____
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, offset
  _____

> _____ begin _eqspec._check_eq_item_name _____
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
      local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
    = local item      offset
    - if `:list sizeof item' != 1 {

```

```

        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item offset
    > ----- end _eqspec._check_eq_item_name -----
        - local value ```.`item'"""
        = local value `""'
        - sreturn local value ```.value'''
        = sreturn local value `""'
        - class exit ```.value'''
        = class exit `""'

    > ----- end _eqspec.item -----
        - class exit ```.s(value)'''
        = class exit `""'

    > ----- end _eqlist.eq -----
        = local eq_offset2
        - local eq_exposure`i' ``.eqlist'.eq exposure `i'

    > ----- begin _eqlist.eq -----
        - version 9.2
        - args item idx
        - if ``item' == "count" {
        = if "exposure" == "count" {
            sreturn local value ```.n_eq'''
            class exit ```.s(value)'''
        }
        - else if 0 < `idx' & `idx' <= `.n_eq' {
        = else if 0 < 2 & 2 <= 2 {
            - .eq_list[`idx'].item `item'
            = .eq_list[2].item exposure

    > ----- begin _eqspec.item -----
        - version 9.2
        - args item
        - .check_eq_item_name item, `item'
        = .check_eq_item_name item, exposure

    > ----- begin _eqspec._check_eq_item_name -----
        - version 9.2
        - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
    > F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
        - if c(rc) error 198

```

```

- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item      exposure
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item exposure

```

---

```

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ``0```
- sreturn local value ```.value```
= sreturn local value ``0```
- class exit ```.value```
= class exit ``0```

```

---

```

> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit ``0```

```

---

```

> ----- end _eqlist.eq -----
= local eq_exposure2 0
- local eq_freeparm`i' ``.eqlist'.eq freeparm `i'

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "freeparm" == "count" {
  sreturn local value ```.n_eq```
  class exit ```.s(value)```
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item freeparm

```

---

```

> ----- begin _eqspec.item -----
- version 9.2
- args item

```

```

- ._check_eq_item_name item, `item'
= ._check_eq_item_name item, freeparm

```

---

```

> _____ begin _eqspec._check_eq_item_name _____
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item freeparm
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item freeparm

```

---

```

> _____ end _eqspec._check_eq_item_name _____
- local value ```.`item'```
= local value `1'
- sreturn local value ```.`value'```
= sreturn local value `1'
- class exit ```.`value'```
= class exit `1'

```

---

```

> _____ end _eqspec.item _____
- class exit ```.s(value)```
= class exit `1'

```

---

```

> _____ end _eqlist.eq _____
= local eq_freeparm2 1
- local lhs `lhs' `depvars'
= local lhs y_timing_max
- local eq_names `eq_names' `eq'
= local eq_names y_timing_max lnalpha
- local nxvars :list sizeof eq_rhs`i'
= local nxvars :list sizeof eq_rhs2
- local TMP : copy local eq_rhs`i'
= local TMP : copy local eq_rhs2
- forval j = 1/`nxvars' {
= forval j = 1/0 {

```



```

gettoken xvar TMP : TMP
local colna ``colna' `eq':`xvar'``
}
- if `eq_freeparm`i'' {
= if 1 {
- if `nxvars' {
= if 0 {
di as err "{it:xvars} not allowed with option {bf:freeparm}"
exit 198
}
- if strpos(```eq'``, ":") {
= if strpos(``lnalpha'`, ":") {
local colna ``colna' `/eq'``
}
- else {
- local colna ``colna' /:`eq'``
= local colna `` y_timing_max:_cons /:lnalpha``
- }
- }
- else if !`eq_nocons`i'' {
= else if !0 {
local colna ``colna' `eq':_cons``
}
- if ``moptobj'' != "" {
= if "" != "" {
if ``eq_offset`i'' != "" {
if usubstr(``eq_offset`i'',1,2)=="__" {
capture clonevar `moptobj'`eq_offset`i' = `eq_offset`i'
local dropmetoo `dropmetoo' `moptobj'`eq_offset`i'
}
}
}
- }
- if ``moptobj'' != "" {
= if "" != "" {
foreach y of local lhs {
if usubstr(``y'',1,2)=="__" {
capture clonevar `moptobj'`y' = `y'
local tmp `tmp' `moptobj'`y'
local dropmetoo `dropmetoo' `moptobj'`y'
}
else {
local tmp `tmp' `y'
}
}
local lhs `tmp'

```

```

    }
- if `waldtest' > 0 & `:length local lf0' {
= if 0 > 0 & 0 {
    opts_exclusive "waldtest() lf0()"
}
- if `:length local continue' {
= if 0 {
    if !`:length local init' {
        tempname init
        local ub0 `init'
    }
    else tempname ub0
    mat `ub0' = e(b)
    tempname v0 iv0
    mat `v0' = e(V)
    mat `iv0' = syminv(`v0')
    global ML_rank0 = colsof(`v0') - diag0cnt(`iv0')
    if !`:length local lf0' & !missing(e(ll)) {
        local lf0 `=colsof(`ub0')' `e(ll)'
    }
    local ub0
}
- if `:length local init' {
= if 14 {
- tempname b0
- local dim : list sizeof colna
- matrix `b0' = J(1,`dim',0)
= matrix __000007 = J(1,2,0)
- matrix colna `b0' = `colna'
= matrix colna __000007 = y_timing_max:_cons /:lnalpha
- `vv' _mkvec `b0', from(`init') update first error("initial vector"
> )
    = version 18: _mkvec __000007, from(__000003, copy) update first err
> or("initial vector")

> ----- begin _mkvec -----
- local vv : display "version " string(_caller()) ":"
- version 6
- gettoken matout 0 : 0, parse(", ")
- if "`matout'"==" | "`matout'"=="," {
= if "__000007"==" | "__000007"=="," {
    error 198
}
- syntax [, FROM(str) UPdate COLnames(str) ERRor(str) FIRST ]
- if "`update'"!=" {
= if "update"!=" {

```

```

- if "`colname'"!="" {
= if ""!="" {
  di in red "update and colnames() cannot both be " "specified"
  exit 198
}
- capture local colname : colfullnames `matout'
= capture local colname : colfullnames __000007
- if _rc {
  di in red "matrix `matout' not found"
  exit 111
}
- if rowsof(matrix(`matout')) != 1 {
= if rowsof(matrix(__000007)) != 1 {
  di in red "`matout' matrix to be updated must be " "a row vector
> "

  exit 503
}
- local matin `matout'
= local matin __000007
- }
- if ``error'""!="" {
= if ``initial vector'""!="" {
- local error ``error': ""
= local error ``initial vector: ""
- }
- tempname mat
- if ``from'""!="" {
= if ``__000003, copy'""!="" {
- From `from'
= From __000003, copy
}

> ----- begin _mkvec.From -----
- gettoken from 0 : 0, parse(",") bind
- sret clear
- sret local from `from'
= sret local from __000003
- if ``0'""!="" {
= if `", copy'""!="" {
- syntax [, COPY SKIP ]
- }
- sret local copy `copy'
= sret local copy copy
- sret local skip `skip'
= sret local skip
}

> ----- end _mkvec.From -----

```

```

- local copy `s(copy)'
= local copy copy
- local skip `s(skip)'
= local skip
- if "`copy'"!="" {
= if "copy"!="" {
- CopyMat `mat' ``error'' `s(from)'
= CopyMat __000008 `initial vector: ' __000003

```

---

```

> ----- begin _mkvec.CopyMat -----
- args matall error
- tempname matadd
- local i 3
- while ``i'!='' {
= while "__000003"!=" {
- capture di matrix(`i'[1,1])
= capture di matrix(__000003[1,1])
- if _rc == 0 {
- ChkMat `matadd' `i' ``error''
= ChkMat __000009 __000003 `initial vector: '

```

---

```

> ----- begin _mkvec.ChkMat -----
- args matout matin error
- capture di matrix(`matin'[1,1])
= capture di matrix(__000003[1,1])
- if _rc {
capture confirm number `matin'
if _rc {
di in red ``error'matrix `matin' not found''
exit 111
}
di in red ``error'list of numbers requires copy option''
exit 198
}
- if matrix(rowsof(`matin')!=1 & colsof(`matin')!=1) {
= if matrix(rowsof(__000003)!=1 & colsof(__000003)!=1) {
di in red "`matin' " matrix(rowsof(`matin')) " x " matrix(co
> lsof(`matin')) " is not a vector"
exit 503
}
- if matrix(rowsof(`matin')) == 1 {
= if matrix(rowsof(__000003)) == 1 {
- mat `matout' = `matin'
= mat __000009 = __000003
- }
- else mat `matout' = `matin''

```

```

= else mat __000009 = __000003'
> ----- end _mkvec.ChkMat -----
- }
- else {
  capture confirm number ``i''
  if _rc {
    di in red ``error'copy option requires '' "either a matrix or
> a list of numbers"
    exit 198
  }
  mat `matadd' = J(1,1,``i'')
  }
- mat `matall' = nullmat(`matall') , `matadd'
= mat __000008 = nullmat(__000008) , __000009
- local i = `i' + 1
= local i = 3 + 1
- }
- while ``i''!= "" {
= while ""!= "" {
  capture di matrix(`i'[1,1])
  if _rc == 0 {
    ChkMat `matadd' ``i'' ``error''
  }
  else {
    capture confirm number ``i''
    if _rc {
      di in red ``error'copy option requires '' "either a matrix or
> a list of numbers"
      exit 198
    }
    mat `matadd' = J(1,1,``i'')
    }
    mat `matall' = nullmat(`matall') , `matadd'
    local i = `i' + 1
  }
}
> ----- end _mkvec.CopyMat -----
- if ``colname''!= "" {
= if ``y_timing_max:_cons /lnalpha''!= "" {
- local dim : list sizeof colname
- if `dim' != colsof(matrix(`mat')) {
= if 2 != colsof(matrix(__000008)) {
  di in red ``error'matrix must '' ``be dimension '' `dim'
  exit 503
}
}

```

```

- `vv' mat colnames `mat' = `colname'
= version 18: mat colnames __000008 = y_timing_max:_cons /lnalpha
- }
- }
- else {
  `vv' BuildMat `mat' ``error'"" ``skip'"" `s(from)'
  if ``colname'""!="" {
    `vv' ChkNames `mat' ``matin'"" ``colname'"" ``error'"" ``skip'""
> ``first'""
    local fill `s(k_fill)'
    }
    }
- }
- else {
  local fill 0
  if ``update'""!="" {
    mat `mat' = `matin'
  }
  else if ``colname'""!="" {
    local dim : list sizeof colname
    mat `mat' = J(1,`dim',0)
    `vv' mat colnames `mat' = `colname'
  }
  else {
    di in red "from() must be specified"
    exit 198
  }
}
- sret clear
- sret local copy `copy'
= sret local copy copy
- sret local skip `skip'
= sret local skip
- sret local k = colsof(matrix(`mat'))
= sret local k = colsof(matrix(__000008))
- if ``fill'""!="" {
= if ""!="" {
  sret local k_fill `fill'
}
- else sret local k_fill = colsof(matrix(`mat'))
= else sret local k_fill = colsof(matrix(__000008))
- matrix rename `mat' `matout', replace
= matrix rename __000008 __000007, replace


---


> 

---

 end _mkvec 

---


- }

```

```

- tempname bomit
- matrix `bomit' = J(1,`:list sizeof colna',0)
= matrix __000008 = J(1,2,0)
- matrix colna `bomit' = `colna'
= matrix colna __000008 = y_timing_max:_cons /:lnalpha
- _ms_omit_info `bomit'
= _ms_omit_info __000008
- if r(k_omit) | `:length local constraints' {
= if r(k_omit) | 0 {
    nobreak {
        capture noisily break {
            _b_post0 `colna'
            local clist : subinstr local constraints "," " ", all
            `vv' makecns `clist', `cnsnotes'
            local constraints ``"r(clist)""
            local k_autoCns = e(k_autoCns)
            tempname C
            capture matrix `C' = e(Cns)
            local rcc = c(rc)
        }
        local rc = c(rc)
        ereturn clear
    }
    if (`rc') exit `rc'
    if (`rcc') {
        local C
        local k_autoCns
    }
}
- if `:length local constraints' {
= if 0 {
    local lf0
}
- local vce `vce' `robust' `cluster'
= local vce robust
- if `:length local vce' {
= if 6 {
- _vce_parse, argoptlist(CList) optlist(OIM OPG Robust) old : `wt'
> , `vce'
= _vce_parse, argoptlist(CList) optlist(OIM OPG Robust) old : , ro
> bust

> ----- begin _vce_parse -----
- version 10
- _on_colon_parse `0'
= _on_colon_parse , argoptlist(CList) optlist(OIM OPG Robust) ol

```

```

> d : , robust
> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `(`before')'
}
- if ``before' == ":" {
= if `", argoptlist(CLister) optlist(OIM OPG Robust) old "" == "
> ":" {
  sreturn local after ``after'
  exit
}
- while ``COLON' != ":" & ``after' != "" {
= while ``' != ":" & `": , robust' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quote
> s
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``COLON' != ":" {
= else if `":' != ":" {
  local before ``before' `COLON'
  local COLON
}
- }
- while ``COLON' != ":" & ``after' != "" {
= while `":' != ":" & `", robust' != "" {
  gettoken COLON after : after, parse(":") bind match(par) quote
> s
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``COLON' != ":" {
    local before ``before' `COLON'
    local COLON
  }
}
- if ``COLON' != ":" {

```



```

= if `":' != ":" {
    di as err "' found where ':' expected"
    exit 198
}
- sreturn local before `"'before'"
= sreturn local before `", argoptlist(CLister) optlist(OIM OPG R
> obust) old "'
- sreturn local after `"'after'"
= sreturn local after `", robust"'

> _____ end _on_colon_parse _____

- local 0 `"'s(before)'"
= local 0 `", argoptlist(CLister) optlist(OIM OPG Robust) old "'
- local user `"'s(after)'"
= local user `", robust"'
- syntax [varname(default=none)], [ OPTlist(namelist) ARGOPTlist(n
> amelist) PWallowed(namelist) OLD_is_ok NOARGOPT_is_ok ]
- local touse `varlist'
= local touse
- tempname o a
- .`o' = ._optlist.new
= .__000009 = ._optlist.new
- local optlist : list uniq optlist
- foreach name of local optlist {
- .`o'.addopt `name'
= .__000009.addopt OIM

> _____ begin _optlist.addopt _____

- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OIM")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "OIM"
- .names.Arrpush "`mname'"
= .names.Arrpush "oim"

> _____ end _optlist.addopt _____

- }
- .`o'.addopt `name'
= .__000009.addopt OPG

```

```

> ----- begin _optlist.addopt -----
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("OPG")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "OPG"
    - .names.Arrpush "`mname'"
    = .names.Arrpush "opg"
-----

> ----- end _optlist.addopt -----
    - }
    - .`o'.addopt `name'
    = .__000009.addopt Robust
-----

> ----- begin _optlist.addopt -----
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("Robust")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "Robust"
    - .names.Arrpush "`mname'"
    = .names.Arrpush "robust"
-----

> ----- end _optlist.addopt -----
    - }
    - .`a' = ._optlist.new
    = .__00000A = ._optlist.new
    - local argoptlist : list uniq argoptlist
    - foreach name of local argoptlist {
    - .`a'.addopt `name'
    = .__00000A.addopt CLuster
-----

> ----- begin _optlist.addopt -----
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")

```

```

= local mname = lower("CLuster")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
  }
- .options.Arrpush "`optspec'"
= .options.Arrpush "CLuster"
- .names.Arrpush "`mname'"
= .names.Arrpush "cluster"

```

---

```

> ----- end _optlist.addopt -----
- }
- local OPTNAMES ```.`o'.dumpnames'''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/3 {
- local namelist `namelist' `.names[`i']'
= local namelist oim
- }
- local namelist `namelist' `.names[`i']'
= local namelist oim opg
- }
- local namelist `namelist' `.names[`i']'
= local namelist oim opg robust
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist `"oim opg robust"'
- class exit ```namelist'''
= class exit `"oim opg robust"'

```

---

```

> ----- end _optlist.dumpnames -----
= local OPTNAMES `"oim opg robust"'
- local ARGOPTNAMES ```.`a'.dumpnames'''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/1 {
- local namelist `namelist' `.names[`i']'
= local namelist cluster
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''

```

```

= sreturn local namelist `"cluster"'
- class exit ``namelist''
= class exit `"cluster"'

```

---

```

> ----- end _optlist.dumpnames -----
= local ARGOPTNAMES `"cluster"'
- local AND : list OPTNAMES & ARGOPTNAMES
- if `:list sizeof AND' {
= if 0 {
  di as err "vcetype `:word 1 of `AND'' specified twice"
  exit 198
}
- local pwallowed : list uniq pwallowed
- local pwlist robust cluster
- foreach n of local pwallowed {
  local lc = lower("`n'")
  local pwlist `pwlist' `lc'
}
- if "`old_is_ok'" != "" {
= if "old_is_ok" != "" {
- local ROBUST robust
- if `:list ROBUST in OPTNAMES' {
= if 1 {
- local robust Robust
- }
- local CLUSTER cluster
- if `:list CLUSTER in ARGOPTNAMES' {
= if 1 {
- local cluster CCluster(varname)
- }
- }
- local 0 ``user''
= local 0 `` , robust''
- syntax [fw aw pw iw] [, VCE(string asis) `robust' `cluster' ]
= syntax [fw aw pw iw] [, VCE(string asis) Robust CCluster(varname)

> ]

- if ``vce'' != "" {
= if `''' != "" {
  gettoken name args : vce, parse(" ,")
  local args : list retok args
  capture ParseVCE `o' `a' , `name'
  if c(rc) {
    di as err "vcetype {bf:`name'} not allowed"
    exit 198
  }
  local name `r(vce)'

```

```

        if "`weight'" == "pweight" & !`:list name in pwlist' {
            di as err "pweights are not allowed with vce(`name')"
            exit 198
        }
        return add
        if ``"robust"' != "" & !inlist("`name'", "", "robust", "eim", "oim")
> , "cluster") {
            opts_exclusive "`robust' vce(`name')"
        }
        if "`name'" == "cluster" {
            if ``"args"' == "" {
                di as err "invalid vce(cluster) option"
                exit 198
            }
            confirm var `args'
            unab args : `args'
            local markout `args'
            if "`cluster'" != "" {
                if !`:list args == cluster' {
                    di as err "options cluster() and vce(cluster) are in conflict"
                    exit 100
                }
            }
            else if "`cluster'" != "" & !inlist("`name'", "eim", "oim") {
                opts_exclusive "cluster() vce(`name')"
            }
            if ``"args"' != "" {
                if !`:list name in ARGOPTNAMES' {
                    di as err "invalid vce() option"
                    exit 198
                }
                if "`name'" == "hc2" {
                    confirm var `args'
                    unab args : `args'
                    local markout `args'
                }
                return local vceargs ```:list retok args'''
            }
            else if `:list name in ARGOPTNAMES' {
                if "`noargopt_is_ok'" == "" {
                    di as err "invalid vce() option"
                    exit 198
                }
            }
        }
    }
}

```

```

- else {
- if "`weight'" == "pweight" | "`robust'" != "" {
= if "" == "pweight" | "robust" != "" {
- local vce robust
- }
- if "`cluster'" != "" {
= if "" != "" {
    local vce cluster
    local args `cluster'
    local markout `cluster'
  }
- return local vce ``vce'''
= return local vce `robust'
- return local vceargs ``args'''
= return local vceargs `'''
- }
- if "`return(vce)'" != "" {
= if "robust" != "" {
- if ``return(vceargs)''' != "" {
= if `''' != "" {
    local ss " "
  }
- return local vceopt `vce(`return(vce)''ss'`return(vceargs)')'
= return local vceopt `vce(robust)''
- }
- if "`return(vce)'" == "cluster" {
= if "robust" == "cluster" {
    return local cluster `return(vceargs)'
  }
- if inlist("`return(vce)'", "robust", "cluster") {
= if inlist("robust", "robust", "cluster") {
- return local robust robust
- }
- if "`touse'" != "" {
= if "" != "" {
    if "`markout'" != "" {
      markout `touse' `markout', strok
    }
  }
}

```

---

> \_\_\_\_\_ end \_vce\_parse \_\_\_\_\_

```

- local vce `r(vce)'
= local vce robust
- }
- else if "`weight'" == "pweight" {
= else if "" == "pweight" {

```

```

    local vce robust
  }
- if `:length local svy' {
= if 0 {
    local vce svy
  }
- if ``"vce" == "cluster" {
= if ``"robust" == "cluster" {
    local cluster `r(cluster)'
    markout $ML_sample `cluster', strok
  }
- if `:length local group' {
= if 0 {
    markout $ML_sample `group', strok
    global ML_group : copy local group
    if bsubstr("`:type `group'",1,3) == "str" {
      if `interactive' {
        SetGlobalVar ML_grp : _MLgr
      }
    }
    else {
      global ML_grp : tempvar
      if ``"moptobj" != "" {
        global ML_grp `moptobj'$ML_grp
      }
    }
    quietly egen $ML_grp = group(`group')
    if ! `:length local cluster' {
      local cluster : copy global ML_grp
    }
  }
  else {
    if ! `:length local cluster' {
      local cluster : copy global ML_group
    }
  }
}
- if "$ML_samp" != "$ML_sample" {
= if "__000005" != "__000005" {
    quietly replace $ML_samp = 0 if $ML_samp & $ML_sample == 0
  }
- if `:length local vce' == 0 {
= if 6 == 0 {
    if ``"technique" == "bhhh" {
      local vce opg
    }
    else local vce oim
  }
}

```

```

    }
    - if inlist("`vce'", "robust", "cluster") & bsubstr("`evaltype'",1,1
> ) == "d" {
    = if inlist("robust", "robust", "cluster") & bsubstr("e2",1,1) == "d
> " {
        local optname = cond("`svy'"=="", "vce(`vce')", "svy")
        di as err "option `optname' is not allowed with evaltype `evaltype
> '"
        exit 198
    }
    - if `:length local crittype' == 0 {
    = if 0 == 0 {
    - if inlist("`vce'", "cluster", "robust", "svy") {
    = if inlist("robust", "cluster", "robust", "svy") {
    - local crittype "log pseudolikelihood"
    - }
    - else local crittype "log likelihood"
    - }
    - global ML_waldtest `waldtest'
    = global ML_waldtest 0
    - if `:length local svy' | `:length local weight' {
    = if 0 | 0 {
        if `interactive' {
            SetGlobalVar ML_w : _MLw
        }
        else {
            global ML_w : tempvar
            if ``moptobj' != "" {
                global ML_w `moptobj'$ML_w
            }
        }
    }
    - if `:length local weight' {
    = if 0 {
        if ``moptobj' != "" {
            cap drop $ML_w
        }
        quietly gen double $ML_w `exp' if $ML_samp
        local wtype ``weight'
        local wexp ``exp'
        global ML_wtyp : copy local wtype
        global ML_wexp : copy local wexp
        if "`wtype'" == "iweight" | "`svy'" != "" {
            local zeroweight zeroweight
        }
    }
}

```



```

- if inlist("`vce'", "svy", "cluster") {
= if inlist("robust", "svy", "cluster") {
  if `interactive' {
    SetGlobalVar ML_subv : _MLsu
  }
  else {
    global ML_subv : tempvar
    if ``moptobj' != "" {
      global ML_subv `moptobj'$ML_subv
    }
  }
}
- quietly count if $ML_sample
= quietly count if __000005
- if r(N) == 0 {
  error 2000
}
- if ``weight' != "" {
= if `'' != "" {
  quietly count if $ML_sample & $ML_w != 0
  if r(N) == 0 {
    di in gr "all observation have zero weights"
    exit 2000
  }
}
- if `:length local lf0' {
= if 0 {
  if `:list sizeof lf0' != 2 {
    di as err "lf0(`lf0') invalid"
    exit 198
  }
  global ML_k0 : word 1 of `lf0'
  global ML_ll_0 : word 2 of `lf0'
  confirm integer number $ML_k0
  confirm number $ML_ll_0
  if $ML_k0 < 0 {
    di as err "lf0(`lf0') invalid"
    exit 198
  }
}
- if "`obs'" == "" {
= if "" == "" {
- if "`wtype'" != "fweight" {
= if "" != "fweight" {
- quietly count if $ML_sample
= quietly count if __000005

```

```

- local obs = r(N)
- }
- else {
  sum $ML_w if $ML_sample, mean
  local obs = r(sum)
}
- }
- if `:length local iterid' {
= if 0 {
  local iterid = trim(bsubstr("`iterid'",1,32))
}
- if `:length local crittype' {
= if 20 {
- local crittype = trim(bsubstr("`crittype'",1,32))
= local crittype = trim(bsubstr("`log pseudolikelihood'",1,32))
- }
- if `:length local derivscale' {
= if 0 {
  confirm matrix `derivscale'
}
- if `:length local derivh' {
= if 0 {
  confirm matrix `derivh'
}
- local l : strlen local deriv
- ParseUsemin, `deriv' len(`l')
= ParseUsemin, len(0)
}

> ----- begin mopt.ParseUsemin -----
- capture syntax [, LEN(integer 0), MIN1(numlist min=2 max=2 missi
> ngokay) NOMIN MIN]
- if (_rc) {
  syntax [, LEN(integer 0) MIN1(string) NOMIN MIN]
}
- if ``min1'' != "" & ``nomin'' != "" {
= if `""' != "" & `""' != "" {
  di as err "options {bf:min()} and {bf:nomin} are mutually exclus
> ive"
  exit 198
}
- if ``min1'' != "" & ``nomin'' != "" {
= if `""' != "" & `""' != "" {
  di as err "options {bf:min} and {bf:nomin} are mutually exclusiv
> e"
  exit 198
}

```

```

- if (``min1'' != '') {
= if (``'' != '') {
    if (_rc) {
        capture confirm matrix `min1'
        if (_rc) {
            di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
            exit 198
        }
    }
    local deriv `min1'
}
- else if (``min'' != '') {
= else if (``'' != '') {
    if (`len' > length(`min')) {
        di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
        exit 198
    }
    local deriv 1
}
- else {
- if (`len' > 0) {
= if (0 > 0) {
    if (``nomin'' == '') {
        di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
        exit 198
    }
    else if (`len' > length(`nomin')) {
        di as err "options {bf:min()} and {bf:nomin} are mutually exclus
> ive"
        exit 198
    }
    else {
        local deriv 0
    }
}
- else {
- local deriv 0
- }
- }
- sreturn local deriv `deriv'
= sreturn local deriv 0

```

---

```

> _____ end mopt.ParseUsemin _____

```

```

- local deriv = s(deriv)
- if `:length local gnwmatrix' {
= if 0 {
  confirm matrix `gnwmatrix'
}
- `vv' mata: Mopt_model()
= version 18: mata: Mopt_model()

> ----- begin mopt_check_program -----
- version 10
- args name
- capture confirm name `name'
= capture confirm name nbreg_lf
- if c(rc) {
  di as err "unrecognized command:  `name' invalid command name"
  exit 199
}
- capture which `name'
= capture which nbreg_lf
- if !c(rc) {
- exit

> ----- end mopt_check_program -----
- global ML_stat model
- if !`interactive' {
= if !0 {
- `vv' MaxMin `maxmin', nooutput `score' `options' moptobj(`moptobj'
> )
= version 18: MaxMin maximize, nooutput  search(off) moptobj()

> ----- begin mopt.MaxMin -----
- local vv : di "version " string(_caller()) ":"
- version 12
- ChkMacros

> ----- begin mopt.ChkMacros -----
- if "$ML_stat" != "model" {
= if "model" != "model" {
  di as err "you must issue -ml model- first"
  exit 198
}
- capture confirm var $ML_samp $ML_w $ML_subv
= capture confirm var __000005 __000005
- if c(rc) {
  di as err "Since issuing the -ml model- statement, you have do
> ne something to drop" _n "temporary variables ml added to your data.  You mu

```

```

> st start again."
    Clear
    exit 111
}

> ----- end mopt.ChkMacros -----
    - syntax name(name=maxmin) [, NEGH TOLerance(numlist max=1 >=0) LT
> OLerance(numlist max=1 >=0) NONRTOLerance NRTOLerance(numlist max=1 >=0) QTO
> Lerance(numlist max=1 >=0) GTOLerance(numlist max=1 >=0) ITERate(numlist max
> =1 >=0) NOTCONCAVE(numlist max=1 >=0) ndami DIFFicult HALFSTEPonly DOOPT SEA
> rch(name) Repeat(passthru) Bounds(string) SCore(string) noWARNING noCLEAR NO
> LOf LOf DOTs SHOWTOLerance SHOWNRTolerance SHOWSTEP TRace COEFDiffs GRADient
> HESSian noSKIpline noOUTput Level(cilevel) moptobj(string) * ]
    - if `:length local nrtolerance' {
= if 0 {
    opts_exclusive "nrtolerance() `nonrtolerance'"
}
    - if `:length local qtolerance' {
= if 0 {
    opts_exclusive "qtolerance() `nonrtolerance'"
}
    - opts_exclusive "`nonrtolerance' `shownrtolerance'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while `""`opts' != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - if `:length local shownrtolerance' {
= if 0 {
    local showtolerance showtolerance
}
    - if "$ML_preserve" != "no" {
= if "no" != "no" {

```

```

        if `:length local score' {
            di as smcl as err "May not specify {bf:score()} option unless" _
> n "      a) estimation subsample is the entire data in memory, or" _n "      b
> ) you specify option {bf:nopreserve} on {bf:ml model} statement (meaning" _n
> "      your evaluation program explicitly restricts itself to obs" _n "
>      for which {bf:$}" "{bf:ML_samp==1}."
            exit 198
        }
        preserve
        local N = c(N)
        quietly keep if $ML_sample
        if (`N' != c(N)) {
            mata: Mopt_init_regetviews()
        }
    }
- if `:length local score' {
= if 0 {
    SetupScore `"$ML_evaltype"' ``score'"' $ML_n $ML_dim
    local scores "`s(scores)'"
}
- _parse_iterlog, `nolog' `log'
= _parse_iterlog,

```

---

```

> ----- begin _parse_iterlog -----
    - version 16
    - syntax [, NOLOG LOG]
    - if "`log'`nolog'" != "" {
= if "" != "" {
    opts_exclusive "`log' `nolog'"
    }
    - else if "`c(iterlog)'" == "on" {
= else if "on" == "on" {
    - local log log
    - }
    - else {
        local nolog nolog
    }
    - sreturn local log `log'
= sreturn local log log
    - sreturn local nolog `nolog'
= sreturn local nolog

```

---

```

> ----- end _parse_iterlog -----
    - local log "`s(nolog)'"
= local log ""
    - local tropts `log' `dots' `showtolerance' `showstep' `trace' `co

```

```

> efdiffs'
= local tropts
- if "`tropts'" != "nolog" {
= if "" != "nolog" {
- local tropts `tropts' `gradient' `hessian'
= local tropts
- if "`skipline'" == "" {
= if "" == "" {
- if "`iterate'" != "0" | `:length local tropts' {
= if "" != "0" | 0 {
- di

- }
- }
- }
- else {
  local gradient
  local hessian
  local nolog nolog
}
- if "`search'" != "off" & "`iterate'" != "0" {
= if "off" != "off" & "" != "0" {
  if "`search'" == "quietly" {
    local sopts nolog
  }
  else {
    if `:length local tropts' {
      if "`tropts'" != "nolog" {
        local sopts trace
      }
      else local sopts nolog
    }
    if "`search'" == "norescale" {
      local sopts `sopts' norescale
    }
    Search `bounds', nopreserve `sopts' `repeat' `maxmin'
  }
- tempname value b V iV
- `vv' mata: Mopt_maxmin()
= version 18: mata: Mopt_maxmin()

-----
> ----- begin nbreg_lf -----
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb

```

```

- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
  - }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __00000K==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000U = exp(__00000Q) if __000005
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000V = 1/(1+exp(__00000Q+__00000R)) if __000005
- if `lnalpha'<-20 {
= if __00000R<-20 {
    replace `sc1' = `y'-`mu' if $ML_samp
    replace `sc2' = 0 if $ML_samp
  }
- else {
    - replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000N = __00000V*(y_timing_max-__00000U) if __00000
> 5
    - replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
    = replace __000000 = __00000S*(digamma(__00000S) - digamma(y_tim
> ing_max+__00000S) - ln(__00000V)) + __00000V*(y_timing_max-__00000U) if __00
> 0005
  - }
- }
- $ML_ec tempname g1 g2

```



```

= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __00000K = __00000N, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __00000K = __00000O, eq(2)
- $ML_ec matrix `g' = (`g1', `g2')
= * matrix __00000L = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __00000K==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __00000W = exp(__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
  mlmatsum `lnf' `d11' = `mu', eq(1)
  mlmatsum `lnf' `d12' = 0, eq(1,2)
  mlmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mlmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
= mlmatsum __00000K __00000X = __00000U*__00000V*(__00000W*__000
> 00V*(y_timing_max-__00000U)+1), eq(1)
- mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
= mlmatsum __00000K __00000Y = __00000W*__00000U*__00000V^2*(y_t
> iming_max-__00000U), eq(1,2)
- mlmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*(`
> `y'-`mu')- 1), eq(2)
= mlmatsum __00000K __00000Z = __00000S*(digamma(__00000S) - dig
> amma(y_timing_max+__00000S) - ln(__00000V) - __00000S*(trigamma(y_timing_max
> +__00000S) - trigamma(__00000S))) + __00000U*__00000V*(__00000W*__00000V*(y_
> timing_max-__00000U)- 1), eq(2)
- }
- matrix `H' = (`d11', `d12' \ `d12'', `d22')
= matrix __00000M = (__00000X, __00000Y \ __00000Y', __00000Z)

> _____ end nbreg_lf _____
Iteration 0: Log pseudolikelihood = -446.52804

> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)

```

```

= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
  mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
  = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
  - }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }

> _____ end nbreg_lf _____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
  mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
  = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))

```

```

- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }
_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
mlsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsum __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }
_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"

```

```

- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
    - }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }

_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
    - }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __00000K==. { exit }
- tempname alpha
- tempvar mu p

```

```

- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000U = exp(__00000Q) if __000005
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000V = 1/(1+exp(__00000Q+__00000R)) if __000005
- if `lnalpha'<-20 {
= if __00000R<-20 {
  replace `sc1' = `y'-`mu' if $ML_samp
  replace `sc2' = 0 if $ML_samp
}
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000N = __00000V*(y_timing_max-__00000U) if __00000
> 5
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
= replace __000000 = __00000S*(digamma(__00000S) - digamma(y_tim
> ing_max+__00000S) - ln(__00000V)) + __00000V*(y_timing_max-__00000U) if __00
> 0005
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __00000K = __00000N, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __00000K = __000000, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000L = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __00000K==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __00000W = exp(__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
  mlmatsum `lnf' `d11' = `mu', eq(1)
  mlmatsum `lnf' `d12' = 0, eq(1,2)
  mlmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mlmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
= mlmatsum __00000K __00000X = __00000U*__00000V*(__00000W*__000
> 00V*(y_timing_max-__00000U)+1), eq(1)
- mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)

```

```

= mlfsum __00000K __00000Y = __00000W*__00000U*__00000V^2*(y_t
> iming_max-__00000U), eq(1,2)
- mlfsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*(
> `y'-`mu')- 1), eq(2)
= mlfsum __00000K __00000Z = __00000S*(digamma(__00000S) - dig
> amma(y_timing_max+__00000S) - ln(__00000V) - __00000S*(trigamma(y_timing_max
> +__00000S) - trigamma(__00000S))) + __00000U*__00000V*(__00000W*__00000V*(y_
> timing_max-__00000U)- 1), eq(2)
- }
- matrix `H' = (`d11',`d12' \ `d12'',`d22')
= matrix __00000M = (__00000X,__00000Y \ __00000Y',__00000Z)

> _____ end nbreg_lf _____
Iteration 1: Log pseudolikelihood = -414.23222

> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
mlsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlfsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlfsum __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }

> _____ end nbreg_lf _____

> _____ begin nbreg_lf _____

```

```

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlsun `lnf' = -lgamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    - mlsun `lnf' = lgamma(`m'+`y') - lgamma(`y'+1) - lgamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __00000K = lgamma(__00000S+y_timing_max) - lgamma(y_ti
> ming_max+1) - lgamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 0 == 0 | __00000K==. { exit }
}

> _____ end nbreg_lf _____

> _____ begin nbreg_lf _____

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlsun `lnf' = -lgamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {

```

```

- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __00000K==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000U = exp(__00000Q) if __000005
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000V = 1/(1+exp(__00000Q+__00000R)) if __000005
- if `lnalpha'<-20 {
= if __00000R<-20 {
replace `sc1' = `y'-`mu' if $ML_samp
replace `sc2' = 0 if $ML_samp
}
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000N = __00000V*(y_timing_max-__00000U) if __00000
> 5
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
= replace __000000 = __00000S*(digamma(__00000S) - digamma(y_tim
> ing_max+__00000S) - ln(__00000V)) + __00000V*(y_timing_max-__00000U) if __00
> 0005
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __00000K = __00000N, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __00000K = __000000, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000L = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __00000K==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __00000W = exp(__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {

```



```

mlmatsum `lnf' `d11' = `mu', eq(1)
mlmatsum `lnf' `d12' = 0, eq(1,2)
mlmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mmlatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
= mmlatsum __00000K __00000X = __00000U*__00000V*(__00000W*__000
> 00V*(y_timing_max-__00000U)+1), eq(1)
- mmlatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
= mmlatsum __00000K __00000Y = __00000W*__00000U*__00000V^2*(y_t
> iming_max-__00000U), eq(1,2)
- mmlatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*(
> `y'-`mu')- 1), eq(2)
= mmlatsum __00000K __00000Z = __00000S*(digamma(__00000S) - dig
> amma(y_timing_max+__00000S) - ln(__00000V) - __00000S*(trigamma(y_timing_max
> +__00000S) - trigamma(__00000S))) + __00000U*__00000V*(__00000W*__00000V*(y_
> timing_max-__00000U)- 1), eq(2)
- }
- matrix `H' = (`d11',`d12' \ `d12'',`d22')
= matrix __00000M = (__00000X,__00000Y \ __00000Y',__00000Z)

> _____ end nbreg_lf _____
Iteration 2: Log pseudolikelihood = -414.22879
> _____ begin nbreg_lf _____

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
mlsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- msum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))

```

```

= mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }
_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __00000K==. { exit }
_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000Q = __00000J, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar

```

```

= mlevel __00000R = __00000J, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000S = exp(-__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
  mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
  mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
  = mlsun __00000K = lngamma(__00000S+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000S) - __00000S*ln1p(exp(__00000Q+__00000R)) - y_
> timing_max*ln1p(exp(-__00000Q-__00000R))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __00000K==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000U = exp(__00000Q) if __000005
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000V = 1/(1+exp(__00000Q+__00000R)) if __000005
- if `lnalpha'<-20 {
= if __00000R<-20 {
  replace `sc1' = `y'-`mu' if $ML_samp
  replace `sc2' = 0 if $ML_samp
}
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000N = __00000V*(y_timing_max-__00000U) if __00000
> 5
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
= replace __000000 = __00000S*(digamma(__00000S) - digamma(y_tim
> ing_max+__00000S) - ln(__00000V)) + __00000V*(y_timing_max-__00000U) if __00
> 0005
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __00000K = __00000N, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)

```

```

= * mlvectsum __00000K = __000000, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000L = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __00000K==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __00000W = exp(__00000R)
- if `lnalpha' < -20 {
= if __00000R < -20 {
    mlmatsum `lnf' `d11' = `mu', eq(1)
    mlmatsum `lnf' `d12' = 0, eq(1,2)
    mlmatsum `lnf' `d22' = 0, eq(2)
  }
- else {
- mlmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
    = mlmatsum __00000K __00000X = __00000U*__00000V*(__00000W*__000
> 00V*(y_timing_max-__00000U)+1), eq(1)
    - mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
    = mlmatsum __00000K __00000Y = __00000W*__00000U*__00000V^2*(y_t
> iming_max-__00000U), eq(1,2)
    - mlmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*
> `y'-`mu')- 1), eq(2)
    = mlmatsum __00000K __00000Z = __00000S*(digamma(__00000S) - dig
> amma(y_timing_max+__00000S) - ln(__00000V) - __00000S*(trigamma(y_timing_max
> +__00000S) - trigamma(__00000S))) + __00000U*__00000V*(__00000W*__00000V*(y_
> timing_max-__00000U)- 1), eq(2)
    - }
    - matrix `H' = (`d11',`d12' \ `d12'',`d22')
    = matrix __00000M = (__00000X,__00000Y \ __00000Y',__00000Z)
  }
}
> _____ end nbreg_lf _____
Iteration 3: Log pseudolikelihood = -414.22879
> _____ begin mopt_post _____
- args b V C wgt touse depvar
- if `:length local touse' {
= if 8 {
- tempname esamp
- quietly gen byte `esamp' = `touse'
= quietly gen byte __00000S = __000005
- }
- ereturn post `b' `V' `C' `wgt', esample(`esamp') depname(`depv
> ar') buildfvinfo

```

```

= ereturn post __00000Q __00000R [], esample(__00000S) depname(
> y_timing_max) buildfvinfo


---


> 

---

 end mopt_post 

---


- ereturn scalar k_eq_model = abs($ML_waldtest)
= ereturn scalar k_eq_model = abs(0)
- if `:length global ML_ll_0' {
= if 0 {
ereturn scalar ll_0 = $ML_ll_0
if `:length global ML_rank0' {
ereturn scalar rank0 = $ML_rank0
ereturn scalar df_m = e(rank) - $ML_rank0
}
else ereturn scalar df_m = e(rank) - $ML_k0
}
- if !`:length global ML_ll_0' | inlist(e(vce), "robust", "cluster
> ", "linearized") {
= if !0 | inlist(e(vce), "robust", "cluster", "linearized") {
- if "$ML_svy" == "" {
= if "" == "" {
- _prefix_model_test, minimum


---


> 

---

 begin _prefix_model_test 

---


- version 9
- syntax [name(name=cmd)] [, svy noADJust noSVYadjust MINimum ]
- if (inlist("`cmd'", "asmixlogit", "cmmixlogit", "cmxtmixlogit"))
> {
= if (inlist("", "asmixlogit", "cmmixlogit", "cmxtmixlogit")) {
_asmixtest, `svy' `adjust' `svyadjust'
exit
}
- local K = 1
- if "`e(k_eq_model)'" != "" {
= if "0" != "" {
- capture confirm integer number `e(k_eq_model)'
= capture confirm integer number 0
- if ! c(rc) {
- local K = e(k_eq_model)
- }
- }
- if "`e(k_eq_model_skip)'" != "" {
= if "" != "" {
capture confirm integer number `e(k_eq_model_skip)'
if ! c(rc) {
local skip = e(k_eq_model_skip)
}
}
}

```

```

    }
    - ereturn local df_m
    - ereturn local F
    - ereturn local chi2
    - ereturn local chi2type
    - ereturn local p
    - if (`K' == 0) {
= if (0 == 0) {
    - exit
}

> ----- end _prefix_model_test -----

- }
- else {
    _prefix_model_test, $ML_svy $ML_noadj
}
- }
- else if `:length global ML_ll_0' {
= else if 0 {
    if ``"maxmin"' == "minimize" {
        ereturn scalar chi2 = -2*(e(ll)-e(ll_0))
    }
    else {
        ereturn scalar chi2 = 2*(e(ll)-e(ll_0))
    }
    ereturn scalar p = chi2tail(e(df_m), e(chi2))
    ereturn local chi2type "LR"
}
- if `:length global ML_wexp' {
= if 0 {
    ereturn local wtype ``$ML_wtyp"'
    ereturn local wexp ``$ML_wexp"'
}
- if `:length global ML_group' {
= if 0 {
    ereturn local group ``$ML_group"'
    if ``"e(clustvar)"' == "$ML_grp" & "$ML_grp" != "" {
        ereturn local clustvar ``$ML_group"'
    }
}
- ereturn local cmd "ml"
- if `:length local score' {
= if 0 {
    mata: Mopt_score(0)
    ereturn local scorevars `scores'
}
- if "$ML_preserve" != "no" {

```

```

= if "no" != "no" {
  restore
  if "`e(prefix)'" == "svy" & e(N_strata_omit) {
    tempname v
    quietly _svy2 $ML_sample if $ML_sample, type(total) svy v(`v') s
> ubpop(`e(subpop)') touse($ML_sample)
  }
  ereturn repost, esample($ML_sample)
}
- if ``moptobj'"" != "" {
= if `""' != "" {
  local clear noclear
}
- if !':length local clear' {
= if !0 {
- Clear

> ----- begin mopt.Clear -----
- syntax [, NOTANOPTION]
- capture drop $ML_w
= capture drop __000005
- capture drop $ML_samp
= capture drop __000005
- capture drop $ML_sample
= capture drop __000005
- capture drop $ML_subv
= capture drop
- capture drop $ML_grp
= capture drop
- mata: Mopt_drop_external()
- macro drop ML_*

> ----- end mopt.Clear -----
- }
- else {
  capture confirm var $ML_sample
  if c(rc) {
    quietly gen byte $ML_sample = e(sample)
  }
}
- if `':length local output' == 0 {
= if 8 == 0 {
  Display, level(`level') `options'
}

> ----- end mopt.MaxMin -----

```

```

- }


---


> 

---

 end mopt.Model 

---


- if c(rc) {
  local rc = c(rc)
  Clear
  exit `rc'
}
- exit


---


> 

---

 end mopt 

---


- exit


---


> 

---

 end ml 

---


- mat `b0' = e(b)
= mat __000003 = e(b)
- scalar `ll0' = e(ll)
= scalar __000004 = e(ll)
- local continu "continue"
- }
- if ``from'""=="" {
= if `""'=="" {
- if ``constan'""=="" {
= if ""=="" {
- local dim = colsof(`bp')
= local dim = colsof(__000002)
- if `dim' > 1 {
= if 9 > 1 {
- tempvar xb
- qui mat score `xb' = `bp' if `touse'
= qui mat score __000005 = __000002 if __000000
- if ``weight'""!=""" {
= if ""!=""" {
  local wt `[aw`exp']""
}
- summarize `xb' `wt' if `touse', meanonly
= summarize __000005 if __000000, meanonly
- if ``offset'""!=""" {
= if ""!=""" {
  qui replace `xb' = `xb' + `b0'[1,1] - r(mean) + `offset'
}
- else {
- qui replace `xb' = `xb' + `b0'[1,1] - r(mean)
= qui replace __000005 = __000005 + __000003[1,1] - r(mean)
- }
- mat `bp'[1,`dim'] = `bp'[1,`dim'] + `b0'[1,1] - r(mean)

```



```

= mat __000002[1,9] = __000002[1,9] + __000003[1,1] - r(mean)
- mat `bp' = (`bp', `b0'[1,2..2])
= mat __000002 = (__000002, __000003[1,2..2])
- `LLprog' `y' `xb' `b0'[1,2] [`weight'`exp'] if `touse', nobs(`r(N)')
= LLalpha y_timing_max __000005 __000003[1,2] [] if __000000, nobs(221)

```

---

```

> ----- begin nbreg.LLalpha -----
- gettoken y 0 : 0
- gettoken z 0 : 0
- gettoken b0 0 : 0
- syntax [fw aw pw iw] [if], Nobs(string)
- if "`weight'"!=" " {
= if ""!=" " {
  if "`weight'"=="fweight" {
    local wt `[fw`exp']"
  }
  else local wt `[aw`exp']"
}
- tempname lnalpha m
- scalar `lnalpha' = `b0'
= scalar __000006 = __000003[1,2]
- local bound -20
- if `lnalpha' < `bound' { scalar `lnalpha' = `bound' }
= if __000006 < -20 { scalar __000006 = -20 }
- scalar `m' = exp(-`lnalpha')
= scalar __000007 = exp(-__000006)
- qui replace `z' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m') - `
> m'*ln1p(exp(`z'+`lnalpha')) - `y'*ln1p(exp(-`z'-`lnalpha')) `if'
= qui replace __000005 = lngamma(__000007+y_timing_max) - lngamma(y_timi
> ng_max+1) - lngamma(__000007) - __000007*ln1p(exp(__000005+__000006)) - y_ti
> ming_max*ln1p(exp(-__000005-__000006)) if __000000
- summarize `z' `wt' `if', meanonly
= summarize __000005 if __000000, meanonly
- if r(N) != `nobs' { exit }
= if r(N) != 221 { exit }
- if "`weight'"=="aweight" {
= if ""=="aweight" {
  ret scalar lnf = r(N)*r(mean)
}
- else ret scalar lnf = r(sum)

```

---

```

> ----- end nbreg.LLalpha -----
- if r(lnf) > `ll0' & r(lnf)<. {
= if r(lnf) > __000004 & r(lnf)<. {
- local initopt "init(`bp')
= local initopt "init(__000002)"

```

```

- }
- }
- if "`initopt'"==" {
= if "init(__000002)"==" {
    local initopt "init(`b0')"
    }
- }
- else {
    tempname b0
    mat `b0' = (0)
    if _caller() < 15 {
    mat colnames `b0' = ln`parm':_cons
    }
    else {
    mat colnames `b0' = /ln`parm'
    }
    mat `bp' = (`bp',`b0')
    local initopt "init(`bp')"
    }
- `nohead' di in gr _n "Fitting full model:"
= di in gr _n "Fitting full model:"

```

Fitting full model:

```

- }
- else local initopt `"init(`from')"'
= else local initopt `"init()"'
- `vv' ml model `mm' `prog' (`yname': `y'=`xvars', `constan' `offopt') /ln
> `parm' if `touse' [weight`exp'], collinear missing max nooutput nopreserve
> `initopt' search(off) `mlopts' `scopt' `robust' `vce' `continu' title(`titl
> e') diparm(ln`parm', exp label("`parm'")) `negh' `moptobj'
= version 18: ml model e2 nbreg_lf (y_timing_max: y_timing_max=cb_authors_
> ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,
> ) /lnalpha if __000000 [], collinear missing max nooutput nopreserve init(_
> __000002) search(off) robust vce(robust) continue title(Negative binomial r
> egression) diparm(lnalpha, exp label("alpha")) negh
_____
> _____ begin ml _____
    - if _caller()>=11 {
    - mopt `0'
    = mopt model e2 nbreg_lf (y_timing_max: y_timing_max=cb_authors_ext esti
> mated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open, ) /lnal
> pha if __000000 [], collinear missing max nooutput nopreserve init(__000002)
> search(off) robust vce(robust) continue title(Negative binomial regressio
> n) diparm(lnalpha, exp label("alpha")) negh
_____
> _____ begin mopt _____

```

```

- local vv : di "version " string(_caller()) ":"
- version 12
- gettoken subcmd 0 : 0, parse(" ,")
- local lcmd : length local subcmd
- if "`subcmd'" == "clear" {
= if "model" == "clear" {
    Clear
    exit
  }
- if ``"subcmd"' == "hold" {
= if ``"model"' == "hold" {
    ml_hold `0'
    exit
  }
- if ``"subcmd"' == "unhold" {
= if ``"model"' == "unhold" {
    ml_unhold `0'
    exit
  }
- if "`subcmd'" == "score" {
= if "model" == "score" {
    if _caller() > 12 {
      `vv' Score `0'
    }
    else {
      Score `0'
    }
    exit
  }
- if "`subcmd'" == "count" {
= if "model" == "count" {
    Count `0'
    exit
  }
- if "`subcmd'" == "trace" {
= if "model" == "trace" {
    Trace `0'
    exit
  }
- if "`subcmd'" == bsubstr("graph",1,max(2,`lcmd')) {
= if "model" == bsubstr("graph",1,max(2,5)) {
    Graph `0'
    exit
  }
- if "`subcmd'" == "init" {
= if "model" == "init" {

```

```

Interactive 0
`vv' Init `0'
exit
}
- if "`subcmd'" == bsubstr("maximize",1,max(3,`lcmd')) {
= if "model" == bsubstr("maximize",1,max(3,5)) {
Interactive 1
`vv' MaxMin maximize `0'
exit
}
- if "`subcmd'" == bsubstr("minimize",1,max(3,`lcmd')) {
= if "model" == bsubstr("minimize",1,max(3,5)) {
Interactive 1
`vv' MaxMin minimize `0'
exit
}
- if "`subcmd'" == bsubstr("display",1,max(2,`lcmd')) {
= if "model" == bsubstr("display",1,max(2,5)) {
Display `0'
exit
}
- if "`subcmd'" == bsubstr("footnote",1,max(4,`lcmd')) {
= if "model" == bsubstr("footnote",1,max(4,5)) {
ml_footnote `0'
exit
}
- if "`subcmd'" == ", " {
= if "model" == ", " {
Display, `0'
exit
}
- if "`subcmd'" == bsubstr("model",1,max(3,`lcmd')) {
= if "model" == bsubstr("model",1,max(3,5)) {
- `vv' capture noi Model `0'
= version 18: capture noi Model e2 nbreg_lf (y_timing_max: y_timing_m
> ax=cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo
> ln_neq open, ) /lnalpha if __000000 [], collinear missing max nooutput nop
> reserve init(__000002) search(off) robust vce(robust) continue title(Negat
> ive binomial regression) diparm(lnalpha, exp label("alpha")) negh

> ----- begin mopt.Model -----
- local version = string(_caller())
- local vv : di "version `version':"
= local vv : di "version 18:"
- version 12
- syntax anything(id="model" name=spec) [if] [in] [fw aw pw iw] [, B

```

```

> RACKET COLlinear MISSing CONTinue lf0(string) INIT(string) CONSTraints(strin
> g) noCNSNOTEs VCE(passthru) NEGH NOPREserve TECHnique(string) ITERID(string)
> CRITtype(string) OBS(numlist max=1 >0) TITle(string) WALDtest(integer -1) S
> VY SUBpop(string) SRSSubpop noSVYadjust SCore(passthru) noOUTput noSCVARS GR
> OUP(varname) ITERPROLOG(string) DERIVPROLOG(string) DERIVH(string) DERIVSCAL
> E(string) DERIV(string) nrderiv GNWMATrix(string) KAUXiliary(numlist int >=0
> ) MAXimize MINimize USERinfo(namelist) PUPDATED Robust CLuster(passthru) mop
> tobj(string) USEVIEWS * ]
    - if ``moptobj`` == "" {
    = if ```` == "" {
    - Clear

> ----- begin mopt.Clear -----
    - syntax [, NOTANOPTION]
    - capture drop $ML_w
    = capture drop
    - capture drop $ML_samp
    = capture drop
    - capture drop $ML_sample
    = capture drop
    - capture drop $ML_subv
    = capture drop
    - capture drop $ML_grp
    = capture drop
    - mata: Mopt_drop_external()
    - macro drop ML_*

> ----- end mopt.Clear -----
    - }
    - local maxmin `minimize' `maximize'
    = local maxmin maximize
    - opts_exclusive ``maxmin''
    = opts_exclusive "maximize"

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts maximize
    - local n 0
    - while ``opts`` != "" {
    = while ``maximize'' != "" {
    - local ++n
    - gettoken item`n' opts : opts, bind
    = gettoken item1 opts : opts, bind
    - }

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----

- if `:length local maxmin' {
= if 8 {
- local interactive 0
- }
- else local interactive 1
- _get_diparmopts, diparmopts(`options') soptions syntaxonly
= _get_diparmopts, diparmopts(search(off) diparm(lnalpha, exp label(
> "alpha"))) soptions syntaxonly
}

> ----- begin _get_diparmopts -----

- if _caller() < 8.2 {
    _get_diparmopts_8 `0'
    exit
    }
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom plus S
> YNTAXonly * ]
- local goptions `options'
= local goptions
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk ,

> ----- begin _get_diparmopts.SpliceGlobals -----

- syntax name(id="macro name" name=c_diparm) [, diparm(string as
> is) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof'' == "" & ``e(df_r)'' != "" {
= if `"' == "" & `"' != "" {
    capture confirm integer number `e(df_r)'
    if !c(rc) {
        local gdof dof(`e(df_r)')
    }
    capture
    }
- else local gdof ``dof''
= else local gdof `"'

```

```

- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)
- if (`"'diparm'" == "") exit
= if (`"' == "") exit

```

---

```

> ----- end _get_diparmopts.SpliceGlobals -----
- local 0 `", `diparmopts'"
= local 0 `", search(off) diparm(lnalpha, exp label("alpha"))'"
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while (`"'diparm'" != "" {
= while (`"lnalpha, exp label("alpha")'" != "" {
- local ++k
- local single 1
- local after
- capture _on_colon_parse `diparm'
= capture _on_colon_parse lnalpha, exp label("alpha")

```

---

```

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `("before')'"
}
- if ``"before'" == ":" {
= if `"lnalpha, exp label("alpha")'" == ":" {
  sreturn local after ``"after'"
  exit
}
- while ``"COLON'" != ":" & ``"after'" != "" {
= while ``'" != ":" & ``'" != "" {
  gettoken COLON after : after, parse(":") bind match(par) quote

```

---

```

> s
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``"COLON'" != ":" {
    local before ``"before' `COLON'"
    local COLON

```

```

    }
    }
    - if ``COLON`` != ":" {
    = if `` != ":" {
    - di as err "" found where ':' expected"
    - exit 198
    }
}

> ----- end _on_colon_parse -----

- if !c(rc) {
  local single 0
  local diparm ``s(before)``
  local after ``s(after)``
}
- capture
- SpliceGlobals diparm , diparm(`diparm`) `syntaxonly` `goptions`
= SpliceGlobals diparm , diparm(lnalpha, exp label("alpha")) synta
> xonly

> ----- begin _get_diparmopts.SpliceGlobals -----
- syntax name(id="macro name" name=c_diparm) [, diparm(string as
> is) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof`` == "" & ``e(df_r)`` != "" {
= if `` == "" & `` != "" {
  capture confirm integer number `e(df_r)`
  if !c(rc) {
    local gdof dof(`e(df_r)`)
  }
  capture
}
- else local gdof ``dof``
= else local gdof ``
- local glevel `level`
= local glevel 95
- local lopt level(`glevel`)
= local lopt level(95)
- if (``diparm`` == "") exit
= if ("lnalpha, exp label("alpha)"" == "") exit
- local 0 `diparm`
= local 0 lnalpha, exp label("alpha")
- syntax anything(id="eqname(s)" name=eqspecs) [, Level(passthru
> ) dof(passthru) * ]
- if ("eqspecs" == "__sep__") exit
= if ("lnalpha" == "__sep__") exit
- if "level" == "" {
= if "" == "" {

```



```

- local lopt level(`glevel')
= local lopt level(95)
- }
- else local lopt `level'
= else local lopt
- if ```dof'"" == "" {
= if `""' == "" {
- local dof ```gdof'""
= local dof `""'
- }
- if "syntaxonly" == "" {
= if "syntaxonly" == "" {
  c_local `c_diparm' ```eqspecs', `options' `lopt' `dof'""
  }
- else {
- c_local `c_diparm' ```diparm'""
= c_local diparm ``lnalpha, exp label("alpha")""
- }

```

---

> \_\_\_\_\_ end \_get\_diparmopts.SpliceGlobals \_\_\_\_\_

```

- local diparm`k' ```diparm'""
= local diparm1 ``lnalpha, exp label("alpha")""
- local diparmopts ```diparmopts' : `diparm'""
= local diparmopts `` : lnalpha, exp label("alpha")""
- if `single' | ```:list retok after'"" == "" {
= if 1 | `""' == "" {
- local 0 `", `options'""
= local 0 `", search(off)""
- syntax [, diparm(string asis) * ]
- }
- else {
  local diparm ```after'""
  }
- }
- while ```diparm'"" != "" {
= while `""' != "" {
  local ++k
  local single 1
  local after
  capture _on_colon_parse `diparm'
  if !c(rc) {
    local single 0
    local diparm ```s(before)""
    local after ```s(after)""
  }
  capture

```

```

SpliceGlobals diparm , diparm(`diparm`) `syntaxonly' `goptions'
local diparm`k' ```diparm'''
local diparmopts ```diparmopts' : `diparm'''
if `single' | ```:list retok after''' == "" {
local 0 `", `options'''
syntax [, diparm(string asis) * ]
}
else {
local diparm ```after'''
}
}
- if ```soptions''' == "" {
= if ``soptions''' == "" {
syntax [, diparm(string asis) ]
}
- if (``execute''' == "") {
= if ("" == "") {
- sreturn clear
- sreturn local options ```options'''
= sreturn local options ``search(off)'''
- sreturn local diparm ```diparmopts'''
= sreturn local diparm `` : lnalpha, exp label("alpha")'''
- forval i = 1/`k' {
= forval i = 1/1 {
- sreturn local diparm`i' ```diparm`i'''
= sreturn local diparm1 ``lnalpha, exp label("alpha")'''
- }
- sreturn local k `k'
= sreturn local k 1
- exit

```

---

> \_\_\_\_\_ end \_get\_diparmopts \_\_\_\_\_

```

- forval k = 1/`s(k)' {
= forval k = 1/1 {
- local diparm`k' ```s(diparm`k)'''
= local diparm1 ``lnalpha, exp label("alpha")'''
- }
- local ndiparm `s(k)'
= local ndiparm 1
- local options ```s(options)'''
= local options ``search(off)'''
- if `interactive' & `:length local options' {
= if 0 & 11 {
local 0 , `options'
syntax [, NOTANOPTION]
}

```

```

- if `:length local nopreserve' | `:length local score' {
= if 10 | 0 {
- global ML_preserve no
- }
- else global ML_preserve yes
- gettoken evaltype spec : spec
- confirm name `evaltype'
= confirm name e2
- gettoken evaluator spec : spec
- global ML_evaltype : copy local evaltype
- if `:length local srssubpop' {
= if 0 {
  local 0 , srssubpop
  syntax [, NONOPTION]
  exit 198
}
- if `:length local subpop' {
= if 0 {
  local svy svy
}
- if `:length local svy' {
= if 0 {
  opts_exclusive "`vce' `svy'" vce
  if `:length local weight' {
    di as err "svy option not allowed with weights"
    exit 198
  }
  global ML_noadj : copy local svyadjust
}
- global ML_svy : copy local svy
- if _caller() >= 15 & c(userversion) >= 15 {
- local fparm ", freeparm"
- }
- local eqspec : copy local spec
- local ieq 0
- while `:length local eqspec' {
= while 127 {
- local ++ieq
- gettoken eq eqspec : eqspec, bind
- if bsubstr(trim("`eq'"),1,1) == "/" {
= if bsubstr(trim("(y_timing_max: y_timing_max=cb_authors_ext estima
> ted wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )"),1,1)
> == "/" {
  local eq = bsubstr(trim("`eq'"),2,.)
  local spec : substr local spec "/"`eq' " ("`eq'":`fparm')"' , wor
> d

```

```

    }
    - else {
    - gettoken p1 p2 : eq, parse("()")
    - local p2 = trim("`p2'")
    = local p2 = trim("`y_timing_max: y_timing_max=cb_authors_ext estima
> ted wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )'")
    - if bsubstr("`p2'",1,1) == "," {
    = if bsubstr("y_timing_max: y_timing_max=cb_authors_ext estimated wg
> _ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )",1,1) == ","
> {
    gettoken comma p2 : p2, parse(",")
    local p2 = trim("`p2'")
    }
    - if "`p1'`p2'" == "()" {
    = if "`(y_timing_max: y_timing_max=cb_authors_ext estimated wg_ndx_m
> ult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open,  )'" == "()" {
    local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
    }
    - }
    - }
    - while `:length local eqspec' {
    = while 9 {
    - local ++ieq
    - gettoken eq eqspec : eqspec, bind
    - if bsubstr(trim("`eq'"),1,1) == "/" {
    = if bsubstr(trim("/lnalpha"),1,1) == "/" {
    - local eq = bsubstr(trim("`eq'"),2,.)
    = local eq = bsubstr(trim("/lnalpha"),2,.)
    - local spec : subinstr local spec "/"`eq'" "`("`eq':`fparm')'", wor
> d
    = local spec : subinstr local spec "/lnalpha" "`("lnalpha":, freepar
> m)", word
    - }
    - else {
    gettoken p1 p2 : eq, parse("()")
    local p2 = trim("`p2'")
    if bsubstr("`p2'",1,1) == "," {
    gettoken comma p2 : p2, parse(",")
    local p2 = trim("`p2'")
    }
    if "`p1'`p2'" == "()" {
    local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
    }
    - }
    - while `:length local eqspec' {

```

```

= while 0 {
  local ++ieq
  gettoken eq eqspec : eqspec, bind
  if bsubstr(trim("`eq'"),1,1) == "/" {
    local eq = bsubstr(trim("`eq'"),2,.)
    local spec : subinstr local spec "/"`eq' " ("`eq'":`fparm')", wor
> d
  }
  else {
    gettoken p1 p2 : eq, parse("(")
    local p2 = trim("`p2'")
    if bsubstr("`p2'",1,1) == "," {
      gettoken comma p2 : p2, parse(",")
      local p2 = trim("`p2'")
    }
    if "`p1'`p2'" == "(" {
      local spec : subinstr local spec "`eq'" "(eq`ieq':)", word
    }
  }
}
- if `interactive' {
= if 0 {
  SetGlobalVar ML_sample : _MLtua
}
- else {
- global ML_sample : tempvar
- if ``moptobj' != "" {
= if `"' != "" {
  global ML_sample `moptobj'$ML_sample
}
- }
- if ``moptobj' != "" {
= if `"' != "" {
  cap drop $ML_sample
}
- `vv' mark $ML_sample `if' `in' [`weight'`exp'], `zeroweight'
= version 18: mark __000006 if __000000 [],
- tempname eqlist
- `eqlist' = _eqlist.new, eqopts(NOCONSTant FREEParam) eqargopts(OF
> Fset EXPosure) noneedvarlist numdepvars(0) needequal
  = __000007 = _eqlist.new, eqopts(NOCONSTant FREEParam) eqargopts(OF
> Fset EXPosure) noneedvarlist numdepvars(0) needequal
  -class {
  -instance:
    - array n_depvars
    - string needequal

```

```

-      string      noneedvarlist
-      string      wtypes
-      string      collinear
-      string      nocons
-      string      nocons_ignore
-      double      n_eq
-      string      gl_if
-      string      gl_in
-      string      gl_wgt
-      string      gl_options
-      _optlist     eq_opts_allowed
-  -class {
-  -instance:
-      array options = {}
-      array names   = {}
-  -}
-      array      eq_list
-      _optlist    mark_opts
-      array      mark_list
-      array      mark_name
-      _optlist    smark_opts
-      array      smark_list
-      array      smark_name
-      _optlist    rmcoll_opts
-      array      rmcoll_list
-      array      rmcoll_name
-      _optlist    rmdcoll_opts
-      array      rmdcoll_list
-      array      rmdcoll_name
-  -}

```

---

```

> ----- begin _eqlist.new -----
- version 9.2
- ._init

```

---

```

> ----- begin _eqlist._init -----
- version 9.2
- .n_eq = 0
- .gl_if = ""
- .gl_in = ""
- .gl_wgt = ""
- .gl_options = ""
- .eq_list = {}

```

---

```

> ----- end _eqlist._init -----
- ._setup `0'

```

```

= ._setup , eqopts(NOCONSTant FREEParam) eqargopts(OFFset EXPosure)
> noneedvarlist numdepvars(0) needequal

> ----- begin _eqlist._setup -----
- version 9.2
- .mark_list = {}
- .mark_name = {}
- .mark_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .smark_list = {}
- .smark_name = {}
- .smark_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .rmcoll_list = {}
- .rmcoll_name = {}
- .rmcoll_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- .rmdcoll_list = {}
- .rmdcoll_name = {}
- .rmdcoll_opts.reset
-----

> ----- begin _optlist.reset -----
- version 9
- .options = {}
- .names = {}
-----

> ----- end _optlist.reset -----
- syntax [, EQopts(namelist) EQARGopts(namelist) noNEEDVARLIST C

```

```

> OMMONopts(string asis) MARKopts(namelist) SMARKopts(namelist) RMCOLLopts(nam
> elist) RMDCOLLopts(namelist) NUMDEPvars(numlist integer >=0) NEEDEQUAL WTYPE
> S(string) IGNORENOCONS ]
    - .eq_opts_allowed.reset
    _____
> _____ begin _optlist.reset _____
    - version 9
    - .options = {}
    - .names = {}
    _____
> _____ end _optlist.reset _____
    - foreach opt of local eqopts {
    - .eq_opts_allowed.addopt `opt'
    = .eq_opts_allowed.addopt NOCONSTant
    _____
> _____ begin _optlist.addopt _____
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("NOCONSTant")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "NOCONSTant"
    - .names.Arrpush "`mname'"
    = .names.Arrpush "noconstant"
    _____
> _____ end _optlist.addopt _____
    - }
    - .eq_opts_allowed.addopt `opt'
    = .eq_opts_allowed.addopt FREEParm
    _____
> _____ begin _optlist.addopt _____
    - version 9
    - syntax name(name=optspec) [, PASSthru]
    - local mname = lower("`optspec'")
    = local mname = lower("FREEParm")
    - if "`passthru'" != "" {
    = if "" != "" {
      local optspec `optspec'(passthru)
    }
    - .options.Arrpush "`optspec'"
    = .options.Arrpush "FREEParm"
    - .names.Arrpush "`mname'"

```



```

= .names.Arrpush "freeparm"
> ----- end _optlist.addopt -----
- }
- foreach opt of local eqargopts {
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt OFFset, passthru
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OFFset")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec OFFset(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "OFFset(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "offset"
> ----- end _optlist.addopt -----
- }
- .eq_opts_allowed.addopt `opt', passthru
= .eq_opts_allowed.addopt EXPosure, passthru
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("EXPosure")
- if "`passthru'" != "" {
= if "passthru" != "" {
- local optspec `optspec'(passthru)
= local optspec EXPosure(passthru)
- }
- .options.Arrpush "`optspec'"
= .options.Arrpush "EXPosure(passthru)"
- .names.Arrpush "`mname'"
= .names.Arrpush "exposure"
> ----- end _optlist.addopt -----
- }
- local i 0

```

```

- foreach opt of local markopts {
  .mark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local smarkopts {
  .smark_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmcollopts {
  .rmcoll_opts.addopt `opt', passthru
}
- local i 0
- foreach opt of local rmdcollopts {
  .rmdcoll_opts.addopt `opt', passthru
}
- .n_depvars.Arrdropall
- if "`numdepvars'" != "" {
= if "0" != "" {
- foreach ndep of local numdepvars {
- .n_depvars.Arrpush `ndep'
= .n_depvars.Arrpush 0
- }
- }
- else .n_depvars.Arrpush 1
- .needequal = "`needequal'"
= .needequal = "needequal"
- .wtypes = "`wtypes'"
= .wtypes = ""
- .noneedvarlist = "`needvarlist'"
= .noneedvarlist = "noneedvarlist"
- .nocons_ignore = "`ignorenocons'"
= .nocons_ignore = ""

> _____ end _eqlist._setup _____
> _____ end _eqlist.new _____
- .`eqlist'.parse `spec'
= .__000007.parse (y_timing_max: y_timing_max=cb_authors_ext estima
> ted wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open, ) ("lnalp
> ha":, freeparm)

> _____ begin _eqlist.parse _____
- version 9.2
- .__init

> _____ begin _eqlist._init _____

```

```

- version 9.2
- .n_eq = 0
- .gl_if = ""
- .gl_in = ""
- .gl_wgt = ""
- .gl_options = ""
- .eq_list = {}

> ----- end _eqlist._init -----
- local MARKOPTLIST ```.mark_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```.optlist'''
= sreturn local optlist `'''
- class exit ```.optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----
= local MARKOPTLIST `'''
- local SMARKOPTLIST ```.smark_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```.optlist'''
= sreturn local optlist `'''
- class exit ```.optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----
= local SMARKOPTLIST `'''
- local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {

```

```

= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist `'''
- class exit ```optlist'''
= class exit `'''

> _____ end _optlist.dumpoptions _____

= local RMCOLLOPTLIST `'''
- local RMDCOLLOPTLIST ```.rmdcoll_opts.dumpoptions'''

> _____ begin _optlist.dumpoptions _____

- version 9
- forval i = 1/`.options.arrnels' {
= forval i = 1/0 {
  local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist `'''
- class exit ```optlist'''
= class exit `'''

> _____ end _optlist.dumpoptions _____

= local RMDCOLLOPTLIST `'''
- _parse expand EQ GL : 0, gweight common( `MARKOPTLIST' `SMARKOPT
> LIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
= _parse expand EQ GL : 0, gweight common( COLLlinear )
- if `EQ_n' == 0 & ``.noneedvarlist' == "" {
= if 2 == 0 & "noneedvarlist" == "" {
  syntax varlist
}
- local rebind = `EQ_n' == 1
= local rebind = 2 == 1
- if `rebind' {
= if 0 {
  tokenize ```0''', parse(",()[]")
  local rebind = ```1''' != "(" & ```2''' == "("
}
- if `rebind' {
= if 0 {
  while `:length local 0' {
    gettoken tok 0 : 0, parse(" ,") bind match(par)
    if "`par'" == "(" {

```

```

gettoken nxt : 0
local nxt = substr("`nxt'",1,1)
if inlist("`nxt'", "#", "|") {
  local ZERO ``ZERO' (`tok')''
}
else {
  local ZERO ``(`ZERO') (`tok') `0'''
  local 0
}
else if inlist("`tok'", "if", "in", ",",) {
  local ZERO ``(`ZERO') `tok' `0'''
  local 0
}
else if inlist(substr("`tok'",1,1), "#", "|") {
  local ZERO ``ZERO`tok'''
}
else {
  local ZERO ``ZERO' `tok'''
}
}
_parse expand EQ GL : ZERO, gweight common( `MARKOPTLIST' `SMARK
> OPTLIST' `RMCOLLOPTLIST' `RMDCOLLOPTLIST' COLLlinear )
}
- forval i = 1/`EQ_n' {
= forval i = 1/2 {
- if `i' <= `.n_depvars.arnels' {
= if 1 <= 1 {
- local ndep = `.n_depvars[`i']'
= local ndep = 0
- }
- .eq_list[`i'] = .eqspec.new, numdepvars(`ndep') `.needequal' `.
> noneedvarlist' wtypes(`.wtypes') `.nocons_ignore'
= .eq_list[1] = .eqspec.new, numdepvars(0) needequal noneedvarlis
> t wtypes()
-class {
-instance:
- double n_depvars
- double needequal
- double needvarlist
- string wtypes
- string name
- string depvars
- double hasequal
- string indepvars
- string other_markvars

```

```

-      double  nocons
-      double  nocons_ignore
-      string  offset
-      double  exposure
-      double  freeparm
-      string  if
-      string  in
-      string  wgt
-      string  options
-}

> ----- begin _eqspec.new -----
- version 9.2
- syntax [, NUMDEPvars(numlist integer >=0 max=1) NEEDEQUAL noNE
> EDVARLIST WTYPES(string) IGNORENOCONS ]
- .n_depvars = 1
- if `:list sizeof numdepvars' {
= if 1 {
- .n_depvars = `numdepvars'
= .n_depvars = 0
- }
- .needequal = ("`needequal'" != "")
= .needequal = ("needequal" != "")
- .needvarlist = ("`needvarlist'" == "")
= .needvarlist = ("noneedvarlist" == "")
- .nocons_ignore = ("`ignorenocons'" != "")
= .nocons_ignore = ("" != "")
- .wtypes = "`wtypes'"
= .wtypes = ""

> ----- end _eqspec.new -----
- .eq_list['i'].parse `.eq_opts_allowed.objkey':`EQ`i''
= .eq_list[1].parse __key(600003ed3840):y_timing_max: y_timing_max
> =cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo l
> n_neq open

> ----- begin _eqspec.parse -----
- version 9.2
- _on_colon_parse `0'
= _on_colon_parse __key(600003ed3840):y_timing_max: y_timing_max
> =cb_authors_ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo l
> n_neq open

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""

```

```

- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before "`('before)'"
}
- if ``"before"' == ":" {
= if ``__key(600003ed3840)'" == ":" {
  sreturn local after ``"after"'
  exit
}
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``"' != ":" & ``:y_timing_max: y_timing_max=cb_authors
> _ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open"
> ' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quo
> tes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if ``"COLON"' != ":" {
= else if ``:"' != ":" {
  local before ``"before' `COLON"'
  local COLON
}
- }
- while ``"COLON"' != ":" & ``"after"' != "" {
= while ``:"' != ":" & ``y_timing_max: y_timing_max=cb_authors
> _ext estimated wg_ndx_mult stky_pr_rottemberg wlth stky_pr_calvo ln_neq open"
> ' != "" {
  gettoken COLON after : after, parse(":") bind match(par) quo
> tes
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if ``"COLON"' != ":" {
    local before ``"before' `COLON"'
    local COLON
  }
}
- if ``"COLON"' != ":" {
= if ``:"' != ":" {
  di as err "" found where ':' expected"

```

```

        exit 198
    }
    - sreturn local before ``before'''
    = sreturn local before ``__key(600003ed3840)'''
    - sreturn local after ``after'''
    = sreturn local after ``y_timing_max: y_timing_max=cb_authors_
> ext estimated wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open'''
    _____
> _____ end _on_colon_parse _____
    - local ol `s(before)'
    = local ol __key(600003ed3840)
    - if `:list sizeof ol' {
    = if 1 {
    - local OPTLIST ```.ol'.dumpoptions'''
    _____
> _____ begin _optlist.dumpoptions _____
    - version 9
    - forval i = 1/`.options.arnels' {
    = forval i = 1/4 {
    - local optlist `optlist' `.options['i']'
    = local optlist NOCONSTant
    - }
    - local optlist `optlist' `.options['i']'
    = local optlist NOCONSTant FREEParam
    - }
    - local optlist `optlist' `.options['i']'
    = local optlist NOCONSTant FREEParam OFFSET(passthru)
    - }
    - local optlist `optlist' `.options['i']'
    = local optlist NOCONSTant FREEParam OFFSET(passthru) EXPosure(
> passthru)
    - }
    - local optlist : list uniq optlist
    - sreturn local optlist ``optlist'''
    = sreturn local optlist ``NOCONSTant FREEParam OFFSET(passthru)
> EXPosure(passthru)'''
    - class exit ``optlist'''
    = class exit ``NOCONSTant FREEParam OFFSET(passthru) EXPosure(p
> assthru)'''
    _____
> _____ end _optlist.dumpoptions _____
    = local OPTLIST ``NOCONSTant FREEParam OFFSET(passthru) EXPosure(
> passthru)'''
    - local OPTNAMES ```.ol'.dumpnames'''
    _____
> _____ begin _optlist.dumpnames _____

```



```

- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/4 {
- local namelist `namelist' `.names[`i']'
= local namelist noconstant
- }
- local namelist `namelist' `.names[`i']'
= local namelist noconstant freeparm
- }
- local namelist `namelist' `.names[`i']'
= local namelist noconstant freeparm offset
- }
- local namelist `namelist' `.names[`i']'
= local namelist noconstant freeparm offset exposure
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist ``noconstant freeparm offset exposure'''
> ""

- class exit ``namelist'''
= class exit ``noconstant freeparm offset exposure'''


---


> 

---

 end _optlist.dumpnames 

---


= local OPTNAMES ``noconstant freeparm offset exposure'''
- local NOCONS constant noconstant
- if ``:list NOCONS & OPTNAMES''' == "" {
= if ``noconstant''' == "" {
local NOCONS noCONSTant
}
- else local NOCONS
- }
- local 0 ``s(after)'''
= local 0 ``y_timing_max: y_timing_max=cb_authors_ext estimated
> wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open'''
- if ``.wtypes''' != "" {
= if ``''' != "" {
local wgt "[`.wtypes']"
}
- syntax [anything(equalok)] [if] [in] `wgt' [, `OPTLIST' `NOCON
> S' *]
= syntax [anything(equalok)] [if] [in] [, NOCONSTant FREEParam 0
> FFset(passthru) EXPosure(passthru) *]
- local HOLDIF ``if'''
= local HOLDIF ``'''
- local HOLDIN ``in'''
= local HOLDIN ``'''

```

```

- if "`wgt'" != "" & "`weight'" != "" {
= if "" != "" & "" != "" {
  local HOLDWGT "`['weight'`exp']"
}
- gettoken NAME REST : anything, parse(":")
- if trim("`NAME'") == ":" {
= if trim("y_timing_max") == ":" {
  di as err "nothing found where equation name expected"
  exit 198
}
- if `:list sizeof REST' {
= if 9 {
- if strpos("`NAME'", ":") {
= if strpos("y_timing_max", ":") {
  capture _msparse `NAME'
  if c(rc) {
    di as err "{bf:`NAME'} invalid equation specification"
    exit 198
  }
}
- else {
- . _confirm_eqname `NAME'
= . _confirm_eqname y_timing_max
}
}

> ----- begin _eqspec._confirm_eqname -----
- version 9.2
- capture confirm number `0'
= capture confirm number y_timing_max
- if (! c(rc)) exit
- local hasdot = strpos("`0'", ".")
= local hasdot = strpos("y_timing_max", ".")
- if `hasdot' {
= if 0 {
  syntax varname(ts)
  exit
}
- capture syntax name(id="equation name")
- if (! c(rc)) exit
}

> ----- end _eqspec._confirm_eqname -----
- }
- local NAME : list retok NAME
- gettoken COLON anything : REST, parse(":")
- }
- else {
  local NAME

```

```

    }
    - gettoken DEPVARs REST : anything, parse("=") bind
    - if "`DEPVARs'" == "=" {
= if "y_timing_max" == "=" {
    local REST : copy local anything
    local DEPVARs
    }
    - if (:length local DEPVARs' == 0 & `.n_depvars') {
= if (12 == 0 & 0) {
    di as err "nothing found where depvar expected"
    exit 198
    }
    - if `"'REST'"' == "" {
= if `"'=cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg w
> lth stky_pr_calvo ln_neq open"' == "" {
    if 0`.needequal' {
    if `.needvarlist' | (:length local DEPVARs' != 0 & `.n_depvar
> s') {
    di as err "equal sign required to separate depvars from indepv
> ars"

    exit 198
    }
    }
    local DEPVARs
    local n_deps : list sizeof anything
    if `n_deps' != 0 | `.needvarlist' {
    if `n_deps' < `.n_depvars' {
    error 102
    }
    forval j = 1/`.n_depvars' {
    gettoken YY anything : anything
    local DEPVARs `DEPVARs' `YY'
    }
    local INDEPVARs : list retok anything
    }
    loca hasequal 0
    }
    - else {
    - gettoken EQ INDEPVARs : REST, parse("=") bind
    - local hasequal 1
    - }
    - if `:list sizeof DEPVARs' {
= if 1 {
    - _fv_check_depvar `DEPVARs'
= _fv_check_depvar y_timing_max
    - tsunab DEPVARs : `DEPVARs'

```

```

= tsunab DEPVARS : y_timing_max
> ----- begin tsunab -----
- version 6
- gettoken user 0: 0, parse(" :")
- gettoken colon 0: 0, parse(" :")
- if ``colon`` != ":" { error 198 }
= if ``:`` != ":" { error 198 }
- syntax [varlist(ts default=empty)] [, MIN(integer 1) MAX(int
> eger 32000) NAME(string)]
- c_local `user' `varlist'
= c_local DEPVARS y_timing_max
- local n : word count `varlist'
= local n : word count y_timing_max
- if `n'>=`min' & `n'<=`max' { exit }
= if 1>=1 & 1<=32000 { exit }
> ----- end tsunab -----
- }
- if `:list sizeof INDEPVARS' {
= if 8 {
- fvunab INDEPVARS : `INDEPVARS'
= fvunab INDEPVARS : cb_authors_ext estimated wg_ndx_mult stky_p
> r_rotemberg wlth stky_pr_calvo ln_neq open
> ----- begin fvunab -----
- version 11
- _on_colon_parse `0'
= _on_colon_parse INDEPVARS : cb_authors_ext estimated wg_ndx_
> mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open
> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quot
> es
- if ``par`` != "" {
= if "" != "" {
local before ``(`before')``
}
- if ``before`` == ":" {
= if ``INDEPVARS `` == ":" {
sreturn local after ``after``
exit
}

```

```

- while `""COLON"" != ":" & `""after"" != "" {
= while `"" != ":" & `": cb_authors_ext estimated wg_ndx_mu
> lt stky_pr_rotemberg wlth stky_pr_calvo ln_neq open"" != "" {
- gettoken COLON after : after, parse(":") bind match(par) q
> uotes

- if ""par"" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if `""COLON"" != ":" {
= else if `": "" != ":" {
  local before `""before' `COLON""
  local COLON
}
- }
- while `""COLON"" != ":" & `""after"" != "" {
= while `": "" != ":" & ` cb_authors_ext estimated wg_ndx_mu
> lt stky_pr_rotemberg wlth stky_pr_calvo ln_neq open"" != "" {
  gettoken COLON after : after, parse(":") bind match(par) q
> uotes

  if ""par"" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if `""COLON"" != ":" {
    local before `""before' `COLON""
    local COLON
  }
}
- if `""COLON"" != ":" {
= if `": "" != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before `""before""
= sreturn local before `INDEPVARs ""
- sreturn local after `""after""
= sreturn local after ` cb_authors_ext estimated wg_ndx_mu
> t stky_pr_rotemberg wlth stky_pr_calvo ln_neq open""

> ----- end _on_colon_parse -----
- local 0 `""s(before)""
= local 0 `INDEPVARs ""
- syntax name(id="macro name" name=user)
- local 0 `""s(after)""

```

```

= local 0 `` cb_authors_ext estimated wg_ndx_mult stky_pr_rote
> mberg wlth stky_pr_calvo ln_neq open''
- syntax [varlist(ts fv default=none)] [, MIN(integer 1) MAX(i
> nteger 32000) NAME(string) ]
- c_local `user' `varlist'
= c_local INDEPVARs cb_authors_ext estimated wg_ndx_mult stky_
> pr_rottemberg wlth stky_pr_calvo ln_neq open
- local n : list sizeof varlist
- if (`min' <= `n' & `n' <= `max') exit
= if (1 <= 8 & 8 <= 32000) exit

> ----- end fvunab -----

- }
- if inlist("noconstant", "`noconstant'", "`constant'") {
= if inlist("noconstant", "", "") {
if "`INDEPVARs'" == "" & !`.nocons_ignore' {
di as err "independent variables are required when option noco
> nstant is specified"
exit 102
}
if "`NOCONS'" != "" {
local options `options' noconstant
}
local NOCONS 1
}
- else local NOCONS 0
- if ``offset'`exposure'`` != "" {
= if `""' != "" {
._check_offset, `offset' `exposure'
local OFFSET ``s(offset)''
local EXPOSURE = (`exposure' != "")
local OTHER `s(offset)'
}
- else local EXPOSURE 0
- local FREEPARM = ``freeparm'`` != ""
= local FREEPARM = `""' != ""
- local OPTIONS
- local SPACE
- foreach OPTNAME of local OPTNAMES {
- if ``OPTNAME'`` != "" {
= if `""' != "" {
local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
local SPACE " "
}
- }
- if ``OPTNAME'`` != "" {

```

```

= if `""' != "" {
  local OPTIONS ``OPTIONS``SPACE``OPTNAME''''
  local SPACE " "
}
- }
- if `""`OPTNAME'''' != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS``SPACE``OPTNAME''''
  local SPACE " "
}
- }
- if `""`OPTNAME'''' != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS``SPACE``OPTNAME''''
  local SPACE " "
}
- }
- if ``options'' != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS``SPACE``options''
}
- .name = ``NAME''
= .name = `y_timing_max'
- .depvars = ``DEPVARs''
= .depvars = `y_timing_max'
- .hasequal = `hasequal'
= .hasequal = 1
- .indepvars = ``:list retok INDEPVARs''
= .indepvars = `cb_authors_ext estimated wg_ndx_mult stky_pr_ro
> temberg wlth stky_pr_calvo ln_neq open'
- .other_markvars = ``:list retok OTHER''
= .other_markvars = `""
- .nocons = `NOCONS'
= .nocons = 0
- .if = ``HOLDIF''
= .if = `""
- .in = ``HOLDIN''
= .in = `""
- .wgt = ``HOLDWGT''
= .wgt = `""
- .offset = ``OFFSET''
= .offset = `""
- .exposure = `EXPOSURE'
= .exposure = 0
- .freeparm = `FREEPARM'
= .freeparm = 0

```

```

- .options = ```OPTIONS``
= .options = ````

```

---

```

> ----- end _eqspec.parse -----
- }
- if `i' <= `.n_depvars.arnels' {
= if 2 <= 1 {
  local ndep = `.n_depvars[`i']'
  }
- .eq_list[`i'] = ._eqspec.new, numdepvars(`ndep') `.needequal' `.
> noneedvarlist' wtypes(`.wtypes') `.nocons_ignore'
= .eq_list[2] = ._eqspec.new, numdepvars(0) needequal noneedvarlis
> t wtypes()

```

---

```

> ----- begin _eqspec.new -----
- version 9.2
- syntax [, NUMDEPvars(numlist integer >=0 max=1) NEEDEQUAL noNE
> EDVARLIST WTYPES(string) IGNORENOCONS ]
- .n_depvars = 1
- if `:list sizeof numdepvars' {
= if 1 {
- .n_depvars = `numdepvars'
= .n_depvars = 0
- }
- .needequal = ("`needequal'" != "")
= .needequal = ("needequal" != "")
- .needvarlist = ("`needvarlist'" == "")
= .needvarlist = ("noneedvarlist" == "")
- .nocons_ignore = ("`ignorenocons'" != "")
= .nocons_ignore = ("" != "")
- .wtypes = "`wtypes'"
= .wtypes = ""

```

---

```

> ----- end _eqspec.new -----
- .eq_list[`i'].parse `.eq_opts_allowed.objkey':`EQ_`i''
= .eq_list[2].parse __key(600003ed3840):"lnalpha":, freeparm

```

---

```

> ----- begin _eqspec.parse -----
- version 9.2
- _on_colon_parse `0'
= _on_colon_parse __key(600003ed3840):"lnalpha":, freeparm

```

---

```

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""

```



```

- gettoken before after : 0, parse(":") bind match(par) quotes
- if ``par`` != "" {
= if "" != "" {
  local before ``(`before')``
  }
- if ````before`` == ":" {
= if ``__key(600003ed3840)`` == ":" {
  sreturn local after ``after``
  exit
  }
- while ``COLON`` != ":" & ``after`` != "" {
= while ```` != ":" & ``:``lnalpha":, freeparm`` != "" {
- gettoken COLON after : after, parse(":") bind match(par) quo

```

> tes

```

- if ``par`` != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
  }
- else if ``COLON`` != ":" {
= else if ``:`` != ":" {
  local before ``before' `COLON``
  local COLON
  }
- }
- while ``COLON`` != ":" & ``after`` != "" {
= while ``:`` != ":" & ``lnalpha":, freeparm`` != "" {
  gettoken COLON after : after, parse(":") bind match(par) quo

```

> tes

```

  if ``par`` != "" {
    local before `before' (`COLON')
    local COLON
    }
  else if ``COLON`` != ":" {
    local before ``before' `COLON``
    local COLON
    }
  }
- if ``COLON`` != ":" {
= if ``:`` != ":" {
  di as err "" found where ':' expected"
  exit 198
  }
- sreturn local before ``before``
= sreturn local before ``__key(600003ed3840)``
- sreturn local after ``after``

```

```

= sreturn local after `""lnalpha":, freeparm""


---


> 

---

 end _on_colon_parse 

---


- local ol `s(before)'
= local ol __key(600003ed3840)
- if `:list sizeof ol' {
= if 1 {
- local OPTLIST `""`.ol'.dumpoptions""


---


> 

---

 begin _optlist.dumpoptions 

---


- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/4 {
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm OFFset(passthru)
- }
- local optlist `optlist' `.options['i']'
= local optlist NOCONSTant FREEParm OFFset(passthru) EXPosure(
> passthru)
- }
- local optlist : list uniq optlist
- sreturn local optlist `""optlist""
= sreturn local optlist `""NOCONSTant FREEParm OFFset(passthru)
> EXPosure(passthru)""
- class exit `""optlist""
= class exit `""NOCONSTant FREEParm OFFset(passthru) EXPosure(p
> assthru)""


---


> 

---

 end _optlist.dumpoptions 

---


= local OPTLIST `""NOCONSTant FREEParm OFFset(passthru) EXPosure(
> passthru)""
- local OPTNAMES `""`.ol'.dumpnames""


---


> 

---

 begin _optlist.dumpnames 

---


- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/4 {
- local namelist `namelist' `.names['i']'
= local namelist noconstant
- }

```

```

- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset
- }
- local namelist `namelist' `.names['i']'
= local namelist noconstant freeparm offset exposure
- }
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist ``noconstant freeparm offset exposure'

> ""

- class exit ``namelist'''
= class exit ``noconstant freeparm offset exposure'''



---


> ----- end _optlist.dumpnames -----

= local OPTNAMES ``noconstant freeparm offset exposure'''
- local NOCONS constant noconstant
- if ``:list NOCONS & OPTNAMES''' == "" {
= if ``noconstant''' == "" {
  local NOCONS noCONSTant
}
- else local NOCONS
- }
- local 0 ``s(after)'''
= local 0 ``lnalpha":, freeparm'''
- if ``.wtypes''' != "" {
= if `''' != "" {
  local wgt "[.wtypes']"
}
- syntax [anything(equalok)] [if] [in] `wgt' [, `OPTLIST' `NOCON

> S' *]

= syntax [anything(equalok)] [if] [in] [, NOCONSTant FREEParam 0
> FFset(passthru) EXPosure(passthru) *]
- local HOLDIF ``if'''
= local HOLDIF `'''
- local HOLDIN ``in'''
= local HOLDIN `'''
- if ``wgt''' != "" & ``weight''' != "" {
= if "" != "" & "" != "" {
  local HOLDWGT ``[weight'exp]'''
}
- gettoken NAME REST : anything, parse(":")
- if trim("`NAME'") == ":" {
= if trim("lnalpha") == ":" {

```

```

        di as err "nothing found where equation name expected"
        exit 198
    }
- if `:list sizeof REST' {
= if 1 {
- if strpos(`"NAME"', ":") {
= if strpos(`"lnalpha"', ":") {
    capture _msparse `NAME'
    if c(rc) {
        di as err "{bf:`NAME'} invalid equation specification"
        exit 198
    }
}
- else {
- . _confirm_eqname `NAME'
= . _confirm_eqname lnalpha
}

> ----- begin _eqspec._confirm_eqname -----
- version 9.2
- capture confirm number `0'
= capture confirm number lnalpha
- if (! c(rc)) exit
- local hasdot = strpos(`"0"', ".")
= local hasdot = strpos(`"lnalpha"', ".")
- if `hasdot' {
= if 0 {
    syntax varname(ts)
    exit
}
- capture syntax name(id="equation name")
- if (! c(rc)) exit
}

> ----- end _eqspec._confirm_eqname -----
- }
- local NAME : list retok NAME
- gettoken COLON anything : REST, parse(":")
- }
- else {
    local NAME
}
- gettoken DEPVARs REST : anything, parse("=") bind
- if "`DEPVARs'" == "=" {
= if "" == "=" {
    local REST : copy local anything
    local DEPVARs
}
}

```

```

- if (`:length local DEPVARs' == 0 & `.n_depvars') {
= if (0 == 0 & 0) {
    di as err "nothing found where depvar expected"
    exit 198
  }
- if ``REST'' == "" {
= if `'' == "" {
- if 0`.needequal' {
= if 01 {
- if `.needvarlist' | (`:length local DEPVARs' != 0 & `.n_depvar
> s') {
= if 0 | (0 != 0 & 0) {
    di as err "equal sign required to separate depvars from indepv
> ars"
    exit 198
  }
- }
- local DEPVARs
- local n_deps : list sizeof anything
- if `n_deps' != 0 | `.needvarlist' {
= if 0 != 0 | 0 {
    if `n_deps' < `.n_depvars' {
      error 102
    }
    forval j = 1/`.n_depvars' {
      gettoken YY anything : anything
      local DEPVARs `DEPVARs' `YY'
    }
    local INDEPVARs : list retok anything
  }
- local hasequal 0
- }
- else {
    gettoken EQ INDEPVARs : REST, parse("=") bind
    local hasequal 1
  }
- if `:list sizeof DEPVARs' {
= if 0 {
    _fv_check_depvar `DEPVARs'
    tsunab DEPVARs : `DEPVARs'
  }
- if `:list sizeof INDEPVARs' {
= if 0 {
    fvunab INDEPVARs : `INDEPVARs'
  }
- if inlist("noconstant", "`noconstant'", "`constant'") {

```

```

= if inlist("noconstant", "", "") {
  if "`INDEPVARs'" == "" & !`.nocons_ignore' {
    di as err "independent variables are required when option noco
> nstant is specified"
    exit 102
  }
  if "`NOCONS'" != "" {
    local options `options' noconstant
  }
  local NOCONS 1
}
- else local NOCONS 0
- if ``offset'`exposure'`` != "" {
= if `""' != "" {
  . _check_offset, `offset' `exposure'
  local OFFSET ``s(offset)''
  local EXPOSURE = (`exposure' != "")
  local OTHER `s(offset)'
}
- else local EXPOSURE 0
- local FREEARM = ``freeparm'`` != ""
= local FREEARM = `freeparm' != ""
- local OPTIONS
- local SPACE
- foreach OPTNAME of local OPTNAMES {
- if ``OPTNAME'`` != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
  local SPACE " "
}
- }
- if ``OPTNAME'`` != "" {
= if `freeparm' != "" {
- local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
= local OPTIONS `freeparm'
- local SPACE " "
- }
- }
- if ``OPTNAME'`` != "" {
= if `""' != "" {
  local OPTIONS ``OPTIONS'`SPACE'`OPTNAME'``
  local SPACE " "
}
- }
- if ``OPTNAME'`` != "" {
= if `""' != "" {

```

```

        local OPTIONS ```OPTIONS``SPACE``OPTNAME````
        local SPACE " "
    }
- }
- if ```options```` != "" {
= if ```` != "" {
    local OPTIONS ```OPTIONS``SPACE``options````
    }
- .name = ```NAME````
= .name = ``lnalpha``
- .depvars = ```DEPVARs````
= .depvars = ````
- .hasequal = `hasequal`
= .hasequal = 0
- .indepvars = ```:list retok INDEPVARs````
= .indepvars = ````
- .other_markvars = ```:list retok OTHER````
= .other_markvars = ````
- .nocons = `NOCONS`
= .nocons = 0
- .if = ```HOLDIF````
= .if = ````
- .in = ```HOLDIN````
= .in = ````
- .wgt = ```HOLDWGT````
= .wgt = ````
- .offset = ```OFFSET````
= .offset = ````
- .exposure = `EXPOSURE`
= .exposure = 0
- .freeparm = `FREEPARM`
= .freeparm = 1
- .options = ```OPTIONS````
= .options = ``freeparm``

> _____ end _eqspec.parse _____
- }
- local MARKOPTLIST ```.mark_opts.dumpoptions````

> _____ begin _optlist.dumpoptions _____
- version 9
- forval i = 1/`.options.arnels` {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']`
    }
- local optlist : list uniq optlist

```

```

- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

```

---

```

> ----- end _optlist.dumpoptions -----
= local MARKOPTLIST `'''
- local MARKOPTNAMES ```.mark_opts.dumpnames'''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/0 {
    local namelist `namelist' `names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

```

---

```

> ----- end _optlist.dumpnames -----
= local MARKOPTNAMES `'''
- local OPTLIST1 : substr local MARKOPTLIST "(passthru)" "(string
> asis)", all
- local SMARKOPTLIST ```.smark_opts.dumpoptions'''

```

---

```

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

```

---

```

> ----- end _optlist.dumpoptions -----
= local SMARKOPTLIST `'''
- local SMARKOPTNAMES ```.smark_opts.dumpnames'''

```

---

```

> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {

```



```

= forval i = 1/0 {
    local namelist `namelist' `.names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

> ----- end _optlist.dumpnames -----

= local SMARKOPTNAMES `'''
- local OPTLIST1s : substr local SMARKOPTLIST "(passthru)" "(str
> ing asis)", all
- local RMCOLLOPTLIST ```.rmcoll_opts.dumpoptions'''

> ----- begin _optlist.dumpoptions -----

- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/0 {
    local optlist `optlist' `.options['i']'
}
- local optlist : list uniq optlist
- sreturn local optlist ``optlist'''
= sreturn local optlist `'''
- class exit ``optlist'''
= class exit `'''

> ----- end _optlist.dumpoptions -----

= local RMCOLLOPTLIST `'''
- local RMCOLLOPTNAMES ```.rmcoll_opts.dumpnames'''

> ----- begin _optlist.dumpnames -----

- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/0 {
    local namelist `namelist' `.names['i']'
}
- local namelist : list uniq namelist
- sreturn local namelist ``namelist'''
= sreturn local namelist `'''
- class exit ``namelist'''
= class exit `'''

> ----- end _optlist.dumpnames -----

= local RMCOLLOPTNAMES `'''
- local OPTLIST2 : substr local RMCOLLOPTLIST "(passthru)" "(str

```

```

> ring asis)", all
    - local RMDCOLLOPTLIST ```.rmdcoll_opts.dumpoptions'''
> ----- begin _optlist.dumpoptions -----
    - version 9
    - forval i = 1/`.options.arnels' {
    = forval i = 1/0 {
        local optlist `optlist' `.options['i']'
    }
    - local optlist : list uniq optlist
    - sreturn local optlist ```optlist'''
    = sreturn local optlist `'''
    - class exit ```optlist'''
    = class exit `'''
> ----- end _optlist.dumpoptions -----
    = local RMDCOLLOPTLIST `'''
    - local RMDCOLLOPTNAMES ```.rmdcoll_opts.dumpnames'''
> ----- begin _optlist.dumpnames -----
    - version 9
    - forval i = 1/`.names.arnels' {
    = forval i = 1/0 {
        local namelist `namelist' `.names['i']'
    }
    - local namelist : list uniq namelist
    - sreturn local namelist ```namelist'''
    = sreturn local namelist `'''
    - class exit ```namelist'''
    = class exit `'''
> ----- end _optlist.dumpnames -----
    = local RMDCOLLOPTNAMES `'''
    - local OPTLIST3 : substr local RMDCOLLOPTLIST "(passthru)" "(st
> ring asis)", all
    - local 0 `", `GL_op'''
    = local 0 `", ""
    - syntax [, `OPTLIST1' `OPTLIST1s' `OPTLIST2' `OPTLIST3' COLLlinear
> noCONSTant * ]
    = syntax [, COLLlinear noCONSTant * ]
    - local GL_op ```options'''
    = local GL_op `'''
    - if ```collinear''' != "" {
    = if `''' != "" {
        local GL_op ```GL_op' `collinear'''
        .collinear = ``collinear'''

```

```

}
- if ``constant'' != "" {
= if ``'' != "" {
    local GL_op ``GL_op' `constant''
    .nocons = ``constant''
}
- local I 0
- foreach OPT of local MARKOPTNAMES {
    if `:list sizeof `OPT'' {
        .mark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .mark_list[`I'].parse "" : ``OPT''
        .mark_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local SMARKOPTNAMES {
    if `:list sizeof `OPT'' {
        .smark_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .smark_list[`I'].parse "" : ``OPT''
        .smark_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local RMCOLLOPTNAMES {
    if `:list sizeof `OPT'' {
        .rmcoll_list[`++I'] = .eqspec.new, numdep(0) wtypes(`.wtypes')
        .rmcoll_list[`I'].parse "" : ``OPT''
        .rmcoll_name.Arrpush ``OPT''
    }
}
- local I 0
- foreach OPT of local RMDCOLLOPTNAMES {
    if `:list sizeof `OPT'' {
        .rmdcoll_list[`++I'] = .eqspec.new, wtypes(`.wtypes')
        .rmdcoll_list[`I'].parse "" : ``OPT''
        .rmdcoll_name.Arrpush ``OPT''
    }
}
- .n_eq = `EQ_n'
= .n_eq = 2
- .gl_if = ``GL_if''
= .gl_if = ``''
- .gl_in = ``GL_in''
= .gl_in = ``''
- .gl_wgt = ``GL_wt''
= .gl_wgt = ``''

```

```

- .gl_options = ``:list retok GL_op'''
= .gl_options = `'''

> ----- end _eqlist.parse -----
- if ``subpop''' != "" & ``missing''' == "" {
= if "" != "" & "missing" == "" {
  if `interactive' {
    SetGlobalVar ML_samp : _MLtu
  }
  else {
    global ML_samp : tempvar
    if ``moptobj''' != "" {
      global ML_samp `moptobj'$ML_samp
    }
  }
  tempvar markuse
  _svy_setup $ML_sample $ML_samp, svy subpop(`subpop')
  quietly gen byte `markuse' = $ML_samp
}
- else global ML_samp $ML_sample
= else global ML_samp __000006
- if ``collinear''' == "" {
= if "collinear" == "" {
  .`eqlist'.rmcoll $ML_samp, replace `missing'
}
- else if ``missing''' == "" {
= else if "missing" == "" {
  .`eqlist'.markout $ML_samp, replace
}
- if "$ML_samp" != "$ML_sample" {
= if "__000006" != "__000006" {
  `vv' quietly replace $ML_sample = 0 if `markuse' & !$ML_samp
  drop `markuse'
}
- local eq_n = `.`eqlist'.eq count'

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item''' == "count" {
= if "count" == "count" {
  - sreturn local value ``.n_eq'''
= sreturn local value ``2'''
  - class exit ``s(value)'''
= class exit ``2'''

```

```

> ----- end _eqlist.eq -----
    = local eq_n = 2
    - global ML_n : copy local eq_n
    - global ML_dim = ``.eqlist'.dim'
    -----

> ----- begin _eqlist.dim -----
    - version 9.2
    - syntax
    - local neq = ``.eq count'
    -----

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item'" == "count" {
    = if "count" == "count" {
    - sreturn local value ``.n_eq'"
    = sreturn local value ``2'"
    - class exit ``s(value)'"
    = class exit ``2'"
    -----

> ----- end _eqlist.eq -----
    = local neq = 2
    - forval i = 1/`neq' {
    = forval i = 1/2 {
    - local xvarsi ``.eq indepvars `i'"
    -----

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item'" == "count" {
    = if "indepvars" == "count" {
    sreturn local value ``.n_eq'"
    class exit ``s(value)'"
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item indepvars
    -----

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, indepvars
    -----

> ----- begin _eqspec._check_eq_item_name -----

```

```

- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
    local other other_markvars
    }
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
    error 198
    }
- c_local `c_item' `item'
= c_local item indepvars

> _____ end _eqspec._check_eq_item_name _____
- local value ```.`item'```
= local value `"cb_authors_ext estimated wg_ndx_mult stky_pr_r
> otemberg wlth stky_pr_calvo ln_neq open"
- sreturn local value ```.value```
= sreturn local value `"cb_authors_ext estimated wg_ndx_mult s
> tky_pr_rotemberg wlth stky_pr_calvo ln_neq open"
- class exit ```.value```
= class exit `"cb_authors_ext estimated wg_ndx_mult stky_pr_ro
> temberg wlth stky_pr_calvo ln_neq open"

> _____ end _eqspec.item _____
- class exit ```.s(value)```
= class exit `"cb_authors_ext estimated wg_ndx_mult stky_pr_rote
> mberg wlth stky_pr_calvo ln_neq open"

> _____ end _eqlist.eq _____
= local xvarsi `"cb_authors_ext estimated wg_ndx_mult stky_pr_rote
> mberg wlth stky_pr_calvo ln_neq open"
- local noconsi ```.eq nocons `i'```

> _____ begin _eqlist.eq _____
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "nocons" == "count" {
    sreturn local value ```.n_eq```
    class exit ```.s(value)```

```

```

    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item nocons
  }
  > ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, nocons
  > ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
  > IF IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
      local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
  > ther' `nocons' `offset' `exposure' `freeparm' `options'
    = local item nocons
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
      error 198
    }
    - c_local `c_item' `item'
    = c_local item nocons
  > ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```
    = local value ``0```
    - sreturn local value ```.value```
    = sreturn local value ``0```
    - class exit ``.value```
    = class exit ``0```
  > ----- end _eqspec.item -----
    - class exit ``s(value)```
    = class exit ``0```
  > ----- end _eqlist.eq -----
    = local noconsi ``0```
    - local dim = `dim' + `:list sizeof xvarsi'

```

```

= local dim = + 8
- if "`noconsi'" == "0" {
= if "0" == "0" {
- local ++dim
- }
- }
- local xvarsi ```.eq indepvars `i'"""

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "indepvars" == "count" {
    sreturn local value ```.n_eq'''
    class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item indepvars

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, indepvars

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPosure FREEParm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
    local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
    error 198
}
- c_local `c_item' `item'
= c_local item indepvars

```



```

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.item'''
    = local value `'''
    - sreturn local value ```.value'''
    = sreturn local value `'''
    - class exit ```.value'''
    = class exit `'''

> ----- end _eqspec.item -----
    - class exit ```.s(value)'''
    = class exit `'''

> ----- end _eqlist.eq -----
    = local xvarsi `'''
    - local noconsi ```.eq nocons `i'''

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``.item' == "count" {
    = if "nocons" == "count" {
        sreturn local value ```.n_eq'''
        class exit ```.s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 2 & 2 <= 2 {
        - .eq_list[`idx'].item `item'
    = .eq_list[2].item nocons

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - ._check_eq_item_name item, `item'
    = ._check_eq_item_name item, nocons

> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars
> IF IN WGT OTHER NOCONS OFFset EXPOsure FREEParM OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
        local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `o
> ther' `nocons' `offset' `exposure' `freeparm' `options'

```

```

        = local item          nocons
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item nocons
    _____
> _____ end _eqspec._check_eq_item_name _____
        - local value ```.`item'"""
        = local value ``0""
        - sreturn local value ```value'"""
        = sreturn local value ``0""
        - class exit ```value'"""
        = class exit ``0""
    _____

> _____ end _eqspec.item _____
        - class exit ```s(value)"""
        = class exit ``0""
    _____

> _____ end _eqlist.eq _____
        = local noconsi ``0""
        - local dim = `dim' + `:list sizeof xvarsi'
        = local dim = 9 + 0
        - if ``noconsi' == "0" {
        = if "0" == "0" {
            - local ++dim
            - }
        - }
        - sreturn local value ```dim'"""
        = sreturn local value ``10""
        - class exit ```s(value)"""
        = class exit ``10""
    _____

> _____ end _eqlist.dim _____
        = global ML_dim = 10
        - forval i = 1/`eq_n' {
        = forval i = 1/2 {
            - local depvars `.`eqlist'.eq depvars `i'
    _____

> _____ begin _eqlist.eq _____
        - version 9.2
        - args item idx
        - if ``item' == "count" {
        = if "depvars" == "count" {
            sreturn local value ```.n_eq'"""

```

```

        class exit ``s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item depvars
    }
> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, depvars
    }
> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
        local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
    = local item depvars
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
        error 198
    }
    - c_local `c_item' `item'
    = c_local item depvars
    }
> ----- end _eqspec._check_eq_item_name -----
    - local value ``.`item''''
    = local value `y_timing_max'
    - sreturn local value ``value'''
    = sreturn local value `y_timing_max'
    - class exit ``value'''
    = class exit `y_timing_max'
    }
> ----- end _eqspec.item -----
    - class exit ``s(value)'''
    = class exit `y_timing_max'
    }
> ----- end _eqlist.eq -----
    = local depvars y_timing_max

```

```

- local eq ``.eqlist'.eq name `i'
> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "name" == "count" {
  sreturn local value ``.n_eq'
  class exit ``s(value)''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item name
> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, name
> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item name
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item name
> ----- end _eqspec._check_eq_item_name -----
- local value ``.item''
= local value ``y_timing_max''
- sreturn local value ``value''
= sreturn local value ``y_timing_max''
- class exit ``value''

```

```

        = class exit `"y_timing_max"'
    > ----- end _eqspec.item -----
        - class exit `"'s(value)'"
        = class exit `"y_timing_max"'
    > ----- end _eqlist.eq -----
        = local eq y_timing_max
        - if `!:length local eq' {
        = if !12 {
            local eq eq`i'
        }
        - if `:list eq in eqnames' {
        = if 0 {
            di as err "equation ``eq'' multiply defined"
            exit 110
        }
        - local eqnames `eqnames' `eq'
        = local eqnames y_timing_max
        - fvexpand `.`eqlist'.eq indepvars `i'' if $ML_samp
    > ----- begin _eqlist.eq -----
        - version 9.2
        - args item idx
        - if ``item'" == "count" {
        = if "indepvars" == "count" {
            sreturn local value `"'n_eq'"
            class exit `"'s(value)'"
        }
        - else if 0 < `idx' & `idx' <= `.n_eq' {
        = else if 0 < 1 & 1 <= 2 {
            - .eq_list[`idx'].item `item'
            = .eq_list[1].item indepvars
    > ----- begin _eqspec.item -----
        - version 9.2
        - args item
        - .check_eq_item_name item, `item'
        = .check_eq_item_name item, indepvars
    > ----- begin _eqspec._check_eq_item_name -----
        - version 9.2
        - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
        - if c(rc) error 198
        - if `:list sizeof other' {

```

```

        = if 0 {
            local other other_markvars
        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
        = local item indepvars
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item indepvars
    }
}

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```
    = local value `"cb_authors_ext estimated wg_ndx_mult stky_pr_rot
> emberg wlth stky_pr_calvo ln_neq open"
    - sreturn local value ```.`value'```
    = sreturn local value `"cb_authors_ext estimated wg_ndx_mult stk
> y_pr_rottemberg wlth stky_pr_calvo ln_neq open"
    - class exit ```.`value'```
    = class exit `"cb_authors_ext estimated wg_ndx_mult stky_pr_rote
> mberg wlth stky_pr_calvo ln_neq open"
}

> ----- end _eqspec.item -----
    - class exit ```.s(value)```
    = class exit `"cb_authors_ext estimated wg_ndx_mult stky_pr_rotemb
> erg wlth stky_pr_calvo ln_neq open"
}

> ----- end _eqlist.eq -----
    = fvexpand cb_authors_ext estimated wg_ndx_mult stky_pr_rotemberg wl
> th stky_pr_calvo ln_neq open if __000006
    - if "`r(tsops)'" == "true" {
    = if "" == "true" {
        global ML_preserve no
    }
    - local eq_rhs`i' ```.r(varlist)```
    = local eq_rhs1 `"cb_authors_ext estimated wg_ndx_mult stky_pr_rotem
> berg wlth stky_pr_calvo ln_neq open"
    - local eq_nocons`i' ```.eqlist'.eq nocons `i'
}

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if "`item'" == "count" {
    = if "nocons" == "count" {

```

```

        sreturn local value ```.n_eq'''
        class exit ```s(value)'''
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 1 & 1 <= 2 {
    - .eq_list[`idx'].item `item'
    = .eq_list[1].item nocons
}

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, nocons
}

> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEArm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {
    = if 0 {
        local other other_markvars
    }
    - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freearm' `options'
    = local item nocons
    - if `:list sizeof item' != 1 {
    = if 1 != 1 {
        error 198
    }
    - c_local `c_item' `item'
    = c_local item nocons
}

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'''
    = local value ``0''
    - sreturn local value ```value'''
    = sreturn local value ``0''
    - class exit ```value'''
    = class exit ``0''
}

> ----- end _eqspec.item -----
    - class exit ```s(value)'''
    = class exit ``0''
}

> ----- end _eqlist.eq -----

```

```

= local eq_nocons1 0
- local eq_offset`i' `.`eqlist'.eq offset `i''

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if "`item'" == "count" {
= if "offset" == "count" {
  sreturn local value ```.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item offset

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, offset

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item offset
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item offset

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item''''
= local value `'''
- sreturn local value ``value'''
= sreturn local value `'''

```



```

        - class exit ``value''
        = class exit `''`
    _____
> _____ end _eqspec.item _____
        - class exit ``s(value)''
        = class exit `''`
    _____
> _____ end _eqlist.eq _____
        = local eq_offset1
        - local eq_exposure`i' ``eqlist'.eq exposure `i'
    _____
> _____ begin _eqlist.eq _____
        - version 9.2
        - args item idx
        - if ``item'' == "count" {
        = if "exposure" == "count" {
            sreturn local value ``.n_eq''
            class exit ``s(value)''
        }
        - else if 0 < `idx' & `idx' <= `.n_eq' {
        = else if 0 < 1 & 1 <= 2 {
            - .eq_list[`idx'].item `item'
            = .eq_list[1].item exposure
    _____
> _____ begin _eqspec.item _____
        - version 9.2
        - args item
        - .check_eq_item_name item, `item'
        = .check_eq_item_name item, exposure
    _____
> _____ begin _eqspec._check_eq_item_name _____
        - version 9.2
        - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEParM OPTions ]
        - if c(rc) error 198
        - if `:list sizeof other' {
        = if 0 {
            local other other_markvars
        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
        = local item exposure
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }

```

```

- c_local `c_item' `item'
= c_local item exposure
_____
> _____ end _eqspec._check_eq_item_name _____
- local value ```.`item'```
= local value ``0```
- sreturn local value ```.value```
= sreturn local value ``0```
- class exit ```.value```
= class exit ``0```
_____
> _____ end _eqspec.item _____
- class exit ```.s(value)```
= class exit ``0```
_____
> _____ end _eqlist.eq _____
= local eq_exposure1 0
- local eq_freeparm`i' ``.eqlist'.eq freeparm `i'
_____
> _____ begin _eqlist.eq _____
- version 9.2
- args item idx
- if ``item' == "count" {
= if "freeparm" == "count" {
sreturn local value ```.n_eq```
class exit ```.s(value)```
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 1 & 1 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[1].item freeparm
_____
> _____ begin _eqspec.item _____
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, freeparm
_____
> _____ begin _eqspec._check_eq_item_name _____
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars

```

```

        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
        = local item freeparm
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item freeparm
    }
}

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'"""
    = local value ``0""
    - sreturn local value ```value'"""
    = sreturn local value ``0""
    - class exit ```value'"""
    = class exit ``0""

> ----- end _eqspec.item -----
    - class exit ```s(value)"""
    = class exit ``0""

> ----- end _eqlist.eq -----
    = local eq_freeparm1 0
    - local lhs `lhs' `depvars'
    = local lhs y_timing_max
    - local eq_names `eq_names' `eq'
    = local eq_names y_timing_max
    - local nxvars :list sizeof eq_rhs`i'
    = local nxvars :list sizeof eq_rhs1
    - local TMP : copy local eq_rhs`i'
    = local TMP : copy local eq_rhs1
    - forval j = 1/`nxvars' {
    = forval j = 1/8 {
        - gettoken xvar TMP : TMP
        - local colna ```colna' `eq':`xvar'"""
    = local colna `` y_timing_max:cb_authors_ext""
        - }
        - gettoken xvar TMP : TMP
        - local colna ```colna' `eq':`xvar'"""
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated"
> '
    - }
    - gettoken xvar TMP : TMP
    - local colna ```colna' `eq':`xvar'"""

```

```

        = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult''
    - }
    - gettoken xvar TMP : TMP
    - local colna ``colna' `eq':`xvar''''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg''
    - }
    - gettoken xvar TMP : TMP
    - local colna ``colna' `eq':`xvar''''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_max:wlth''
    - }
    - gettoken xvar TMP : TMP
    - local colna ``colna' `eq':`xvar''''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_max:wlth y_
> timing_max:stky_pr_calvo''
    - }
    - gettoken xvar TMP : TMP
    - local colna ``colna' `eq':`xvar''''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_max:wlth y_
> timing_max:stky_pr_calvo y_timing_max:ln_neq''
    - }
    - gettoken xvar TMP : TMP
    - local colna ``colna' `eq':`xvar''''
    = local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_max:wlth y_
> timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open''
    - }
    - if `eq_freeparm`i'' {
= if 0 {
    if `nxvars' {
        di as err "{it:xvars} not allowed with option {bf:freeparm}"
        exit 198
    }
    if strpos(`"eq'','', ":") {
        local colna ``colna' /\`eq''''
    }
    else {
        local colna ``colna' /\:`eq''''
    }
}
- else if !`eq_nocons`i'' {
= else if !0 {

```

```

- local colna ``colna' `eq':_cons''
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlth y_
> timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_max:
> _cons''
- }
- if ``moptobj'' != "" {
= if "" != "" {
  if ``eq_offset`i'' != "" {
    if usubstr("eq_offset`i'",1,2)=="__" {
      capture clonevar `moptobj'`eq_offset`i' = `eq_offset`i'
      local dropmetoo `dropmetoo' `moptobj'`eq_offset`i'
    }
  }
}
- }
- local depvars `.`eqlist'.eq depvars `i'

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item'' == "count" {
= if "depvars" == "count" {
  sreturn local value ``.n_eq''
  class exit ``s(value)''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item depvars

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, depvars

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}

```

```

- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
- local item depvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item depvars

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'```
= local value ````
- sreturn local value ```.value```
= sreturn local value ````
- class exit ```.value```
= class exit ````

> ----- end _eqspec.item -----
- class exit ```.s(value)```
= class exit ````

> ----- end _eqlist.eq -----
= local depvars
- local eq `.`eqlist'.eq name `i'

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "name" == "count" {
sreturn local value ```.n_eq```
class exit ```.s(value)```
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item name

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, name

> ----- begin _eqspec._check_eq_item_name -----

```

```

- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item name
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item name

> ----- end _eqspec._check_eq_item_name -----
- local value ```.`item'"""
= local value `"\lnalpha"
- sreturn local value ```\value'"""
= sreturn local value `"\lnalpha"
- class exit ```\value'"""
= class exit `"\lnalpha"

> ----- end _eqspec.item -----
- class exit ```\s(value)"""
= class exit `"\lnalpha"

> ----- end _eqlist.eq -----
= local eq lnalpha
- if !`:length local eq' {
= if !7 {
  local eq eq`i'
}
- if `:list eq in eqnames' {
= if 0 {
  di as err "equation ``eq'' multiply defined"
  exit 110
}
- local eqnames `eqnames' `eq'
= local eqnames y_timing_max lnalpha
- fvexpand `.`eqlist'.eq indepvars `i' if $ML_samp

> ----- begin _eqlist.eq -----

```

```

- version 9.2
- args item idx
- if ``item'' == "count" {
= if "indepvars" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item indepvars
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, indepvars
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParM OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item indepvars
- if `:list sizeof item' != 1 {
= if 1 != 1 {
  error 198
}
- c_local `c_item' `item'
= c_local item indepvars
}

> ----- end _eqspec._check_eq_item_name -----
- local value ``.item'''
= local value `'''
- sreturn local value ``value'''
= sreturn local value `'''
- class exit ``value'''
= class exit `'''
}

> ----- end _eqspec.item -----

```



```

- class exit ``s(value)'''
= class exit `'''

```

---

```

> ----- end _eqlist.eq -----
= fvexpand if __000006
- if ``r(tsops)''' == "true" {
= if "" == "true" {
  global ML_preserve no
}
- local eq_rhs`i' ``r(varlist)'''
= local eq_rhs2 `'''
- local eq_nocons`i' ``.eqlist'.eq nocons `i''

```

---

```

> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item''' == "count" {
= if "nocons" == "count" {
  sreturn local value ``.n_eq'''
  class exit ``s(value)'''
}
- else if 0 < `idx' & `idx' <= `.n_eq' {
= else if 0 < 2 & 2 <= 2 {
- .eq_list[`idx'].item `item'
= .eq_list[2].item nocons

```

---

```

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, nocons

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPOsure FREEparm OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
  local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item nocons
- if `:list sizeof item' != 1 {
= if 1 != 1 {

```

```

        error 198
    }
    - c_local `c_item' `item'
    = c_local item nocons

```

---

```

> ----- end _eqspec._check_eq_item_name -----
    - local value ```.`item'```
    = local value ``0```
    - sreturn local value ```value'```
    = sreturn local value ``0```
    - class exit ```value'```
    = class exit ``0```

```

---

```

> ----- end _eqspec.item -----
    - class exit ```s(value)'```
    = class exit ``0```

```

---

```

> ----- end _eqlist.eq -----
    = local eq_nocons2 0
    - local eq_offset`i' ``.eqlist'.eq offset `i'

```

---

```

> ----- begin _eqlist.eq -----
    - version 9.2
    - args item idx
    - if ``item' == "count" {
    = if "offset" == "count" {
        sreturn local value ```.n_eq```
        class exit ```s(value)'```
    }
    - else if 0 < `idx' & `idx' <= `.n_eq' {
    = else if 0 < 2 & 2 <= 2 {
        - .eq_list[`idx'].item `item'
        = .eq_list[2].item offset
    }

```

---

```

> ----- begin _eqspec.item -----
    - version 9.2
    - args item
    - .check_eq_item_name item, `item'
    = .check_eq_item_name item, offset

```

---

```

> ----- begin _eqspec._check_eq_item_name -----
    - version 9.2
    - capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParm OPTions ]
    - if c(rc) error 198
    - if `:list sizeof other' {

```

```

        = if 0 {
            local other other_markvars
        }
        - local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
        = local item offset
        - if `:list sizeof item' != 1 {
        = if 1 != 1 {
            error 198
        }
        - c_local `c_item' `item'
        = c_local item offset
    _____
> _____ end _eqspec._check_eq_item_name _____
        - local value ```.`item'"""
        = local value `"""'
        - sreturn local value ```.value'"""
        = sreturn local value `"""'
        - class exit ```.value'"""
        = class exit `"""'
    _____
> _____ end _eqspec.item _____
        - class exit ```.s(value)"""
        = class exit `"""'
    _____
> _____ end _eqlist.eq _____
        = local eq_offset2
        - local eq_exposure`i' ``.eqlist'.eq exposure `i'
    _____
> _____ begin _eqlist.eq _____
        - version 9.2
        - args item idx
        - if ``item' == "count" {
        = if "exposure" == "count" {
            sreturn local value ```.n_eq'"""
            class exit ```.s(value)"""
        }
        - else if 0 < `idx' & `idx' <= `.n_eq' {
        = else if 0 < 2 & 2 <= 2 {
            - .eq_list[`idx'].item `item'
            = .eq_list[2].item exposure
    _____
> _____ begin _eqspec.item _____
        - version 9.2
        - args item
        - .check_eq_item_name item, `item'

```

```

= . _check_eq_item_name item, exposure
> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item exposure
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item exposure
> ----- end _eqspec._check_eq_item_name -----
- local value ```. `item'``
= local value ``0''
- sreturn local value ```value'``
= sreturn local value ``0''
- class exit ```value'``
= class exit ``0''
> ----- end _eqspec.item -----
- class exit ```s(value)'``
= class exit ``0''
> ----- end _eqlist.eq -----
= local eq_exposure2 0
- local eq_freeparm`i' ``. `eqlist'.eq freeparm `i'
> ----- begin _eqlist.eq -----
- version 9.2
- args item idx
- if ``item' == "count" {
= if "freeparm" == "count" {
sreturn local value ```.n_eq'``
class exit ```s(value)'``
}
- else if 0 < `idx' & `idx' <= `.n_eq' {

```

```

= else if 0 < 2 & 2 <= 2 {
- .eq_list['idx'].item `item'
= .eq_list[2].item freeparm
}

> ----- begin _eqspec.item -----
- version 9.2
- args item
- .check_eq_item_name item, `item'
= .check_eq_item_name item, freeparm
}

> ----- begin _eqspec._check_eq_item_name -----
- version 9.2
- capture syntax name(name=c_item) [, NAME DEPvars INDEPvars I
> F IN WGT OTHER NOCONS OFFset EXPosure FREEParam OPTions ]
- if c(rc) error 198
- if `:list sizeof other' {
= if 0 {
local other other_markvars
}
- local item `name' `depvars' `indepvars' `if' `in' `wgt' `oth
> er' `nocons' `offset' `exposure' `freeparm' `options'
= local item freeparm
- if `:list sizeof item' != 1 {
= if 1 != 1 {
error 198
}
- c_local `c_item' `item'
= c_local item freeparm
}

> ----- end _eqspec._check_eq_item_name -----
- local value ```.item'''
= local value `1'
- sreturn local value ``value'''
= sreturn local value `1'
- class exit ``value'''
= class exit `1'

> ----- end _eqspec.item -----
- class exit ``s(value)'''
= class exit `1'

> ----- end _eqlist.eq -----
= local eq_freeparm2 1
- local lhs `lhs' `depvars'
= local lhs y_timing_max
- local eq_names `eq_names' `eq'

```

```

= local eq_names y_timing_max lnalpha
- local nxvars :list sizeof eq_rhs`i'
= local nxvars :list sizeof eq_rhs2
- local TMP : copy local eq_rhs`i'
= local TMP : copy local eq_rhs2
- forval j = 1/`nxvars' {
= forval j = 1/0 {
    gettoken xvar TMP : TMP
    local colna ``colna' `eq':`xvar'""
}
- if `eq_freeparm`i' {
= if 1 {
- if `nxvars' {
= if 0 {
    di as err "{it:xvars} not allowed with option {bf:freeparm}"
    exit 198
}
- if strpos(`eq'"" , ":") {
= if strpos(`lnalpha'"" , ":") {
    local colna ``colna' /`eq'""
}
- else {
- local colna ``colna' /:`eq'""
= local colna `` y_timing_max:cb_authors_ext y_timing_max:estimated
> y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlth y_
> timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_max:
> _cons /:lnalpha""
- }
- }
- else if !`eq_nocons`i' {
= else if !0 {
    local colna ``colna' `eq':_cons""
}
- if "`moptobj'" != "" {
= if "" != "" {
    if "`eq_offset`i'" != "" {
    if usubstr("`eq_offset`i'",1,2)=="__" {
    capture clonevar `moptobj'`eq_offset`i' = `eq_offset`i'
    local dropmetoo `dropmetoo' `moptobj'`eq_offset`i'
}
}
}
- }
- if "`moptobj'" != "" {
= if "" != "" {
    foreach y of local lhs {

```

```

if usubstr("`y'",1,2)=="__" {
capture clonevar `moptobj'`y' = `y'
local tmp `tmp' `moptobj'`y'
local dropmetoo `dropmetoo' `moptobj'`y'
}
else {
local tmp `tmp' `y'
}
}
local lhs `tmp'
}
- if `waldtest' > 0 & `:length local lf0' {
= if -1 > 0 & 0 {
opts_exclusive "waldtest() lf0()"
}
- if `:length local continue' {
= if 8 {
- if !`:length local init' {
= if !8 {
tempname init
local ub0 `init'
}
- else tempname ub0
- mat `ub0' = e(b)
= mat __000008 = e(b)
- tempname v0 iv0
- mat `v0' = e(V)
= mat __000009 = e(V)
- mat `iv0' = syminv(`v0')
= mat __00000A = syminv(__000009)
- global ML_rank0 = colsof(`v0') - diag0cnt(`iv0')
= global ML_rank0 = colsof(__000009) - diag0cnt(__00000A)
- if !`:length local lf0' & !missing(e(ll)) {
= if !0 & !missing(e(ll)) {
- local lf0 `=colsof(`ub0')' `e(ll)'
= local lf0 2 -414.2287919313009
- }
- local ub0
- }
- if `:length local init' {
= if 8 {
- tempname b0
- local dim : list sizeof colna
- matrix `b0' = J(1,`dim',0)
= matrix __00000B = J(1,10,0)
- matrix colna `b0' = `colna'

```

```

= matrix colna __00000B = y_timing_max:cb_authors_ext y_timing_max:
> estimated y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rottemberg y_timing_m
> ax:wlth y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_t
> iming_max:_cons /:lnalpha
- `vv' _mkvec `b0', from(`init') update first error("initial vector"
> )
= version 18: _mkvec __00000B, from(__000002) update first error("in
> itial vector")

> ----- begin _mkvec -----
- local vv : display "version " string(_caller()) ":"
- version 6
- gettoken matout 0 : 0, parse(", ")
- if "`matout'"==" | "`matout'"=="," {
= if "__00000B"==" | "__00000B"=="," {
error 198
}
- syntax [, FROM(str) UPdate COLnames(str) ERRor(str) FIRST ]
- if "`update'"!=" {
= if "update"!=" {
- if "`colname'"!=" {
= if ""!=" {
di in red "update and colnames() cannot both be " "specified"
exit 198
}
- capture local colname : colfullnames `matout'
= capture local colname : colfullnames __00000B
- if _rc {
di in red "matrix `matout' not found"
exit 111
}
- if rowsof(matrix(`matout')) != 1 {
= if rowsof(matrix(__00000B)) != 1 {
di in red "`matout' matrix to be updated must be " "a row vector
> "

exit 503
}
- local matin `matout'
= local matin __00000B
- }
- if ``error'"'"'!=" {
= if `"initial vector'"'"'!=" {
- local error ``error': ''
= local error `initial vector: ''
- }
- tempname mat

```



```

- if ```from'``!=`` {
= if ``__000002'``!=`` {
- From `from'
= From __000002

> ----- begin _mkvec.From -----
- gettoken from 0 : 0, parse(",") bind
- sret clear
- sret local from `from'
= sret local from __000002
- if ```0'``!=`` {
= if ``'``!=`` {
  syntax [, COPY SKIP ]
}
- sret local copy `copy'
= sret local copy
- sret local skip `skip'
= sret local skip

> ----- end _mkvec.From -----
- local copy `s(copy)'
= local copy
- local skip `s(skip)'
= local skip
- if ```copy'``!=`` {
= if ``'``!=`` {
  CopyMat `mat' ```error'`` `s(from)'
  if ```colname'``!=`` {
    local dim : list sizeof colname
    if `dim' != colsof(matrix(`mat')) {
      di in red ```error'matrix must ' ' `be dimension ' ' `dim'
      exit 503
    }
    `vv' mat colnames `mat' = `colname'
  }
}
- else {
- `vv' BuildMat `mat' ```error'`` ``skip'`` `s(from)'
= version 18: BuildMat __00000C ``initial vector: ' ' `` __000002

> ----- begin _mkvec.BuildMat -----
- local vv : display "version " string(_caller()) ":"
- version 6
- args matall error skip
- macro shift 3
- tokenize ```*'``, parse(" =")

```

```

= tokenize `__000002`, parse(" =")
- tempname matadd
- local i 1
- while "`i'" != "" {
= while "__000002" != "" {
- local k = `i' + 1
= local k = 1 + 1
- if "`k'" == "" {
= if "" == "" {
    local j = `i' + 2
    `vv' MkMat `matadd' ``i' ``j' `skip'
    local i = `i' + 3
  }
- else {
- ChkMat `matadd' ``i' ``error'
= ChkMat __00000D __000002 `initial vector: '

```

---

```

> ----- begin _mkvec.ChkMat -----
- args matout matin error
- capture di matrix(`matin'[1,1])
= capture di matrix(__000002[1,1])
- if _rc {
  capture confirm number `matin'
  if _rc {
    di in red ``error'matrix `matin' not found'
    exit 111
  }
  di in red ``error'list of numbers requires copy option'
  exit 198
}
- if matrix(rowsof(`matin')!=1 & colsof(`matin')!=1) {
= if matrix(rowsof(__000002)!=1 & colsof(__000002)!=1) {
  di in red "`matin' " matrix(rowsof(`matin')) " x " matrix(co
> lsof(`matin')) " is not a vector"
  exit 503
}
- if matrix(rowsof(`matin')) == 1 {
= if matrix(rowsof(__000002)) == 1 {
- mat `matout' = `matin'
= mat __00000D = __000002
- }
- else mat `matout' = `matin'
= else mat __00000D = __000002'

```

---

```

> ----- end _mkvec.ChkMat -----
- local i = `i' + 1

```

```

= local i = 1 + 1
- }
- mat `matall' = nullmat(`matall') , `matadd'
= mat __00000C = nullmat(__00000C) , __00000D
- }
- while ``i'!='' {
= while ''!='' {
    local k = `i' + 1
    if ``k'=='' {
        local j = `i' + 2
        `vv' MkMat `matadd' ``i' ``j' `skip'
        local i = `i' + 3
    }
    else {
        ChkMat `matadd' ``i' ``error'
        local i = `i' + 1
    }
    mat `matall' = nullmat(`matall') , `matadd'
}

> ----- end _mkvec.BuildMat -----

- if ``colname'!='' {
= if `y_timing_max:cb_authors_ext y_timing_max:estimated y_timing
> _max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlth y_timing_m
> ax:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_max:_cons /l
> nalpha'!='' {
    - `vv' ChkNames `mat' ``matin' ``colname' ``error' `skip'
> ``first'
    = version 18: ChkNames __00000C "__00000B" `y_timing_max:cb_autho
> rs_ext y_timing_max:estimated y_timing_max:wg_ndx_mult y_timing_max:stky_pr_
> rotemberg y_timing_max:wlth y_timing_max:stky_pr_calvo y_timing_max:ln_neq y
> _timing_max:open y_timing_max:_cons /lnalpha' `initial vector: ' "" "firs
> t"

> ----- begin _mkvec.ChkNames -----

- local vv : display "version " string(_caller()) ":"
- version 6
- args mat matin names error skip first
- local dim = colsof(matrix(`mat'))
= local dim = colsof(matrix(__00000C))
- if ``first'!='' {
= if "first"!='' {
- gettoken name : names
- local eq1 = bsubstr(`name',1,index(`name',""))
= local eq1 = bsubstr(`y_timing_max:cb_authors_ext',1,index(`"
> y_timing_max:cb_authors_ext',""))

```

```

- }
- tempname new fill x
- local k : list sizeof names
- if "`matin'"!="" {
= if "__00000B"!=" " {
- mat `new' = `matin'
= mat __00000D = __00000B
- }
- else {
  mat `new' = J(1,`k',0)
  `vv' mat colnames `new' = `names'
}
- mat `fill' = J(1,`k',0)
= mat __00000E = J(1,10,0)
- local i 1
- while `i' <= `dim' {
= while 1 <= 10 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000F = __00000C[1,1..1]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
  local spec "/"`name'
}
- else {
- local spec "`eq':`name'"
= local spec "y_timing_max:cb_authors_ext"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/"`eq':`name'"
= local sspec "/y_timing_max:cb_authors_ext"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse cb_authors_ext, noomit
- if c(rc) == 0 {
- if r(omit) {
  local oname ``r(stripe)''
  local ospec "`eq':`oname'"
}
- }

```

```

- `vv' local j = colnumb(`new',`""`spec'")
= version 18: local j = colnumb(__00000D,`"y_timing_max:cb_autho
> rs_ext'')
- if `j' == . & ""`oname'" != "" {
= if 1 == . & "" != "" {
  `vv' local j = colnumb(`new',`""`ospec'")
  }
- if `j' == . & ""`sspec'" != "" {
= if 1 == . & "/y_timing_max:cb_authors_ext" != "" {
  `vv' local j = colnumb(`new',`""`sspec'")
  }
- if `j'==. & ""`eq'""=="_" & ""`eq1'""!="" {
= if 1==. & ""`y_timing_max'""=="_" & ""`y_timing_max:'""!="" {
  `vv' local j = colnumb(`new',`""`eq1'`name'")
  if `j' == . & ""`oname'" != "" {
    `vv' local j = colnumb(`new',`""`eq1'`oname'")
  }
  }
- if `j'!=. {
= if 1!=. {
- if `fill'[1,`j'] {
= if __00000E[1,1] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ""`eq'""=="_" {
      di in red ""`error'duplicate "'`"entries for `name' found"
    }
    else di in red ""`error'"" "`duplicate entries for `eq':`name'
> found"

    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
  }
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,1] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,1] = __00000C[1,1]
- }
- else if ""`skip'""=="" {
= else if ""=="" {
  if ""`eq'""=="_" {
    local spec : copy local name
  }
  di in red ""`error'extra parameter "'`""`spec' found" _n "spe

```

```

> cify skip option if necessary"
    exit 111
    }
    - local i = `i' + 1
    = local i = 1 + 1
    - }
    - while `i' <= `dim' {
    = while 2 <= 10 {
    - mat `x' = `mat'[1,`i'..`i']
    = mat __00000F = __00000C[1,2..2]
    - local eq : coleq `x'
    = local eq : coleq __00000F
    - local name : colname `x'
    = local name : colname __00000F
    - if "`eq'" == "/" {
    = if "y_timing_max" == "/" {
        local spec "/"`name'"
    }
    - else {
    - local spec "`eq':`name'"
    = local spec "y_timing_max:estimated"
    - if (usubstr("`eq'",1,1) != "/") {
    = if (usubstr("y_timing_max",1,1) != "/") {
    - local sspec "/"`eq':`name'"
    = local sspec "/y_timing_max:estimated"
    - }
    - }
    - local oname
    - local ospec
    - capture _msparse `name', noomit
    = capture _msparse estimated, noomit
    - if c(rc) == 0 {
    - if r(omit) {
        local oname ``r(stripe)'"
        local ospec "`eq':`oname'"
    }
    - }
    - `vv' local j = colnumb(`new',``spec'')
    = version 18: local j = colnumb(__00000D,`"y_timing_max:estimate
> d'')

    - if `j' == . & "`oname'" != "" {
    = if 2 == . & "" != "" {
        `vv' local j = colnumb(`new',``ospec'')
    }
    - if `j' == . & "`sspec'" != "" {
    = if 2 == . & "/y_timing_max:estimated" != "" {

```

```

        `vv' local j = colnumb(`new',````sspec'``')
    }
- if `j'==. & ``eq'``'=="_" & ``eq1'``'!="" {
= if 2==. & ``y_timing_max'``'=="_" & ``y_timing_max:'``'!="" {
    `vv' local j = colnumb(`new',````eq1'`name'``')
    if `j' == . & ``oname'``' != "" {
        `vv' local j = colnumb(`new',````eq1'`oname'``')
    }
}
- if `j'!=. {
= if 2!=. {
- if `fill'[1,`j'] {
= if __00000E[1,2] {
    _ms_parse_parts `name'
    if !r(omit) {
        if ``eq'``'=="_" {
            di in red ``error'duplicate "' `entries for `name' found"'
        }
        else di in red ``error'``' `duplicate entries for `eq':`name'
> found"'
    }
    exit 507
}
else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,2] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,2] = __00000C[1,2]
- }
- else if ``skip'``'="" {
= else if ""="" {
    if ``eq'``'=="_" {
        local spec : copy local name
    }
    di in red ``error'extra parameter "' ``spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
}
- local i = `i' + 1
= local i = 2 + 1
- }
- while `i' <= `dim' {
= while 3 <= 10 {
- mat `x' = `mat'[1,`i'..`i']

```

```

= mat __00000F = __00000C[1,3..3]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
    local spec "/"`name'
  }
- else {
- local spec "`eq':`name'"
= local spec "y_timing_max:wg_ndx_mult"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/"`eq':`name'"
= local sspec "/y_timing_max:wg_ndx_mult"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse wg_ndx_mult, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)''
    local ospec "`eq':`oname'"
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:wg_ndx_m
> ult'')
- if `j' == . & "`oname'" != "" {
= if 3 == . & "" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
- if `j' == . & "`sspec'" != "" {
= if 3 == . & "/y_timing_max:wg_ndx_mult" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
- if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
= if 3==. & `"y_timing_max'""=="_" & `"y_timing_max:'""!="" {
    `vv' local j = colnumb(`new',``eq1``name'')
    if `j' == . & "`oname'" != "" {
      `vv' local j = colnumb(`new',``eq1``oname'')
    }
  }
}
}

```



```

- if `j'!=. {
= if 3!=. {
- if `fill'[1,`j'] {
= if __00000E[1,3] {
    _ms_parse_parts `name'
    if !r(omit) {
    if "`eq'"=="_" {
    di in red `"'error'duplicate "'`"entries for `name' found"'
    }
    else di in red `"'error'"'"'`"duplicate entries for `eq':`name'
> found"'

    exit 507
    }
    else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
    }
    }
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,3] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,3] = __00000C[1,3]
- }
- else if "`skip'"==" " {
= else if ""==" " {
    if "`eq'"=="_" {
    local spec : copy local name
    }
    di in red `"'error'extra parameter "'`"spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
    }
- local i = `i' + 1
= local i = 3 + 1
- }
- while `i' <= `dim' {
= while 4 <= 10 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000F = __00000C[1,4..4]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
    local spec "/"`name'"
    }
}

```

```

- else {
- local spec "`eq':`name'"
= local spec "y_timing_max:stky_pr_rottemberg"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "/y_timing_max:stky_pr_rottemberg"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse stky_pr_rottemberg, noomit
- if c(rc) == 0 {
- if r(omit) {
local oname ``r(stripe)''
local ospec ``eq':`oname'
}
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:stky_pr_
> rottemberg"')
- if `j' == . & ``oname' != "" {
= if 4 == . & "" != "" {
`vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & ``sspec' != "" {
= if 4 == . & "/y_timing_max:stky_pr_rottemberg" != "" {
`vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'=="_" & ``eq1'!="" {
= if 4==. & `y_timing_max'=="_" & `y_timing_max:'!="" {
`vv' local j = colnumb(`new',``eq1``name'')
if `j' == . & ``oname' != "" {
`vv' local j = colnumb(`new',``eq1``oname'')
}
}
- if `j'!=. {
= if 4!=. {
- if `fill'[1,`j'] {
= if __00000E[1,4] {
_ms_parse_parts `name'
if !r(omit) {
if "`eq'=="_" {
di in red ``error'duplicate "" ``entries for `name' found''
}
}
}

```

```

else di in red ``error'' ``duplicate entries for `eq':`name'
> found''
    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,4] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,4] = __00000C[1,4]
- }
- else if ``skip'')=="" {
= else if ""==" {
  if ``eq'')== "_" {
    local spec : copy local name
  }
  di in red ``error'extra parameter "' ``spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
  }
  - local i = `i' + 1
  = local i = 4 + 1
  - }
  - while `i' <= `dim' {
  = while 5 <= 10 {
  - mat `x' = `mat'[1,`i'..`i']
  = mat __00000F = __00000C[1,5..5]
  - local eq : coleq `x'
  = local eq : coleq __00000F
  - local name : colname `x'
  = local name : colname __00000F
  - if ``eq'')== "/" {
  = if "y_timing_max" == "/" {
    local spec "/"`name'
  }
  - else {
  - local spec "`eq':`name'"
  = local spec "y_timing_max:wlth"
  - if (usubstr("`eq'",1,1) != "/") {
  = if (usubstr("y_timing_max",1,1) != "/") {
  - local sspec "/"`eq':`name'"
  = local sspec "/y_timing_max:wlth"
  - }
  - }

```

```

- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse wlth, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)'''
    local ospec ``eq':`name'''
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:wlth"')
- if `j' == . & ``oname''' != "" {
= if 5 == . & "" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
- if `j' == . & ``sspec''' != "" {
= if 5 == . & "/y_timing_max:wlth" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
- if `j'==. & ``eq'''=="_" & ``eq1'''!="" {
= if 5==. & ``y_timing_max'''=="_" & ``y_timing_max:'''!="" {
    `vv' local j = colnumb(`new',``eq1``name'')
    if `j' == . & ``oname''' != "" {
      `vv' local j = colnumb(`new',``eq1``oname'')
    }
  }
- if `j'!=. {
= if 5!=. {
- if `fill'[1,`j'] {
= if __00000E[1,5] {
    _ms_parse_parts `name'
    if !r(omit) {
      if ``eq'''=="_" {
        di in red ``error'duplicate "'`"entries for `name' found"'
      }
      else di in red ``error''''`"duplicate entries for `eq':`name'
> found'''
    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,5] = 1

```

```

- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,5] = __00000C[1,5]
- }
- else if "`skip'"==" {
= else if ""==" {
    if "`eq'"=="_" {
        local spec : copy local name
    }
    di in red `"'error'extra parameter "' `"'spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
}
- local i = `i' + 1
= local i = 5 + 1
- }
- while `i' <= `dim' {
= while 6 <= 10 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000F = __00000C[1,6..6]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
    local spec "/"`name'
}
- else {
- local spec "`eq':`name'"
= local spec "y_timing_max:stky_pr_calvo"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/"`eq':`name'"
= local sspec "/y_timing_max:stky_pr_calvo"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse stky_pr_calvo, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname `"'r(stripe)'"
    local ospec "`eq':`oname'"
}
- }

```

```

- `vv' local j = colnumb(`new',`""spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:stky_pr_
> calvo'')
- if `j' == . & ""oname'" != "" {
= if 6 == . & "" != "" {
  `vv' local j = colnumb(`new',`""ospec'')
  }
- if `j' == . & ""sspec'" != "" {
= if 6 == . & "/y_timing_max:stky_pr_calvo" != "" {
  `vv' local j = colnumb(`new',`""sspec'')
  }
- if `j'==. & ""`eq'""=="_" & ""`eq1'""!="" {
= if 6==. & ""y_timing_max""=="_" & ""y_timing_max:""!="" {
  `vv' local j = colnumb(`new',`""eq1``name'')
  if `j' == . & ""oname'" != "" {
    `vv' local j = colnumb(`new',`""eq1``oname'')
  }
  }
- if `j'!=. {
= if 6!=. {
- if `fill'[1,`j'] {
= if __00000E[1,6] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ""`eq'""=="_" {
      di in red ""`error'duplicate "" ``entries for `name' found""
    }
    else di in red ""`error'"" ``duplicate entries for `eq':`name'
> found""
  }
  exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
  }
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,6] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,6] = __00000C[1,6]
- }
- else if ""skip'""=="" {
= else if ""=="" {
  if ""`eq'""=="_" {
    local spec : copy local name
  }
  di in red ""`error'extra parameter "" ``spec' found"" _n "spe

```

```

> cify skip option if necessary"
    exit 111
    }
    - local i = `i' + 1
    = local i = 6 + 1
    - }
    - while `i' <= `dim' {
    = while 7 <= 10 {
    - mat `x' = `mat'[1,`i'..`i']
    = mat __00000F = __00000C[1,7..7]
    - local eq : coleq `x'
    = local eq : coleq __00000F
    - local name : colname `x'
    = local name : colname __00000F
    - if "`eq'" == "/" {
    = if "y_timing_max" == "/" {
        local spec "/"`name'"
    }
    - else {
    - local spec "`eq':`name'"
    = local spec "y_timing_max:ln_neq"
    - if (usubstr("`eq'",1,1) != "/") {
    = if (usubstr("y_timing_max",1,1) != "/") {
    - local sspec "/"`eq':`name'"
    = local sspec "/y_timing_max:ln_neq"
    - }
    - }
    - local oname
    - local ospec
    - capture _msparse `name', noomit
    = capture _msparse ln_neq, noomit
    - if c(rc) == 0 {
    - if r(omit) {
        local oname ``r(stripe)'"
        local ospec "`eq':`oname'"
    }
    - }
    - `vv' local j = colnumb(`new',``spec'')
    = version 18: local j = colnumb(__00000D,`"y_timing_max:ln_neq"')

> )

    - if `j' == . & "`oname'" != "" {
    = if 7 == . & "" != "" {
        `vv' local j = colnumb(`new',``ospec'')
    }
    - if `j' == . & "`sspec'" != "" {
    = if 7 == . & "/y_timing_max:ln_neq" != "" {

```

```

        `vv' local j = colnumb(`new', ```sspec'`)
    }
- if `j'==. & ```eq'`'=="_" & ```eq1'`'!="" {
= if 7==. & ``y_timing_max'`'=="_" & ``y_timing_max:'`'!="" {
    `vv' local j = colnumb(`new', ```eq1'`name'`)
    if `j' == . & ``oname'`' != "" {
        `vv' local j = colnumb(`new', ```eq1'`oname'`)
    }
}
- if `j'!=. {
= if 7!=. {
- if `fill'[1,`j'] {
= if __00000E[1,7] {
    _ms_parse_parts `name'
    if !r(omit) {
        if ```eq'`'=="_" {
            di in red ```error'duplicate "'`entries for `name' found"'
        }
        else di in red ```error'``' ``duplicate entries for `eq':`name'
> found"'
    }
    exit 507
}
else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,7] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,7] = __00000C[1,7]
- }
- else if ```skip'`'==" " {
= else if ""==" " {
    if ```eq'`'=="_" {
        local spec : copy local name
    }
    di in red ```error'extra parameter "'` ```spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
}
- local i = `i' + 1
= local i = 7 + 1
- }
- while `i' <= `dim' {
= while 8 <= 10 {
- mat `x' = `mat'[1,`i'..`i']

```



```

= mat __00000F = __00000C[1,8..8]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
    local spec "/"`name'
  }
- else {
- local spec "`eq':`name'"
= local spec "y_timing_max:open"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/"`eq':`name'"
= local sspec "/y_timing_max:open"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse open, noomit
- if c(rc) == 0 {
- if r(omit) {
    local oname ``r(stripe)'''
    local ospec "`eq':`oname'"
  }
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:open"')
- if `j' == . & ``oname'" != "" {
= if 8 == . & "" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
- if `j' == . & ``sspec'" != "" {
= if 8 == . & "/y_timing_max:open" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
- if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
= if 8==. & ``y_timing_max'""=="_" & ``y_timing_max:'""!="" {
    `vv' local j = colnumb(`new',``eq1``name'')
    if `j' == . & ``oname'" != "" {
      `vv' local j = colnumb(`new',``eq1``oname'')
    }
  }
- if `j'!=. {

```

```

= if 8!=. {
- if `fill'[1,`j'] {
= if __00000E[1,8] {
  _ms_parse_parts `name'
  if !r(omit) {
    if "`eq'"=="_" {
      di in red ``error'duplicate "' `entries for `name' found"'
    }
    else di in red ``error'""' `duplicate entries for `eq':`name'
> found"'

    exit 507
  }
  else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
  }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,8] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,8] = __00000C[1,8]
- }
- else if "`skip'"==" " {
= else if ""==" " {
  if "`eq'"=="_" {
    local spec : copy local name
  }
  di in red ``error'extra parameter "' `spec' found"' _n "spe
> cify skip option if necessary"
  exit 111
}
- local i = `i' + 1
= local i = 8 + 1
- }
- while `i' <= `dim' {
= while 9 <= 10 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000F = __00000C[1,9..9]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "y_timing_max" == "/" {
  local spec "/"`name'
}
- else {

```

```

- local spec "`eq':`name'"
= local spec "y_timing_max:_cons"
- if (usubstr("`eq'",1,1) != "/") {
= if (usubstr("y_timing_max",1,1) != "/") {
- local sspec "/`eq':`name'"
= local sspec "/y_timing_max:_cons"
- }
- }
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse _cons, noomit
- if c(rc) == 0 {
- if r(omit) {
  local oname ``r(stripe)''
  local ospec "`eq':`oname'"
}
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"y_timing_max:_cons"')
- if `j' == . & ``oname' != "" {
= if 9 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & ``sspec' != "" {
= if 9 == . & "/y_timing_max:_cons" != "" {
  `vv' local j = colnumb(`new',``sspec'')
}
- if `j'==. & ``eq'=="_" & ``eq1'!="" {
= if 9==. & ``y_timing_max'=="_" & ``y_timing_max:'!="" {
  `vv' local j = colnumb(`new',``eq1`name'')
  if `j' == . & ``oname' != "" {
    `vv' local j = colnumb(`new',``eq1`oname'')
  }
}
- if `j'!=. {
= if 9!=. {
- if `fill'[1,`j'] {
= if __00000E[1,9] {
  _ms_parse_parts `name'
  if !r(omit) {
    if "`eq'=="_" {
      di in red ``error'duplicate "'`entries for `name' found"'
    }
    else di in red ``error'""`duplicate entries for `eq':`name'

```

> found''

```

        exit 507
    }
    else if !missing(`new'[1,`j']) {
        matrix `mat'[1,`i'] = `new'[1,`j']
    }
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,9] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,9] = __00000C[1,9]
- }
- else if "`skip'"==" {
= else if ""==" {
    if "`eq'"=="_" {
        local spec : copy local name
    }
    di in red ``error'extra parameter "' ``spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
}
- local i = `i' + 1
= local i = 9 + 1
- }
- while `i' <= `dim' {
= while 10 <= 10 {
- mat `x' = `mat'[1,`i'..`i']
= mat __00000F = __00000C[1,10..10]
- local eq : coleq `x'
= local eq : coleq __00000F
- local name : colname `x'
= local name : colname __00000F
- if "`eq'" == "/" {
= if "/" == "/" {
- local spec "/`name'"
= local spec "/lnalpha"
- }
- else {
    local spec "`eq':`name'"
    if (usubstr("`eq'",1,1) != "/") {
        local sspec "/`eq':`name'"
    }
}
- local oname
- local ospec
- capture _msparse `name', noomit
= capture _msparse lnalpha, noomit

```

```

- if c(rc) == 0 {
- if r(omit) {
  local oname ``r(stripe)''
  local ospec ``eq':`oname''
}
- }
- `vv' local j = colnumb(`new',``spec'')
= version 18: local j = colnumb(__00000D,`"/lnalpha'')
- if `j' == . & ``oname'' != "" {
= if 10 == . & "" != "" {
  `vv' local j = colnumb(`new',``ospec'')
}
- if `j' == . & ``spec'' != "" {
= if 10 == . & "/y_timing_max:_cons" != "" {
  `vv' local j = colnumb(`new',``spec'')
}
- if `j'==. & ``eq''==_" & ``eq1''!= "" {
= if 10==. & `"/'==_" & `y_timing_max:'!= "" {
  `vv' local j = colnumb(`new',``eq1``name'')
  if `j' == . & ``oname'' != "" {
    `vv' local j = colnumb(`new',``eq1``oname'')
  }
}
- if `j'!=. {
= if 10!=. {
- if `fill'[1,`j'] {
= if __00000E[1,10] {
  _ms_parse_parts `name'
  if !r(omit) {
    if ``eq''==_" {
      di in red ``error'duplicate '' ``entries for `name' found''
    }
    else di in red ``error'' ``duplicate entries for `eq':`name'
> found''

  exit 507
}
else if !missing(`new'[1,`j']) {
  matrix `mat'[1,`i'] = `new'[1,`j']
}
}
- else mat `fill'[1,`j'] = 1
= else mat __00000E[1,10] = 1
- mat `new'[1,`j'] = `mat'[1,`i']
= mat __00000D[1,10] = __00000C[1,10]
- }
- else if ``skip''=="" {

```

```

= else if ""==" {
  if "`eq'"=="_" {
    local spec : copy local name
  }
  di in red ``error'extra parameter "' ``spec' found"' _n "spe
> cify skip option if necessary"
  exit 111
}
- local i = `i' + 1
= local i = 10 + 1
- }
- while `i' <= `dim' {
= while 11 <= 10 {
  mat `x' = `mat'[1,`i'..`i']
  local eq : coleq `x'
  local name : colname `x'
  if "`eq'" == "/" {
    local spec "/"`name'
  }
  else {
    local spec "`eq':`name'
    if (usubstr("`eq'",1,1) != "/") {
      local sspec "/"`eq':`name'
    }
  }
  local oname
  local ospec
  capture _msparse `name', noomit
  if c(rc) == 0 {
    if r(omit) {
      local oname ``r(stripe)'''
      local ospec "`eq':`oname'
    }
  }
  `vv' local j = colnumb(`new',``spec'')
  if `j' == . & "`oname'" != "" {
    `vv' local j = colnumb(`new',``ospec'')
  }
  if `j' == . & "`sspec'" != "" {
    `vv' local j = colnumb(`new',``sspec'')
  }
  if `j'==. & ``eq'""=="_" & ``eq1'""!="" {
    `vv' local j = colnumb(`new',``eq1'`name'')
    if `j' == . & "`oname'" != "" {
      `vv' local j = colnumb(`new',``eq1'`oname'')
    }
  }
}

```

```

    }
    if `j'!=. {
    if `fill'[1,`j'] {
    _ms_parse_parts `name'
    if !r(omit) {
    if "`eq'"=="_" {
    di in red `"'`error'duplicate "' `"'entries for `name' found"'
    }
    else di in red `"'`error'""' `"'duplicate entries for `eq':`name'
> found"'

    exit 507
    }
    else if !missing(`new'[1,`j']) {
    matrix `mat'[1,`i'] = `new'[1,`j']
    }
    }
    else mat `fill'[1,`j'] = 1
    mat `new'[1,`j'] = `mat'[1,`i']
    }
    else if "`skip'"==" " {
    if "`eq'"=="_" {
    local spec : copy local name
    }
    di in red `"'`error'extra parameter "' `"'`spec' found"' _n "spe
> cify skip option if necessary"
    exit 111
    }
    local i = `i' + 1
    }
    - mat `fill' = `fill'*`fill''
    = mat __00000E = __00000E*__00000E'
    - sret clear
    - sret local k_fill = `fill'[1,1]
    = sret local k_fill = __00000E[1,1]
    - mat `mat' = `new'
    = mat __00000C = __00000D

    _____
> _____ end _mkvec.ChkNames _____

    - local fill `s(k_fill)'
    = local fill 10
    - }
    - }
    - }
    - else {
    local fill 0
    if "`update'"!=" " {

```





```

local clist : subinstr local constraints "," " ", all
`vv' makecns `clist', `cnsnotes'
local constraints ``r(clist)'''
local k_autoCns = e(k_autoCns)
tempname C
capture matrix `C' = e(Cns)
local rcc = c(rc)
}
local rc = c(rc)
ereturn clear
}
if (`rc') exit `rc'
if (`rcc') {
local C
local k_autoCns
}
- if `:length local constraints' {
= if 0 {
local lf0
}
- local vce `vce' `robust' `cluster'
= local vce vce(robust) robust
- if `:length local vce' {
= if 18 {
- _vce_parse, argoptlist(CList) optlist(OIM OPG Robust) old : `wt'
> , `vce'
= _vce_parse, argoptlist(CList) optlist(OIM OPG Robust) old : , vc
> e(robust) robust


---


> 

---

 begin _vce_parse 

---


- version 10
- _on_colon_parse `0'
= _on_colon_parse, argoptlist(CList) optlist(OIM OPG Robust) ol
> d : , vce(robust) robust


---


> 

---

 begin _on_colon_parse 

---


- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if ""par"" != "" {
= if "" != "" {
local before `"(before)'"
}
- if ``before'"" == ":" {

```

```

= if `", argoptlist(CLister) optlist(OIM OPG Robust) old "" == "
> : " {
    sreturn local after `""after'""
    exit
    }
- while `""COLON'"" != ":" & `""after'"" != "" {
= while `""' != ":" & `": , vce(robust) robust"" != "" {
- gettoken COLON after : after, parse(":") bind match(par) quote
> s
- if "`par'" != "" {
= if "" != "" {
    local before `before' (`COLON')
    local COLON
    }
- else if `""COLON'"" != ":" {
= else if `":' != ":" {
    local before `""before' `COLON'""
    local COLON
    }
- }
- while `""COLON'"" != ":" & `""after'"" != "" {
= while `":' != ":" & `", vce(robust) robust"" != "" {
    gettoken COLON after : after, parse(":") bind match(par) quote
> s
    if "`par'" != "" {
        local before `before' (`COLON')
        local COLON
        }
    else if `""COLON'"" != ":" {
        local before `""before' `COLON'""
        local COLON
        }
    }
- if `""COLON'"" != ":" {
= if `":' != ":" {
    di as err ""' found where ':' expected"
    exit 198
    }
- sreturn local before `""before'""
= sreturn local before `", argoptlist(CLister) optlist(OIM OPG R
> obust) old ""
- sreturn local after `""after'""
= sreturn local after `", vce(robust) robust""

-----
> ----- end _on_colon_parse -----
- local 0 `""s(before)""

```

```

= local 0 `", argoptlist(CLister) optlist(OIM OPG Robust) old ""
- local user ``s(after)""
= local user `", vce(robust) robust"
- syntax [varname(default=none)], [ OPTlist(namelist) ARGOPTlist(n
> amelist) PWallowed(namelist) OLD_is_ok NOARGOPT_is_ok ]
- local touse `varlist'
= local touse
- tempname o a
- .`o' = ._optlist.new
= __00000D = ._optlist.new
- local optlist : list uniq optlist
- foreach name of local optlist {
- .`o'.addopt `name'
= __00000D.addopt OIM
}

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OIM")
- if "`passthru'" != "" {
= if "" != "" {
local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "OIM"
- .names.Arrpush "`mname'"
= .names.Arrpush "oim"
}

> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= __00000D.addopt OPG
}

> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("OPG")
- if "`passthru'" != "" {
= if "" != "" {
local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "OPG"
- .names.Arrpush "`mname'"
}

```

```

= .names.Arrpush "opg"
> ----- end _optlist.addopt -----
- }
- .`o'.addopt `name'
= .__00000D.addopt Robust
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("Robust")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "Robust"
- .names.Arrpush "`mname'"
= .names.Arrpush "robust"
> ----- end _optlist.addopt -----
- }
- .`a' = ._optlist.new
= .__00000E = ._optlist.new
- local argoptlist : list uniq argoptlist
- foreach name of local argoptlist {
- .`a'.addopt `name'
= .__00000E.addopt CLuster
> ----- begin _optlist.addopt -----
- version 9
- syntax name(name=optspec) [, PASSthru]
- local mname = lower("`optspec'")
= local mname = lower("CLuster")
- if "`passthru'" != "" {
= if "" != "" {
    local optspec `optspec'(passthru)
}
- .options.Arrpush "`optspec'"
= .options.Arrpush "CLuster"
- .names.Arrpush "`mname'"
= .names.Arrpush "cluster"
> ----- end _optlist.addopt -----
- }

```

```

- local OPTNAMES ```.`o'.dumpnames'''
-----
> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/3 {
- local namelist `namelist' `'.names[`i']'
= local namelist oim
- }
- local namelist `namelist' `'.names[`i']'
= local namelist oim opg
- }
- local namelist `namelist' `'.names[`i']'
= local namelist oim opg robust
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist ``oim opg robust''
- class exit ```namelist'''
= class exit ``oim opg robust''
-----
> ----- end _optlist.dumpnames -----
= local OPTNAMES ``oim opg robust''
- local ARGOPTNAMES ```.`a'.dumpnames'''
-----
> ----- begin _optlist.dumpnames -----
- version 9
- forval i = 1/`.names.arnels' {
= forval i = 1/1 {
- local namelist `namelist' `'.names[`i']'
= local namelist cluster
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist ``cluster''
- class exit ```namelist'''
= class exit ``cluster''
-----
> ----- end _optlist.dumpnames -----
= local ARGOPTNAMES ``cluster''
- local AND : list OPTNAMES & ARGOPTNAMES
- if `:list sizeof AND' {
= if 0 {
di as err "vctype ``:word 1 of `AND'' specified twice"
exit 198
}

```

```

- local pwallowed : list uniq pwallowed
- local pwlist robust cluster
- foreach n of local pwallowed {
  local lc = lower("`n'")
  local pwlist `pwlist' `lc'
}
- if "`old_is_ok'" != "" {
= if "old_is_ok" != "" {
- local ROBUST robust
- if `:list ROBUST in OPTNAMES' {
= if 1 {
- local robust Robust
- }
- local CLUSTER cluster
- if `:list CLUSTER in ARGOPTNAMES' {
= if 1 {
- local cluster CLuster(varname)
- }
- }
- local 0 ``user'""
= local 0 `` , vce(robust) robust""
- syntax [fw aw pw iw] [, VCE(string asis) `robust' `cluster' ]
= syntax [fw aw pw iw] [, VCE(string asis) Robust CLuster(varname)

> ]

- if ``vce'"" != "" {
= if ``robust'"" != "" {
- gettoken name args : vce, parse(" ,")
- local args : list retok args
- capture ParseVCE `o' `a' , `name'
= capture ParseVCE __00000D __00000E , robust

> ----- begin _vce_parse.ParseVCE -----
- gettoken 0 0 : 0, parse(" ,")
- gettoken A 0 : 0, parse(" ,")
- local OPTSPECS ```.0'.dumpoptions' `.`A'.dumpoptions'""

> ----- begin _optlist.dumpoptions -----
- version 9
- forval i = 1/`.options.arnels' {
= forval i = 1/3 {
- local optlist `optlist' `.options['i']'
= local optlist OIM
- }
- local optlist `optlist' `.options['i']'
= local optlist OIM OPG
- }

```

```

- local optlist `optlist' `.options['i']'
= local optlist OIM OPG Robust
- }
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist ``OIM OPG Robust''
- class exit ```optlist'''
= class exit ``OIM OPG Robust''

> _____ end _optlist.dumpoptions _____
> _____ begin _optlist.dumpoptions _____
- version 9
- forval i = 1/`.options.arrnells' {
= forval i = 1/1 {
- local optlist `optlist' `.options['i']'
= local optlist CLuster
- }
- local optlist : list uniq optlist
- sreturn local optlist ```optlist'''
= sreturn local optlist ``CLuster''
- class exit ```optlist'''
= class exit ``CLuster''

> _____ end _optlist.dumpoptions _____
= local OPTSPECS ``OIM OPG Robust CLuster''
- local OPTNAMES ```. `0'.dumpnames' `.`A'.dumpnames'''

> _____ begin _optlist.dumpnames _____
- version 9
- forval i = 1/`.names.arrnells' {
= forval i = 1/3 {
- local namelist `namelist' `.names['i']'
= local namelist oim
- }
- local namelist `namelist' `.names['i']'
= local namelist oim opg
- }
- local namelist `namelist' `.names['i']'
= local namelist oim opg robust
- }
- local namelist : list uniq namelist
- sreturn local namelist ```namelist'''
= sreturn local namelist ``oim opg robust''
- class exit ```namelist'''
= class exit ``oim opg robust''

```

```

> _____ end _optlist.dumpnames _____
> _____ begin _optlist.dumpnames _____
    - version 9
    - forval i = 1/`.names.arnels' {
    = forval i = 1/1 {
    - local namelist `namelist' `.names[`i']'
    = local namelist cluster
    - }
    - local namelist : list uniq namelist
    - sreturn local namelist ``namelist'''
    = sreturn local namelist "cluster"
    - class exit ``namelist'''
    = class exit "cluster"
> _____ end _optlist.dumpnames _____
    = local OPTNAMES "oim opg robust cluster"
    - syntax [, `OPTSPECS']
    = syntax [, OIM OPG Robust CLuster]
    - foreach OPT of local OPTNAMES {
    - if ````OPT'''' != "" {
    = if ``'' != "" {
      return local vce `OPT'
      continue, break
    }
    - }
    - if ````OPT'''' != "" {
    = if ``'' != "" {
      return local vce `OPT'
      continue, break
    }
    - }
    - if ````OPT'''' != "" {
    = if "robust" != "" {
      - return local vce `OPT'
      = return local vce robust
      - continue, break
      }
    }
> _____ end _vce_parse.ParseVCE _____
    - if c(rc) {
      di as err "vcetype {bf:`name'} not allowed"
      exit 198
    }

```



```

- local name `r(vce)'
= local name robust
- if "`weight'" == "pweight" & !`:list name in pwlist' {
= if "" == "pweight" & !1 {
    di as err "pweights are not allowed with vce(`name')"
    exit 198
}
- return add
- if ```robust'" != "" & !inlist("`name'", "", "robust", "eim", "oim"
> , "cluster") {
= if "`robust'" != "" & !inlist("robust", "", "robust", "eim", "oim", "
> cluster") {
    opts_exclusive "`robust' vce(`name')"
}
- if "`name'" == "cluster" {
= if "robust" == "cluster" {
    if ```args'" == "" {
    di as err "invalid vce(cluster) option"
    exit 198
}
    confirm var `args'
    unab args : `args'
    local markout `args'
    if "`cluster'" != "" {
    if !`:list args == cluster' {
    di as err "options cluster() and vce(cluster) are in conflict"
    exit 100
}
}
}
- else if "`cluster'" != "" & !inlist("`name'", "eim", "oim") {
= else if "" != "" & !inlist("robust", "eim", "oim") {
    opts_exclusive "cluster() vce(`name')"
}
- if ```args'" != "" {
= if `""' != "" {
    if !`:list name in ARGOPTNAMES' {
    di as err "invalid vce() option"
    exit 198
}
    if "`name'" == "hc2" {
    confirm var `args'
    unab args : `args'
    local markout `args'
}
    return local vceargs ```:list retok args'''

```

```

    }
- else if `:list name in ARGOPTNAMES' {
= else if 0 {
    if "`noargopt_is_ok'" == "" {
        di as err "invalid vce() option"
        exit 198
    }
}
- }
- else {
    if "`weight'" == "pweight" | "`robust'" != "" {
        local vce robust
    }
    if "`cluster'" != "" {
        local vce cluster
        local args `cluster'
        local markout `cluster'
    }
    return local vce ``vce'''
    return local vceargs ``args'''
}
- if "`return(vce)'" != "" {
= if "robust" != "" {
- if ``return(vceargs)''' != "" {
= if `'' != "" {
    local ss " "
}
- return local vceopt `"vce(`return(vce)''ss`return(vceargs)')'"
= return local vceopt `"vce(robust)'"
- }
- if "`return(vce)'" == "cluster" {
= if "robust" == "cluster" {
    return local cluster `return(vceargs)'
}
- if inlist("`return(vce)'", "robust", "cluster") {
= if inlist("robust", "robust", "cluster") {
- return local robust robust
- }
- if "`touse'" != "" {
= if "" != "" {
    if "`markout'" != "" {
        markout `touse' `markout', strok
    }
}
}

```

---

> \_\_\_\_\_ end \_vce\_parse \_\_\_\_\_

```

- local vce `r(vce)'
= local vce robust
- }
- else if "`weight'" == "pweight" {
= else if "" == "pweight" {
    local vce robust
  }
- if `:length local svy' {
= if 0 {
    local vce svy
  }
- if ``"vce"' == "cluster" {
= if `"robust"' == "cluster" {
    local cluster `r(cluster)'
    markout $ML_sample `cluster', strok
  }
- if `:length local group' {
= if 0 {
    markout $ML_sample `group', strok
    global ML_group : copy local group
    if bsubstr("`type `group'",1,3) == "str" {
      if `interactive' {
        SetGlobalVar ML_grp : _MLgr
      }
      else {
        global ML_grp : tempvar
        if ``moptobj"' != "" {
          global ML_grp `moptobj'$ML_grp
        }
      }
      quietly egen $ML_grp = group(`group')
      if ! `:length local cluster' {
        local cluster : copy global ML_grp
      }
    }
    else {
      if ! `:length local cluster' {
        local cluster : copy global ML_group
      }
    }
  }
- if "$ML_samp" != "$ML_sample" {
= if "__000006" != "__000006" {
    quietly replace $ML_samp = 0 if $ML_samp & $ML_sample == 0
  }
- if `:length local vce' == 0 {

```

```

= if 6 == 0 {
  if ``technique''' == "bhhh" {
    local vce opg
  }
  else local vce oim
}
- if inlist("`vce'", "robust", "cluster") & bsubstr("`evaltype'",1,1
> ) == "d" {
= if inlist("robust", "robust", "cluster") & bsubstr("e2",1,1) == "d
> " {
  local optname = cond("`svy'"=="", "vce(`vce')", "svy")
  di as err "option `optname' is not allowed with evaltype `evaltype
> '"

  exit 198
}
- if `:length local crittype' == 0 {
= if 0 == 0 {
- if inlist("`vce'", "cluster", "robust", "svy") {
= if inlist("robust", "cluster", "robust", "svy") {
- local crittype "log pseudolikelihood"
- }
- else local crittype "log likelihood"
- }
- global ML_waldtest `waldtest'
= global ML_waldtest -1
- if `:length local svy' | `:length local weight' {
= if 0 | 0 {
  if `interactive' {
    SetGlobalVar ML_w : _MLw
  }
  else {
    global ML_w : tempvar
    if ``moptobj''' != "" {
      global ML_w `moptobj'$ML_w
    }
  }
}
- if `:length local weight' {
= if 0 {
  if ``moptobj''' != "" {
    cap drop $ML_w
  }
  quietly gen double $ML_w `exp' if $ML_samp
  local wtype ``weight'''
  local wexp ``exp'''
  global ML_wtyp : copy local wtype

```

```

global ML_wexp : copy local wexp
if "`wtype'" == "iweight" | "`svy'" != "" {
  local zeroweight zeroweight
}
}
- if inlist("`vce'", "svy", "cluster") {
= if inlist("robust", "svy", "cluster") {
  if `interactive' {
    SetGlobalVar ML_subv : _MLsu
  }
  else {
    global ML_subv : tempvar
    if ``moptobj' != "" {
      global ML_subv `moptobj'$ML_subv
    }
  }
}
- quietly count if $ML_sample
= quietly count if __000006
- if r(N) == 0 {
  error 2000
}
- if ``weight' != "" {
= if `'' != "" {
  quietly count if $ML_sample & $ML_w != 0
  if r(N) == 0 {
    di in gr "all observation have zero weights"
    exit 2000
  }
}
- if `:length local lf0' {
= if 20 {
- if `:list sizeof lf0' != 2 {
= if 2 != 2 {
  di as err "lf0(`lf0') invalid"
  exit 198
}
- global ML_k0 : word 1 of `lf0'
= global ML_k0 : word 1 of 2 -414.2287919313009
- global ML_ll_0 : word 2 of `lf0'
= global ML_ll_0 : word 2 of 2 -414.2287919313009
- confirm integer number $ML_k0
= confirm integer number 2
- confirm number $ML_ll_0
= confirm number -414.2287919313009
- if $ML_k0 < 0 {

```

```

= if 2 < 0 {
  di as err "lf0(`lf0') invalid"
  exit 198
}
- }
- if "`obs'" == "" {
= if "" == "" {
- if "`wtype'" != "fweight" {
= if "" != "fweight" {
- quietly count if $ML_sample
= quietly count if __000006
- local obs = r(N)
- }
- else {
  sum $ML_w if $ML_sample, mean
  local obs = r(sum)
}
- }
- if `:length local iterid' {
= if 0 {
  local iterid = trim(bsubstr("`iterid'",1,32))
}
- if `:length local crittype' {
= if 20 {
- local crittype = trim(bsubstr("`crittype'",1,32))
= local crittype = trim(bsubstr("`log pseudolikelihood'",1,32))
- }
- if `:length local derivscale' {
= if 0 {
  confirm matrix `derivscale'
}
- if `:length local derivh' {
= if 0 {
  confirm matrix `derivh'
}
- local l : strlen local deriv
- ParseUsemin, `deriv' len(`l')
= ParseUsemin, len(0)

```

---

```

> ----- begin mopt.ParseUsemin -----
- capture syntax [, LEN(integer 0), MIN1(numlist min=2 max=2 missi
> ngokay) NOMIN MIN]
- if (_rc) {
  syntax [, LEN(integer 0) MIN1(string) NOMIN MIN]
}
- if ``min1'' != "" & ``nomin'' != "" {

```

```

= if `""` != "" & `""` != "" {
  di as err "options {bf:min()} and {bf:nomin} are mutually exclus
> ive"
  exit 198
}
- if `""`min`""` != "" & `""`nomin`""` != "" {
= if `""` != "" & `""` != "" {
  di as err "options {bf:min} and {bf:nomin} are mutually exclusiv
> e"
  exit 198
}
- if (`""`min1`""` != "") {
= if (`""` != "") {
  if (_rc) {
    capture confirm matrix `min1'
    if (_rc) {
      di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
      exit 198
    }
  }
  local deriv `min1'
}
- else if (`""`min`""` != "") {
= else if (`""` != "") {
  if (`len' > length(`""`min`""`)) {
    di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
    exit 198
  }
  local deriv 1
}
- else {
- if (`len' > 0) {
= if (0 > 0) {
  if (`nomin' == "") {
    di as err "option {bf:min()} requires that either two numbers or
> a 1X2 matrix be specified"
    exit 198
  }
  else if (`len' > length(`""`nomin`""`)) {
    di as err "options {bf:min()} and {bf:nomin} are mutually exclus
> ive"
    exit 198
  }
  else {

```

```

        local deriv 0
        }
        }
    - else {
    - local deriv 0
    - }
    - }
    - sreturn local deriv `deriv'
    = sreturn local deriv 0

```

---

```

> ----- end mopt.ParseUsemin -----
    - local deriv = s(deriv)
    - if `:length local gnwmatrix' {
    = if 0 {
        confirm matrix `gnwmatrix'
    }
    - `vv' mata: Mopt_model()
    = version 18: mata: Mopt_model()

```

---

```

> ----- begin mopt_check_program -----
    - version 10
    - args name
    - capture confirm name `name'
    = capture confirm name nbreg_lf
    - if c(rc) {
        di as err "unrecognized command:  `name' invalid command name"
        exit 199
    }
    - capture which `name'
    = capture which nbreg_lf
    - if !c(rc) {
    - exit

```

---

```

> ----- end mopt_check_program -----
    - global ML_stat model
    - if !`interactive' {
    = if !0 {
    - `vv' MaxMin `maxmin', nooutput `score' `options' moptobj(`moptobj'
> )
    = version 18: MaxMin maximize, nooutput  search(off) moptobj()

```

---

```

> ----- begin mopt.MaxMin -----
    - local vv : di "version " string(_caller()) ":"
    - version 12
    - ChkMacros

```

---



```

> ----- begin mopt.ChkMacros -----
    - if "$ML_stat" != "model" {
= if "model" != "model" {
    di as err "you must issue -ml model- first"
    exit 198
    }
    - capture confirm var $ML_samp $ML_w $ML_subv
= capture confirm var __000006 __000006
    - if c(rc) {
    di as err "Since issuing the -ml model- statement, you have do
> ne something to drop" _n "temporary variables ml added to your data. You mu
> st start again."
    Clear
    exit 111
    }
}

> ----- end mopt.ChkMacros -----
    - syntax name(name=maxmin) [, NEGH TOLerance(numlist max=1 >=0) LT
> OLerance(numlist max=1 >=0) NONRTOLerance NRTOLerance(numlist max=1 >=0) QTO
> Lerance(numlist max=1 >=0) GTOLerance(numlist max=1 >=0) ITERate(numlist max
> =1 >=0) NOTCONCAVE(numlist max=1 >=0) ndami DIFficult HALFSTEPonly DOOPT SEA
> rch(name) Repeat(passthru) Bounds(string) SCore(string) noWARNing noCLEAR NO
> LOg LOg DOTs SHOWTOLerance SHOWNRtolerance SHOWSTEP TRace COEFDIffs GRADient
> HESSian noSKIpline noOUTput Level(cilevel) moptobj(string) * ]
    - if `:length local nrtolerance' {
= if 0 {
    opts_exclusive "nrtolerance() `nonrtolerance'"
    }
    - if `:length local qtolerance' {
= if 0 {
    opts_exclusive "qtolerance() `nonrtolerance'"
    }
    - opts_exclusive "`nonrtolerance' `shownrtolerance'"
= opts_exclusive " "
}

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'`` != "" {
= while `"" != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- if `:length local shownrtolerance' {
= if 0 {
    local showtolerance showtolerance
}
- if "$ML_preserve" != "no" {
= if "no" != "no" {
    if `:length local score' {
        di as smcl as err "May not specify {bf:score()} option unless" _
> n "      a) estimation subsample is the entire data in memory, or" _n "      b
> ) you specify option {bf:nopreserve} on {bf:ml model} statement (meaning" _n
> "      your evaluation program explicitly restricts itself to obs" _n "
> for which {bf:$}" "{bf:ML_samp==1}."
        exit 198
    }
    preserve
    local N = c(N)
    quietly keep if $ML_sample
    if (`N' != c(N)) {
        mata: Mopt_init_regetviews()
    }
}
- if `:length local score' {
= if 0 {
    SetupScore ``$ML_evaltype"' ``score"' $ML_n $ML_dim
    local scores ``s(scores)'"
}
- _parse_iterlog, `nolog' `log'
= _parse_iterlog,

> ----- begin _parse_iterlog -----
- version 16
- syntax [, NOLOG LOG]
- if "`log'`nolog'" != "" {
= if "" != "" {
    opts_exclusive "`log' `nolog'"
}
- else if "`c(iterlog)'" == "on" {
= else if "on" == "on" {
    local log log
- }
- else {

```

```

        local nolog nolog
    }
    - sreturn local log `log'
    = sreturn local log log
    - sreturn local nolog `nolog'
    = sreturn local nolog

```

---

```

> ----- end _parse_iterlog -----
    - local log ``s(nolog)''
    = local log ""
    - local tropts `log' `dots' `showtolerance' `showstep' `trace' `co
> efdiffs'
    = local tropts
    - if ``tropts'' != "nolog" {
    = if "" != "nolog" {
    - local tropts `tropts' `gradient' `hessian'
    = local tropts
    - if ``skipline'' == "" {
    = if "" == "" {
    - if ``iterate'' != "0" | `:length local tropts' {
    = if "" != "0" | 0 {
    - di

    - }
    - }
    - }
    - else {
        local gradient
        local hessian
        local nolog nolog
    }
    - if ``search'' != "off" & ``iterate'' != "0" {
    = if "off" != "off" & "" != "0" {
        if ``search'' == "quietly" {
            local sopts nolog
        }
        else {
            if `:length local tropts' {
                if ``tropts'' != "nolog" {
                    local sopts trace
                }
            }
            else local sopts nolog
        }
    }
    if ``search'' == "norescale" {
        local sopts `sopts' norescale
    }

```

```

    }
    Search `bounds', nopreserve `sopts' `repeat' `maxmin'
  }
- tempname value b V iV
- `vv' mata: Mopt_maxmin()
= version 18: mata: Mopt_maxmin()

> ----- begin nbreg_lf -----
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __000000==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000Y = exp(__00000U) if __000006
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
- if `lnalpha'<-20 {
= if __00000V<-20 {
    replace `sc1' = `y'-`mu' if $ML_samp
    replace `sc2' = 0 if $ML_samp
  }
- else {

```

```

- replace `sc1' = `p'*(`y'-'mu') if $ML_samp
= replace __00000R = __00000Z*(y_timing_max-__00000Y) if __00000
> 6
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p')
> ) + `p'*(`y'-'mu') if $ML_samp
= replace __00000S = __00000W*(digamma(__00000W) - digamma(y_tim
> ing_max+__00000W) - ln(__00000Z)) + __00000Z*(y_timing_max-__00000Y) if __00
> 0006
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __000000 = __00000R, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __000000 = __00000S, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000P = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __000000==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __000010 = exp(__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
mlmatsum `lnf' `d11' = `mu', eq(1)
mlmatsum `lnf' `d12' = 0, eq(1,2)
mlmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mlmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-'mu')+1), eq
> (1)
= mlmatsum __000000 __000011 = __00000Y*__00000Z*(__000010*__000
> 00Z*(y_timing_max-__00000Y)+1), eq(1)
- mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-'mu'), eq(1,2)
= mlmatsum __000000 __000012 = __000010*__00000Y*__00000Z^2*(y_t
> iming_max-__00000Y), eq(1,2)
- mlmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*(
> `y'-'mu')- 1), eq(2)
= mlmatsum __000000 __000013 = __00000W*(digamma(__00000W) - dig
> amma(y_timing_max+__00000W) - ln(__00000Z) - __00000W*(trigamma(y_timing_max
> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
- }
- matrix `H' = (`d11',`d12' \ `d12'',`d22')

```

```

= matrix __00000Q = (__000011,__000012 \ __000012',__000013)
> ----- end nbreg_lf -----
Iteration 0: Log pseudolikelihood = -372.54415
> ----- begin nbreg_lf -----
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
mlsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsum __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __000000==. { exit }
> ----- end nbreg_lf -----
> ----- begin nbreg_lf -----
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')

```

```

= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 0 == 0 | __000000==. { exit }
}

> _____ end nbreg_lf _____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 2 == 0 | __000000==. { exit }
- tempname alpha
- tempvar mu p
- quietly {

```

```

- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000Y = exp(__00000U) if __000006
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
- if `lnalpha'<-20 {
= if __00000V<-20 {
    replace `sc1' = `y'-`mu' if $ML_samp
    replace `sc2' = 0 if $ML_samp
  }
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000R = __00000Z*(y_timing_max-__00000Y) if __00000
> 6
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
= replace __00000S = __00000W*(digamma(__00000W) - digamma(y_tim
> ing_max+__00000W) - ln(__00000Z)) + __00000Z*(y_timing_max-__00000Y) if __00
> 0006
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __000000 = __00000R, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __000000 = __00000S, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000P = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __000000==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __000010 = exp(__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlmatsum `lnf' `d11' = `mu', eq(1)
    mlmatsum `lnf' `d12' = 0, eq(1,2)
    mlmatsum `lnf' `d22' = 0, eq(2)
  }
- else {
- mlmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
= mlmatsum __000000 __000011 = __00000Y*__00000Z*(__000010*__000
> 00Z*(y_timing_max-__00000Y)+1), eq(1)
- mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
= mlmatsum __000000 __000012 = __000010*__00000Y*__00000Z^2*(y_t

```



```

> iming_max-__00000Y), eq(1,2)
      - mلماتsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*(
> `y'-`mu')- 1), eq(2)
      = mلماتsum __000000 __000013 = __00000W*(digamma(__00000W) - dig
> amma(y_timing_max+__00000W) - ln(__00000Z) - __00000W*(trigamma(y_timing_max
> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
      - }
      - matrix `H' = (`d11',`d12' \ `d12'',`d22')
      = matrix __00000Q = (__000011,__000012 \ __000012',__000013)

> _____ end nbreg_lf _____
Iteration 1:  Log pseudolikelihood = -362.08158

> _____ begin nbreg_lf _____

      - version 6.0
      - args todo b lnf g H sc1 sc2
      - tempvar xb
      - tempname lnalpha m
      - mlevel `xb' = `b', eq(1)
      = mlevel __00000U = __00000N, eq(1)
      - mlevel `lnalpha' = `b', eq(2) scalar
      = mlevel __00000V = __00000N, eq(2) scalar
      - local y "$ML_y1"
      = local y "y_timing_max"
      - scalar `m' = exp(-`lnalpha')
      = scalar __00000W = exp(-__00000V)
      - if `lnalpha' < -20 {
      = if __00000V < -20 {
          mلماتsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
      }
      - else {
      - mلماتsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
      = mلماتsum __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
      - }
      - if `todo' == 0 | `lnf'==. { exit }
      = if 0 == 0 | __000000==. { exit }

> _____ end nbreg_lf _____

> _____ begin nbreg_lf _____

      - version 6.0

```

```

- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __000000==. { exit }
}

> _____ end nbreg_lf _____

> _____ begin nbreg_lf _____

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')

```

```

> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
      = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
      - }
      - if `todo' == 0 | `lnf'==. { exit }
      = if 2 == 0 | __000000==. { exit }
      - tempname alpha
      - tempvar mu p
      - quietly {
      - gen double `mu' = exp(`xb') if $ML_samp
      = gen double __00000Y = exp(__00000U) if __000006
      - gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
      = gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
      - if `lnalpha'<-20 {
      = if __00000V<-20 {
          replace `sc1' = `y'-`mu' if $ML_samp
          replace `sc2' = 0 if $ML_samp
        }
      - else {
      - replace `sc1' = `p'*(`y'-`mu') if $ML_samp
      = replace __00000R = __00000Z*(y_timing_max-__00000Y) if __00000
> 6
      - replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
      = replace __00000S = __00000W*(digamma(__00000W) - digamma(y_tim
> ing_max+__00000W) - ln(__00000Z)) + __00000Z*(y_timing_max-__00000Y) if __00
> 0006
      - }
      - }
      - $ML_ec tempname g1 g2
      = * tempname g1 g2
      - $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
      = * mlvecsum __000000 = __00000R, eq(1)
      - $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
      = * mlvecsum __000000 = __00000S, eq(2)
      - $ML_ec matrix `g' = (`g1',`g2')
      = * matrix __00000P = (,)
      - if `todo' == 1 | `lnf'==. { exit }
      = if 2 == 1 | __000000==. { exit }
      - tempname alpha d11 d12 d22
      - scalar `alpha' = exp(`lnalpha')
      = scalar __000010 = exp(__00000V)
      - if `lnalpha' < -20 {
      = if __00000V < -20 {
          mlmatsum `lnf' `d11' = `mu', eq(1)

```

```

        mllmatsum `lnf' `d12' = 0, eq(1,2)
        mllmatsum `lnf' `d22' = 0, eq(2)
    }
    - else {
    - mllmatsum `lnf' `d11' = `mu'*`p*(`alpha'*`p*(`y'-`mu')+1), eq
> (1)
        = mllmatsum __000000 __000011 = __00000Y*__00000Z*(__000010*__000
> 00Z*(y_timing_max-__00000Y)+1), eq(1)
        - mllmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
        = mllmatsum __000000 __000012 = __000010*__00000Y*__00000Z^2*(y_t
> iming_max-__00000Y), eq(1,2)
        - mllmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p*(`alpha'*`p*(
> `y'-`mu')- 1), eq(2)
        = mllmatsum __000000 __000013 = __00000W*(digamma(__00000W) - dig
> amma(y_timing_max+__00000W) - ln(__00000Z) - __00000W*(trigamma(y_timing_max
> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
        - }
        - matrix `H' = (`d11',`d12' \ `d12'',`d22')
        = matrix __00000Q = (__000011,__000012 \ __000012',__000013)
    }
}
> _____ end nbreg_lf _____
Iteration 2: Log pseudolikelihood = -360.87234
> _____ begin nbreg_lf _____
    - version 6.0
    - args todo b lnf g H sc1 sc2
    - tempvar xb
    - tempname lnalpha m
    - mlevel `xb' = `b', eq(1)
    = mlevel __00000U = __00000N, eq(1)
    - mlevel `lnalpha' = `b', eq(2) scalar
    = mlevel __00000V = __00000N, eq(2) scalar
    - local y "$ML_y1"
    = local y "y_timing_max"
    - scalar `m' = exp(-`lnalpha')
    = scalar __00000W = exp(-__00000V)
    - if `lnalpha' < -20 {
    = if __00000V < -20 {
        mllsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
    - else {
        - mllsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
        = mllsum __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti

```

```

> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 0 == 0 | __000000==. { exit }
    _____
> _____ end nbreg_lf _____
    _____
> _____ begin nbreg_lf _____
    - version 6.0
    - args todo b lnf g H sc1 sc2
    - tempvar xb
    - tempname lnalpha m
    - mlevel `xb' = `b', eq(1)
    = mlevel __00000U = __00000N, eq(1)
    - mlevel `lnalpha' = `b', eq(2) scalar
    = mlevel __00000V = __00000N, eq(2) scalar
    - local y "$ML_y1"
    = local y "y_timing_max"
    - scalar `m' = exp(-`lnalpha')
    = scalar __00000W = exp(-__00000V)
    - if `lnalpha' < -20 {
    = if __00000V < -20 {
        mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
    - else {
        - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
        = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 0 == 0 | __000000==. { exit }
    _____
> _____ end nbreg_lf _____
    _____
> _____ begin nbreg_lf _____
    - version 6.0
    - args todo b lnf g H sc1 sc2
    - tempvar xb
    - tempname lnalpha m
    - mlevel `xb' = `b', eq(1)
    = mlevel __00000U = __00000N, eq(1)
    - mlevel `lnalpha' = `b', eq(2) scalar
    = mlevel __00000V = __00000N, eq(2) scalar

```

```

- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __000000==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000Y = exp(__00000U) if __000006
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
- if `lnalpha'<-20 {
= if __00000V<-20 {
    replace `sc1' = `y'-`mu' if $ML_samp
    replace `sc2' = 0 if $ML_samp
  }
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000R = __00000Z*(y_timing_max-__00000Y) if __00000
> 6
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
    = replace __00000S = __00000W*(digamma(__00000W) - digamma(y_tim
> ing_max+__00000W) - ln(__00000Z)) + __00000Z*(y_timing_max-__00000Y) if __00
> 0006
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __000000 = __00000R, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __000000 = __00000S, eq(2)

```

```

- $ML_ec matrix `g' = (`g1', `g2')
= * matrix __00000P = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __000000==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __000010 = exp(__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
  mpmatsum `lnf' `d11' = `mu', eq(1)
  mpmatsum `lnf' `d12' = 0, eq(1,2)
  mpmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mpmatsum `lnf' `d11' = `mu'*`p'*(`alpha'*`p'*(`y'-`mu')+1), eq
> (1)
= mpmatsum __000000 __000011 = __00000Y*__00000Z*(__000010*__000
> 00Z*(y_timing_max-__00000Y)+1), eq(1)
- mpmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
= mpmatsum __000000 __000012 = __000010*__00000Y*__00000Z^2*(y_t
> iming_max-__00000Y), eq(1,2)
- mpmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p'*(`alpha'*`p'*
> `y'-`mu')- 1), eq(2)
= mpmatsum __000000 __000013 = __00000W*(digamma(__00000W) - dig
> amma(y_timing_max+__00000W) - ln(__00000Z) - __00000W*(trigamma(y_timing_max
> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
- }
- matrix `H' = (`d11', `d12' \ `d12'', `d22')
= matrix __00000Q = (__000011, __000012 \ __000012', __000013)

> _____ end nbreg_lf _____

Iteration 3: Log pseudolikelihood = -360.83675

> _____ begin nbreg_lf _____

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"

```

```

- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __000000==. { exit }

_____
> _____ end nbreg_lf _____
_____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
    }
- else {
    - mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __000000==. { exit }

_____
> _____ end nbreg_lf _____

```



---

```

> begin nbreg_lf
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
mlsum `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsum `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsum __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __000000==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000Y = exp(__00000U) if __000006
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
- if `lnalpha'<-20 {
= if __00000V<-20 {
replace `sc1' = `y'-`mu' if $ML_samp
replace `sc2' = 0 if $ML_samp
}
- else {
- replace `sc1' = `p'*(`y'-`mu') if $ML_samp
= replace __00000R = __00000Z*(y_timing_max-__00000Y) if __00000
> 6
- replace `sc2' = `m'*(digamma(`m') - digamma(`y'+`m') - ln(`p'))
> ) + `p'*(`y'-`mu') if $ML_samp
= replace __00000S = __00000W*(digamma(__00000W) - digamma(y_tim

```

```

> ing_max+__00000W) - ln(__00000Z)) + __00000Z*(y_timing_max-__00000Y) if __00
> 0006
- }
- }
- $ML_ec tempname g1 g2
= * tempname g1 g2
- $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
= * mlvecsum __000000 = __00000R, eq(1)
- $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
= * mlvecsum __000000 = __00000S, eq(2)
- $ML_ec matrix `g' = (`g1',`g2')
= * matrix __00000P = (,)
- if `todo' == 1 | `lnf'==. { exit }
= if 2 == 1 | __000000==. { exit }
- tempname alpha d11 d12 d22
- scalar `alpha' = exp(`lnalpha')
= scalar __000010 = exp(__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
  mlmatsum `lnf' `d11' = `mu', eq(1)
  mlmatsum `lnf' `d12' = 0, eq(1,2)
  mlmatsum `lnf' `d22' = 0, eq(2)
}
- else {
- mlmatsum `lnf' `d11' = `mu'*`p*(`alpha'*`p*(`y'-`mu')+1), eq
> (1)
= mlmatsum __000000 __000011 = __00000Y*__00000Z*(__000010*__000
> 00Z*(y_timing_max-__00000Y)+1), eq(1)
- mlmatsum `lnf' `d12' = `alpha'*`mu'*`p'^2*(`y'-`mu'), eq(1,2)
= mlmatsum __000000 __000012 = __000010*__00000Y*__00000Z^2*(y_t
> iming_max-__00000Y), eq(1,2)
- mlmatsum `lnf' `d22' = `m'*(digamma(`m') - digamma(`y'+`m') -
> ln(`p') - `m'*(trigamma(`y'+`m') - trigamma(`m')))) + `mu'*`p*(`alpha'*`p*(`
> `y'-`mu')- 1), eq(2)
= mlmatsum __000000 __000013 = __00000W*(digamma(__00000W) - dig
> amma(y_timing_max+__00000W) - ln(__00000Z) - __00000W*(trigamma(y_timing_max
> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
- }
- matrix `H' = (`d11',`d12' \ `d12'',`d22')
= matrix __00000Q = (__000011,__000012 \ __000012',__000013)
_____
> _____ end nbreg_lf _____
Iteration 4: Log pseudolikelihood = -360.83656
_____
> _____ begin nbreg_lf _____

```

```

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lgamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {
    - mlsun `lnf' = lgamma(`m'+`y') - lgamma(`y'+1) - lgamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
    = mlsun __000000 = lgamma(__00000W+y_timing_max) - lgamma(y_ti
> ming_max+1) - lgamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
    - }
    - if `todo' == 0 | `lnf'==. { exit }
    = if 0 == 0 | __000000==. { exit }
}

> _____ end nbreg_lf _____

> _____ begin nbreg_lf _____

- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
    mlsun `lnf' = -lgamma(`y'+1) - exp(`xb') + `y'*`xb'
  }
- else {

```

```

- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 0 == 0 | __000000==. { exit }

> _____ end nbreg_lf _____
> _____ begin nbreg_lf _____
- version 6.0
- args todo b lnf g H sc1 sc2
- tempvar xb
- tempname lnalpha m
- mlevel `xb' = `b', eq(1)
= mlevel __00000U = __00000N, eq(1)
- mlevel `lnalpha' = `b', eq(2) scalar
= mlevel __00000V = __00000N, eq(2) scalar
- local y "$ML_y1"
= local y "y_timing_max"
- scalar `m' = exp(-`lnalpha')
= scalar __00000W = exp(-__00000V)
- if `lnalpha' < -20 {
= if __00000V < -20 {
mlsun `lnf' = -lngamma(`y'+1) - exp(`xb') + `y'*`xb'
}
- else {
- mlsun `lnf' = lngamma(`m'+`y') - lngamma(`y'+1) - lngamma(`m')
> - `m'*ln1p(exp(`xb'+`lnalpha')) - `y'*ln1p(exp(-`xb'-`lnalpha'))
= mlsun __000000 = lngamma(__00000W+y_timing_max) - lngamma(y_ti
> ming_max+1) - lngamma(__00000W) - __00000W*ln1p(exp(__00000U+__00000V)) - y_
> timing_max*ln1p(exp(-__00000U-__00000V))
- }
- if `todo' == 0 | `lnf'==. { exit }
= if 2 == 0 | __000000==. { exit }
- tempname alpha
- tempvar mu p
- quietly {
- gen double `mu' = exp(`xb') if $ML_samp
= gen double __00000Y = exp(__00000U) if __000006
- gen double `p' = 1/(1+exp(`xb'+`lnalpha')) if $ML_samp
= gen double __00000Z = 1/(1+exp(__00000U+__00000V)) if __000006
- if `lnalpha'<-20 {
= if __00000V<-20 {

```

```

        replace `sc1' = `y' - `mu' if $ML_samp
        replace `sc2' = 0 if $ML_samp
    }
    - else {
    - replace `sc1' = `p' * (`y' - `mu') if $ML_samp
    = replace __00000R = __00000Z * (y_timing_max - __00000Y) if __00000
> 6
        - replace `sc2' = `m' * (digamma(`m') - digamma(`y' + `m') - ln(`p'))
> ) + `p' * (`y' - `mu') if $ML_samp
        = replace __00000S = __00000W * (digamma(__00000W) - digamma(y_tim
> ing_max + __00000W) - ln(__00000Z)) + __00000Z * (y_timing_max - __00000Y) if __00
> 0006
    - }
    - }
    - $ML_ec tempname g1 g2
    = * tempname g1 g2
    - $ML_ec mlvecsum `lnf' `g1' = `sc1', eq(1)
    = * mlvecsum __000000 = __00000R, eq(1)
    - $ML_ec mlvecsum `lnf' `g2' = `sc2', eq(2)
    = * mlvecsum __000000 = __00000S, eq(2)
    - $ML_ec matrix `g' = (`g1', `g2')
    = * matrix __00000P = (,)
    - if `todo' == 1 | `lnf' == . { exit }
    = if 2 == 1 | __000000 == . { exit }
    - tempname alpha d11 d12 d22
    - scalar `alpha' = exp(`lnalpha')
    = scalar __000010 = exp(__00000V)
    - if `lnalpha' < -20 {
    = if __00000V < -20 {
        mlmatsum `lnf' `d11' = `mu', eq(1)
        mlmatsum `lnf' `d12' = 0, eq(1,2)
        mlmatsum `lnf' `d22' = 0, eq(2)
    }
    - else {
    - mlmatsum `lnf' `d11' = `mu' * `p' * (`alpha' * `p' * (`y' - `mu') + 1), eq
> (1)
        = mlmatsum __000000 __000011 = __00000Y * __00000Z * (__000010 * __000
> 00Z * (y_timing_max - __00000Y) + 1), eq(1)
        - mlmatsum `lnf' `d12' = `alpha' * `mu' * `p'^2 * (`y' - `mu'), eq(1,2)
        = mlmatsum __000000 __000012 = __000010 * __00000Y * __00000Z^2 * (y_t
> iming_max - __00000Y), eq(1,2)
        - mlmatsum `lnf' `d22' = `m' * (digamma(`m') - digamma(`y' + `m') -
> ln(`p')) - `m' * (trigamma(`y' + `m') - trigamma(`m')) + `mu' * `p' * (`alpha' * `p' *
> `y' - `mu') - 1), eq(2)
        = mlmatsum __000000 __000013 = __00000W * (digamma(__00000W) - dig
> amma(y_timing_max + __00000W) - ln(__00000Z) - __00000W * (trigamma(y_timing_max

```

```

> +__00000W) - trigamma(__00000W))) + __00000Y*__00000Z*(__000010*__00000Z*(y_
> timing_max-__00000Y)- 1), eq(2)
- }
- matrix `H' = (`d11', `d12' \ `d12'', `d22')
= matrix __00000Q = (__000011, __000012 \ __000012', __000013)
_____
> _____ end nbreg_lf _____
Iteration 5: Log pseudolikelihood = -360.83656
_____
> _____ begin mopt_post _____
- args b V C wgt touse depvar
- if `:length local touse' {
= if 8 {
- tempname esamp
- quietly gen byte `esamp' = `touse'
= quietly gen byte __00000W = __000006
- }
- ereturn post `b' `V' `C' `wtg', esample(`esamp') depname(`depv
> ar') buildfvinfo
= ereturn post __00000U __00000V [], esample(__00000W) depname(
> y_timing_max) buildfvinfo
_____
> _____ end mopt_post _____
- ereturn scalar k_eq_model = abs($ML_waldtest)
= ereturn scalar k_eq_model = abs(-1)
- if `:length global ML_ll_0' {
= if 18 {
- ereturn scalar ll_0 = $ML_ll_0
= ereturn scalar ll_0 = -414.2287919313009
- if `:length global ML_rank0' {
= if 1 {
- ereturn scalar rank0 = $ML_rank0
= ereturn scalar rank0 = 2
- ereturn scalar df_m = e(rank) - $ML_rank0
= ereturn scalar df_m = e(rank) - 2
- }
- else ereturn scalar df_m = e(rank) - $ML_k0
= else ereturn scalar df_m = e(rank) - 2
- }
- if !`:length global ML_ll_0' | inlist(e(vce), "robust", "cluster
> ", "linearized") {
= if !18 | inlist(e(vce), "robust", "cluster", "linearized") {
- if "$ML_svy" == "" {
= if "" == "" {
- _prefix_model_test, minimum
_____

```

```

> ----- begin _prefix_model_test -----
- version 9
- syntax [name(name=cmd)] [, svy noADJust noSVYadjust MINimum ]
- if (inlist("`cmd'", "asmixlogit", "cmmixlogit", "cmxtmixlogit"))
> {
= if (inlist("", "asmixlogit", "cmmixlogit", "cmxtmixlogit")) {
  _asmixtest, `svy' `adjust' `svyadjust'
  exit
}
- local K = 1
- if "`e(k_eq_model)'" != "" {
= if "1" != "" {
- capture confirm integer number `e(k_eq_model)'
= capture confirm integer number 1
- if ! c(rc) {
- local K = e(k_eq_model)
- }
- }
- if "`e(k_eq_model_skip)'" != "" {
= if "" != "" {
  capture confirm integer number `e(k_eq_model_skip)'
  if ! c(rc) {
    local skip = e(k_eq_model_skip)
  }
}
- ereturn local df_m
- ereturn local F
- ereturn local chi2
- ereturn local chi2type
- ereturn local p
- if (`K' == 0) {
= if (1 == 0) {
  exit
}
- forval i = 1/`K' {
= forval i = 1/1 {
- if !':list i in skip' {
= if !0 {
- local spec "`spec' ([#`i'])"
= local spec " ([#1])"
- }
- }
- tempname cns omit
- capture matrix `cns' = get(Cns)
= capture matrix __00000W = get(Cns)
- if (c(rc)) local cns

```

```

- if "`cns'" != "" {
= if "" != "" {
  local k_autoCns = e(k_autoCns)
  if missing(`k_autoCns') {
    local k_autoCns 0
  }
  if rowsof(`cns') == `k_autoCns' {
    local cns
  }
}
- if "`svy'" != "" & "`adjust'`svyadjust'" != "" {
= if "" != "" & "" != "" {
  local adjust nosvyadjust
}
- else local adjust
- _ms_omit_info e(b), nocons
- matrix `omit' = r(omit)
= matrix __00000X = r(omit)
- capture test `spec', `adjust'
= capture test ([#1]),

```

---

```

> ----- begin test -----
- if "`e(cmd)'" == "anova" | "`e(cmd)'" == "manova" {
= if "mopt" == "anova" | "mopt" == "manova" {
  AnovaTestChooser `0'
  return add
  exit
}
- if "`e(cmd)'" == "mixed" {
= if "mopt" == "mixed" {
  MixedTestChooser `0'
  return add
  exit
}
- if _caller() < 8 {
  _test `0'
  return add
  exit
}
- is_svy

```

---

```

> ----- begin is_svy -----
- version 8
- syntax [, Regression]
- if ``e(prefix)'' == "svy" {
= if `""' == "svy" {

```



```

        return scalar is_svy = 1
    }
- else if "`regression'" == "" {
= else if "" == "" {
- return scalar is_svy = ("`e(N_psu)'" != "")
= return scalar is_svy = (" " != " ")
- }
- else {
    capt mat list e(b)
    local rc = _rc
    return scalar is_svy = ("`e(N_psu)'" != "") & (`rc' == 0)
}
}

> ----- end is_svy -----

- local is_svy = r(is_svy)
- if replay() & `is_svy' == 0 {
= if replay() & 0 == 0 {
    _test `0'
    return add
    exit
}
- version 8
- WaldTest `is_svy' `0'
= WaldTest 0 ([#1]),

> ----- begin test.WaldTest -----

- gettoken is_svy 0 : 0
- if "`is_svy'" == "1" {
= if "0" == "1" {
    if "`e(complete)'" == "available" {
        di as err "`must run svy command with "complete" "" "optio
> n before using this command"
        exit 301
    }
}

- gettoken tok : 0, parse(" (,") bind
- local single single
- local zero ``0'
= local zero ` ([#1]), "
- if ``tok' == "(" {
= if `(' == "(" {
- local single
- while ``tok' == "(" {
= while `(' == "(" {
- gettoken tok 0 : 0, parse(" (,") match(paren) bind
- local eqns ``eqns' ``tok'

```

```

= local eqns `"'`[#1]''''
- gettoken tok : 0, parse(" (,") bind
- }
- while `"'`tok'''' == "(" {
= while `"','' == "(" {
  gettoken tok 0 : 0, parse(" (,") match(paren) bind
  local eqns `"'`eqns' `"'`tok''''
  gettoken tok : 0, parse(" (,") bind
  }
- if `"'`tok'''' != ",," & `"'`tok'''' != "" {
= if `"','' != ",," & `"','' != "" {
  local single single
  }
- }
- if "`single'"=="single" {
= if ""=="single" {
  local eqns
  local 0 `"'`zero''''
  while `"'`tok'''' != "" & `"'`tok'''' != ",," {
    gettoken tok 0 : 0, parse(" ,") bind
    local eqns `"'`eqns' `tok''''
    gettoken tok : 0, parse(" ,") bind
  }
  local eqns `"'`eqns''''
  }
- syntax [, Accumulate noSVYadjust common CONSTant MINimum M
> test Mtest2(passthru) noTest COEF matvlg(passthru) DF(numlist max=1 >0) ]
- if "`df'" != "" {
= if "" != "" {
  local dof = `df'
  }
- else if "`e(df_r)'" != "" {
= else if "" != "" {
  local dof = e(df_r)
  }
- else {
- local dof = .
- }
- if `"'`mtest'`mtest2'''' != "" {
= if `'''' != "" {
  if inlist("`e(cmd)'", "svymean", "svytotal", "svyratio") {
    di as err `"'`mtest option not allowed after `e(cmd)''''
    exit 301
  }
  _mtest syntax, `mtest' `mtest2'
  local mtmethod `r(method)'

```

```

    }
- if "`mtmethod'" != "" {
= if "" != "" {
    if "`test'" != "" {
        di as err "options mtest and notest may not be combined"
        exit 198
    }
    if "`accumulate'" != "" {
        di as err "options mtest and accumulate may not be combine
> d"

        exit 198
    }
}
- if `is_svy' {
= if 0 {
    if ``e(cmd)''' == "margins" {
        local vce ``e(model_vce)'''
    }
    else local vce ``e(vce)'''
    if inlist(```vce''' , "bootstrap", "sdr") {
        if missing(`dof') {
            local svyadjust nosvyadjust
        }
    }
}
- if !`is_svy' & "`svyadjust'" != "" {
= if !0 & "" != "" {
    di as err "option nosvyadjust is allowed only with svy dat
> a"

    exit 198
}
- local svy_adjust = `is_svy' & "`svyadjust'" == ""
= local svy_adjust = 0 & "" == ""
- if `svy_adjust' & missing(`dof') & (missing(e(N_psu)) | mi
> ssing(e(N_strata))) {
= if 0 & missing(.) & (missing(e(N_psu)) | missing(e(N_strat
> a))) {

    di as err "impossible to adjust p-values"
    exit 198
}
- if `is_svy' & ``test''' == "" {
= if 0 & `''' == "" {
    if "`svyadjust'" == "" {
        di _n as txt "Adjusted Wald test"
    }
    else di _n as txt "Unadjusted Wald test"

```

```

    }
    - local tmp : substr local eqns "==" "", all count(local
> nch)
    - if `nch' > 0 {
= if 0 > 0 {
    di as err "==" invalid operator"
    exit 198
    }
    - local eqns : substr local eqns "==" "=", all
    - ExpandExpns ``eqns'"" ``common'"" ``constant'""
= ExpandExpns `` "[#1]"" "" ""

> ----- begin test.ExpandExpns -----
    - args expns common constant
    - gettoken exp expns : expns
    - while ``exp'"" != "" {
= while ``[#1]"" != "" {
    - Expand ``exp'"" ``common'"" ``constant'""
= Expand ``[#1]"" "" ""

> ----- begin test.Expand -----
    - args exp common constant
    - local junk : substr local exp "=" "=", count(local n
> ch) all
    - if `nch' <= 1 {
= if 0 <= 1 {
    - return local explist ``exp'""
= return local explist ``[#1]""
    - return local nexp 1
    - exit

> ----- end test.Expand -----
    - local explist ``explist' `r(explist)''
= local explist `` "[#1]""
    - local nexp = `nexp' + `r(nexp)'
= local nexp = + 1
    - gettoken exp expns : expns
    - }
    - while ``exp'"" != "" {
= while ``'"" != "" {
    Expand ``exp'"" ``common'"" ``constant'""
    local explist ``explist' `r(explist)''
    local nexp = `nexp' + `r(nexp)'
    gettoken exp expns : expns
    }
    - return local explist ``explist'""

```

```

= return local explist ``"``[#1]""''
- return local nexp `nexp'
= return local nexp 1

```

---

```

> ----- end test.ExpandExps -----
- local explist ``"``r(explist)""''
= local explist ``"``[#1]""''
- local nexp `r(nexp)'
= local nexp 1
- local acc `accumulate'
= local acc
- foreach exp of local explist {
- quietly _test `exp' , `acc' `common' `constant' notest
= quietly _test [#1] , notest
- local acc acc
- }
- if ``test' == "notest" {
= if "" == "notest" {
_test , notest
exit
}
- quietly _test , `matvlc'
= quietly _test ,
- if `dof' < . {
= if . < . {
local testtype F
tempname F df1 df2 p drop
if ``r(F)' != "" {
scalar `F' = r(F)
}
else {
scalar `F' = r(chi2)/r(df)
}
scalar `df1' = r(df)
scalar `df2' = `dof'
scalar `p' = Ftail(`df1', `df2', `F')
scalar `drop' = r(drop)
}
- else {
- local testtype chi2
- tempname chi2 df1 p drop
- scalar `chi2' = r(chi2)
= scalar __00000Y = r(chi2)
- scalar `df1' = r(df)
= scalar __00000Z = r(df)
- scalar `p' = r(p)

```

```

= scalar __000010 = r(p)
- scalar `drop' = r(drop)
= scalar __000011 = r(drop)
- }
- local i 1
- while "`r(dropped_`i')'" != "" {
= while "" != "" {
    local dropped_`i' `r(dropped_`i')'
    local ++i
}
- if `svy_adjust' {
= if 0 {
    if "`df'" != "" {
        local svy_d = `df'
    }
    else if !missing(e(df_r)) {
        local svy_d = e(df_r)
    }
    else local svy_d = e(N_psu)-e(N_strata)
    if missing(`svy_d') {
        di as err "internal error in adjustment for svy"
        exit 10001
    }
    if "`testtype'" == "F" {
        scalar `F' = ((`svy_d'-`df1'+1)/`svy_d') * `F'
    }
    else {
        tempname F df2
        scalar `F' = ((`svy_d'-`df1'+1)/(`svy_d'*`df1')) * `chi2'
    }
    scalar `df2' = `svy_d' - `df1' + 1
    if `df2' < 0 {
        scalar `F' = .
    }
    scalar `p' = Ftail(`df1', `df2', `F')
    local testtype F
}
- if "`mtmethod'" != "" {
= if "" != "" {
    tempname R r Rr V b VR tm Wald
    mat `V' = e(V)
    mat `b' = e(b)
    mat `Rr' = get(Rr)
    local nb = rowsof(`V')
    local nr = rowsof(`Rr')
    mat `R' = `Rr'[1...,1..`nb']

```

```

mat `VR' = `R'*`V'*`R'
mat `r' = `R'*`b' - `Rr'[1...,`= `nb'+1']
mat `Wald' = `r'*syminv(`VR')*`r'
mat `tm' = J(`nr', 3, .)
mat colnames `tm' = `testtype' df p
forvalues it = 1 / `nr' {
mat `Wald' = `r'[`it',1]^2 / `VR'[`it',`it']
mat `tm'[`it',1] = `Wald'[1,1]
mat `tm'[`it',2] = 1
if `svy_adjust' {
mat `tm'[`it',3] = Ftail(1,`svy_d',`tm'[`it',1])
}
else if "`testtype'" == "chi2" {
mat `tm'[`it',3] = chi2tail(1,`tm'[`it',1])
}
else {
mat `tm'[`it',3] = Ftail(1,`df2',`tm'[`it',1])
}
}
if "`mtmethod'" != "noadjust" {
_mtest adjust `tm', mtest(`mtmethod') pindex(3) append
mat `tm' = r(result)
local pindex 4
}
else local pindex 3
}
- _test, notest
- local i 1
- while "`dropped_`i'" != "" {
= while "" != "" {
di as txt " Constraint `dropped_`i' dropped"
local ++i
}
- if "`mtmethod'" == "" {
= if "" == "" {
- di
- if "`testtype'" == "chi2" {
= if "chi2" == "chi2" {
- di as txt _col(12) "chi2(" %3.0f `df1' ") =" as res %8.2f
> `chi2'
= di as txt _col(12) "chi2(" %3.0f __00000Z ") =" as res %8.
> 2f __00000Y
- di as txt _col(10) "Prob > chi2 = " as res %8.4f `p'
= di as txt _col(10) "Prob > chi2 = " as res %8.4f __000010
- }
- else {

```

```

di as txt "          F(" %3.0f `df1' "," %6.0f `df2' ") =" as
> res %8.2f `F'
di as txt _col(13) "Prob > F =" as res %10.4f `p'
}
- }
- else {
local nexp = rowsof(`tm')
local nc = colsof(`tm')
if "`testtype'" == "F" {
local d "F(df,`= `df2'')"
local dip "p > F"
}
else {
local d chi2
local dip "p > chi2"
}
di
di as txt "{hline 7}{c TT}{hline 30}"
di as txt "          {c |}{ralign 12:`d'}          df{ralign 11:`d
> ip'}"
di as txt "{hline 7}{c +}{hline 30}"
forvalues it = 1 / `nexp' {
di as txt "    (`it') {c |}" as res _col(12) %9.2f `tm'[`it
> ',1] _col(22) %6.0f `tm'[`it',2] _col(33) %6.4f `tm'[`it',`pindex'] as txt "
> *"
}
di as txt "{hline 7}{c +}{hline 30}"
di as txt "    All {c |}" as res _col(12) %9.2f ``testtype'
> ' _col(22) %6.0f `df1' _col(33) %6.4f `p'
di as txt "{hline 7}{c BT}{hline 30}"
_mttest footer 1 "`mtmethod'" "*" "
}
- if "`coef'" != "" {
= if "" != "" {
Table
}
- return scalar p = `p'
= return scalar p = __000010
- return scalar df = `df1'
= return scalar df = __00000Z
- if "`testtype'" == "F" {
= if "chi2" == "F" {
return scalar F = `F'
return scalar df_r = `df2'
}
- else return scalar chi2 = `chi2'

```



```

= else return scalar chi2 = __00000Y
- return scalar drop = `drop'
= return scalar drop = __000011
- local i 1
- while "`dropped_`i'" != "" {
= while "" != "" {
    return scalar dropped_`i' = `dropped_`i''
    local ++i
  }
- if "`mtmethod'" != "" {
= if "" != "" {
    return matrix mtest `tm'
    return local mtmethod `mtmethod'
  }
}

> ----- end test.WaldTest -----
- return add

> ----- end test -----
- if !c(rc) {
- if `:length local minimum' {
= if 7 {
- quietly test, min

> ----- begin test -----
- if "`e(cmd)'" == "anova" | "`e(cmd)'" == "manova" {
= if "mopt" == "anova" | "mopt" == "manova" {
    AnovaTestChooser `0'
    return add
    exit
  }
- if "`e(cmd)'" == "mixed" {
= if "mopt" == "mixed" {
    MixedTestChooser `0'
    return add
    exit
  }
- if _caller() < 8 {
    _test `0'
    return add
    exit
  }
- is_svy

> ----- begin is_svy -----
- version 8

```

```

- syntax [, Regression]
- if ``e(prefix)'' == "svy" {
= if ``'' == "svy" {
    return scalar is_svy = 1
  }
- else if ``regression'' == "" {
= else if "" == "" {
- return scalar is_svy = ("e(N_psu)'' != "")
= return scalar is_svy = ("'' != "")
- }
- else {
    capt mat list e(b)
    local rc = _rc
    return scalar is_svy = ("e(N_psu)'' != "") & (`rc' == 0)
  }
}

> ----- end is_svy -----

- local is_svy = r(is_svy)
- if replay() & `is_svy' == 0 {
= if replay() & 0 == 0 {
- _test `0'
= _test , min
- return add
- exit

> ----- end test -----

- }
- ereturn scalar df_m = r(df)
- local drop = r(drop)
- if `drop' {
= if 0 {
    local drop 0
    local i 1
    local idrop ``r(dropped_`i')''
    while ``idrop'' != "" {
      if !`omit'[1,`idrop'] {
        local ++drop
      }
      local ++i
      local idrop ``r(dropped_`i')''
    }
  }
- if `drop' == 0 | ``cns'' != "" {
= if 0 == 0 | "" != "" {
- if !missing(e(df_r)) {
    ereturn scalar F = r(F)
  }
}

```

```

    }
- else {
-   ereturn scalar chi2 = r(chi2)
-   ereturn local chi2type Wald
- }
- ereturn scalar p = r(p)
- }
- else {
  if !missing(e(df_r)) {
    ereturn scalar F = .
  }
  else {
    ereturn scalar chi2 = .
    ereturn local chi2type Wald
  }
  ereturn scalar p = .
}
- }
- else {
  if missing(e(df_r)) {
    ereturn local chi2type Wald
  }
  ereturn scalar df_m = 0
}

> ----- end _prefix_model_test -----
- }
- else {
  _prefix_model_test, $ML_svy $ML_noadj
}
- }
- else if `:length global ML_ll_0' {
= else if 18 {
  if ``maxmin''' == "minimize" {
    ereturn scalar chi2 = -2*(e(ll)-e(ll_0))
  }
  else {
    ereturn scalar chi2 = 2*(e(ll)-e(ll_0))
  }
  ereturn scalar p = chi2tail(e(df_m), e(chi2))
  ereturn local chi2type "LR"
}
- if `:length global ML_wexp' {
= if 0 {
  ereturn local wtype ``$ML_wtyp''
  ereturn local wexp ``$ML_wexp''
}

```

```

    }
- if `:length global ML_group' {
= if 0 {
    ereturn local group `"$ML_group"'
    if ``"e(clustvar)'" == "$ML_grp" & "$ML_grp" != "" {
    ereturn local clustvar `"$ML_group"'
    }
}
- ereturn local cmd "ml"
- if `:length local score' {
= if 0 {
    mata: Mopt_score(0)
    ereturn local scorevars `scores'
}
- if "$ML_preserve" != "no" {
= if "no" != "no" {
    restore
    if ""e(prefix)'" == "svy" & e(N_strata_omit) {
    tempname v
    quietly _svy2 $ML_sample if $ML_sample, type(total) svy v(`v') s
> ubpop(`e(subpop)') touse($ML_sample)
    }
    ereturn repost, esample($ML_sample)
}
- if ``"moptobj'" != "" {
= if `""' != "" {
    local clear noclear
}
- if !`:length local clear' {
= if !0 {
- Clear
}

> ----- begin mopt.Clear -----

- syntax [, NOTANOPTION]
- capture drop $ML_w
= capture drop __000006
- capture drop $ML_samp
= capture drop __000006
- capture drop $ML_sample
= capture drop __000006
- capture drop $ML_subv
= capture drop
- capture drop $ML_grp
= capture drop
- mata: Mopt_drop_external()
- macro drop ML_*

```

```

> ----- end mopt.Clear -----
- }
- else {
  capture confirm var $ML_sample
  if c(rc) {
    quietly gen byte $ML_sample = e(sample)
  }
}
- if `:length local output' == 0 {
= if 8 == 0 {
  Display, level(`level') `options'
}
}

> ----- end mopt.MaxMin -----
- }

> ----- end mopt.Model -----
- if c(rc) {
  local rc = c(rc)
  Clear
  exit `rc'
}
- exit

> ----- end mopt -----
- exit

> ----- end ml -----
- est local cmd
- if "`score'"!="" {
= if ""!="" {
  label var `scvar1' "Score index for x*b from nbreg"
  rename `scvar1' `sname1'
  label var `scvar2' "Score index for /ln`parm' from nbreg"
  rename `scvar2' `sname2'
  est local scorevars `sname1' `sname2'
}
- _b_pclass PCDEF : default

> ----- begin _b_pclass -----
- version 15
- args c_value COLON key
- _on_colon_parse `0'
= _on_colon_parse PCDEF : default

```

```

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `"(`before)'"
}
- if `""before'" == ":" {
= if `PCDEF '" == ":" {
  sreturn local after `""after'"
  exit
}
- while `""COLON'" != ":" & `""after'" != "" {
= while `"" != ":" & `": default'" != "" {
- gettoken COLON after : after, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
  local before `before' (`COLON')
  local COLON
}
- else if `""COLON'" != ":" {
= else if `":'" != ":" {
  local before `""before' `COLON'"
  local COLON
}
- }
- while `""COLON'" != ":" & `""after'" != "" {
= while `":'" != ":" & ` default'" != "" {
  gettoken COLON after : after, parse(":") bind match(par) quotes
  if "`par'" != "" {
    local before `before' (`COLON')
    local COLON
  }
  else if `""COLON'" != ":" {
    local before `""before' `COLON'"
    local COLON
  }
}
- if `""COLON'" != ":" {
= if `":'" != ":" {
  di as err "" found where ':' expected"
  exit 198
}
- sreturn local before `""before'"

```

```

= sreturn local before `"PCDEF "`
- sreturn local after `""`after'''
= sreturn local after `" default"'

> ----- end _on_colon_parse -----
- local c_value `""`s(before)'''
= local c_value `"PCDEF "`
- local key `""`s(after)'''
= local key `" default"'
- confirm name `c_value'
= confirm name PCDEF
- mata: st_b_pclass("`list retok key'")
= mata: st_b_pclass("default")
- c_local `c_value' `value'
= c_local PCDEF 0

> ----- end _b_pclass -----
- _b_pclass PCAUX : aux

> ----- begin _b_pclass -----
- version 15
- args c_value COLON key
- _on_colon_parse `0'
= _on_colon_parse PCAUX : aux

> ----- begin _on_colon_parse -----
- version 8.2
- sreturn local before ""
- sreturn local after ""
- gettoken before after : 0, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
    local before `("before')"'
}
- if `""before'"" == ":" {
= if `"PCAUX "' == ":" {
    sreturn local after `""`after'''
    exit
}
- while `""COLON'"" != ":" & `""after'"" != "" {
= while `""' != ":" & `": aux"' != "" {
- gettoken COLON after : after, parse(":") bind match(par) quotes
- if "`par'" != "" {
= if "" != "" {
    local before `before' (`COLON')
    local COLON

```

```

    }
- else if ``COLON''' != ":" {
= else if `":'" != ":" {
    local before ``before' `COLON'''
    local COLON
    }
- }
- while ``COLON''' != ":" & ``after''' != "" {
= while `":'" != ":" & `aux'" != "" {
    gettoken COLON after : after, parse(":") bind match(par) quotes
    if "par'" != "" {
        local before `before' (`COLON')
        local COLON
    }
    else if ``COLON''' != ":" {
        local before ``before' `COLON'''
        local COLON
    }
}
- if ``COLON''' != ":" {
= if `":'" != ":" {
    di as err "" found where ':' expected"
    exit 198
}
- sreturn local before ``before'''
= sreturn local before `PCAUX "'
- sreturn local after ``after'''
= sreturn local after `aux"'
}

> ----- end _on_colon_parse -----

- local c_value ``s(before)'''
= local c_value `PCAUX "'
- local key ``s(after)'''
= local key `aux"'
- confirm name `c_value'
= confirm name PCAUX
- mata: st_b_pclass("`list retok key'")
= mata: st_b_pclass("aux")
- c_local `c_value' `value'
= c_local PCAUX 2

> ----- end _b_pclass -----

- tempname pclass
- matrix `pclass' = e(b)
= matrix __000006 = e(b)
- local dim = colsof(`pclass')

```



```

= local dim = colsof(__000006)
- matrix `pclass'[1,1] = J(1,`dim',`PCDEF')
= matrix __000006[1,1] = J(1,10,0)
- matrix `pclass'[1,`dim'] = `PCAUX'
= matrix __000006[1,10] = 2
- est hidden matrix b_pclass `pclass'
= est hidden matrix b_pclass __000006
- if "`llp'"!=" {
= if ""!=" {
  est local chi2_ct "LR"
  est scalar ll_c = `llp'
  if (e(ll) < e(ll_c)) | (_b[/ln`parm'] < -20) {
    est scalar chi2_c = 0
  }
  else est scalar chi2_c = 2*(e(ll)-e(ll_c))
}
- est scalar r2_p = 1 - e(ll)/e(ll_0)
- est scalar k_aux = 1
- est hidden local diparm_opt2 noprob
- est scalar `parm' = exp(_b[/ln`parm'])
= est scalar alpha = exp(_b[/lnalpha])
- est local dispers "`dispers'"
= est local dispers "mean"
- est local offset "`offvar'"
= est local offset ""
- est local offset1
- est hidden local marginsderiv default N IR
- est local predict "nbreg_p"
- est local cmd "nbreg"
- global S_E_ll = e(ll)
- global S_E_ll0 `e(ll_0)'
= global S_E_ll0 -414.2287919313009
- global S_E_llc `e(ll_c)'
= global S_E_llc
- global S_1 `e(chi2_c)'
= global S_1
- global S_E_chi2 = e(chi2)
- global S_E_mdf = e(df_m)
- global S_E_pr2 `e(r2_p)'
= global S_E_pr2 .1288955078812051
- global S_E_nobs = e(N)
- global S_E_tdf = e(N)
- global S_E_off `e(offset)'
= global S_E_off
- global S_E_depv `e(depvar)'
= global S_E_depv y_timing_max

```

```

- global S_E_cmd `e(cmd)'
= global S_E_cmd nbreg
- if "`display'"==" " {
= if ""==" " {
- Display, `irr' level(`level') `diopts'
= Display, level(95)

```

---

```

> ----- begin nbreg.Display -----
- syntax [, Level(cilevel) IRr *]
- _get_diopts diopts, `options'
= _get_diopts diopts,

```

---

```

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels
> NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NO
> OMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(string) sfor
> mat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(
> string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV
> ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels A
> LLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells
> EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat
> (string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passth
> ru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passth
> ru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `""' != "" {
local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'`` != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'`` != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts'"' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
}
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
}
- }

```



```

    }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth(`"'cformat'"') > 9 {
        local cformat "%9.0g"
        di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat `"'cformat'"'
    local cformat `"cformat(`cformat')"'
}
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth(`"'sformat'"') > 8 {
        local sformat "%8.2f"
        di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat `"'sformat'"'
    local sformat `"sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`"'pformat'"') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `"'pformat'"'
    local pformat `"pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {

```

```

        di as err "invalid fvwrapon() option;"
        syntax [, WOrd Width]
        exit 198
    }
    - opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
= local fvwrapon
    - if `:length local fvwrapon' {
= if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }
> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
= if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevel
> s' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend
> ' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwr
> ap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
= c_local diopts
    - if `K' == 2 {
= if 2 == 2 {
    - c_local `c_opts' ```options'""
= c_local options `""'
    - }
    - sreturn local coding `coding'

```

```

= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' `""diopts""'
= c_local diopts `""'
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_after = s(after)
- }
- local S_`sname' = s(`sname')
= local S_before = s(before)
- }
- local S_`sname' = s(`sname')
= local S_log = s(log)
- }
- local S_`sname' = s(`sname')
= local S_deriv = s(deriv)
- }
- local S_`sname' = s(`sname')
= local S_namelist = s(namelist)
- }
- local S_`sname' = s(`sname')
= local S_optlist = s(optlist)
- }
- local S_`sname' = s(`sname')
= local S_k_fill = s(k_fill)
- }
- local S_`sname' = s(`sname')
= local S_k = s(k)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels
> NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NO

```

```

> OMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sfor
> mat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(
> string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV
> ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels A
> LLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells
> EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat
> (string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passth
> ru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passth
> ru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

```

---

```

> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {

```

```

        - exit
    > ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nobaselevels'"
    = opts_exclusive " "

    > ----- begin opts_exclusive -----
        - version 8.2
        - args opts optname errcode optslab
        - local opts `opts'
        = local opts
        - local n 0
        - while ``opts' != "" {
        = while `"' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit

    > ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"
    = opts_exclusive " "

    > ----- begin opts_exclusive -----
        - version 8.2
        - args opts optname errcode optslab
        - local opts `opts'
        = local opts
        - local n 0
        - while ``opts' != "" {
        = while `"' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit

    > ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
    = opts_exclusive " "

    > ----- begin opts_exclusive -----
        - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'`opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "'`omitted' `noomitted'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'`opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "'`lstretch' `nolstretch'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'`opts'"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```



```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {

```

```

- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {

```

```

= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`"pformat"') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `"pformat"'
    local pformat `pformat(pformat)'"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, W0rd WIdth]
    exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `"'opts"' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----

```

```

- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')
}
}

> ----- end _get_diopts.Wrapon -----

- if "`fvwrap'" != "" {
= if "" != "" {
  FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevel
> s' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend
> ' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwr
> ap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbrev'
= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
  c_local `c_opts' ``options'
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

- if ``name2' == "" {
= if ``' == "" {
- GetDiopts DUPS, `options'
= GetDiopts DUPS,

> ----- begin _get_diopts.GetDiopts -----

- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASElevels
> NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYcells NO
> OMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string) sfor
> mat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVWRAPON(
> string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOABBREV
> ABBREV

- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASElevels A
> LLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells

```

```

> EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat
> (string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passth
> ru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passth
> ru) NOABBREV ABBREV *)
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
    _____

> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____

> _____ end opts_exclusive _____
    - opts_exclusive "`allbaselevels' `nbaselevels'"
    = opts_exclusive " "
    _____

> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____

> _____ end opts_exclusive _____
    - opts_exclusive "`baselevels' `nbaselevels'"
    = opts_exclusive " "
    _____

> _____ begin opts_exclusive _____

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ```opts'"" != "" {
= while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```



```

- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

        = if 0 < 2 {
        - exit
    }
> ----- end opts_exclusive -----
    - opts_exclusive "`abbrev' `noabbrev'"
    = opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    }
> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'
    = local omitted
    - local lstretch `lstretch' `nolstretch'
    = local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
    = local fvlabel
    - local abbrev `abbrev' `noabbrev'
    = local abbrev
    - if "`allbaselevels'`baselevels'" == "" {
    = if "" == "" {
    - if c(showbaselevels) == "all" {
        local allbaselevels allbaselevels
    }
    - else if c(showbaselevels) == "on" {
        local baselevels baselevels
    }
    - }

```

```

- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
  }

```

```

        local pformat `"pformat(`pformat')"'
    }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

```

```

> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `baselevel
> s' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coeflegend
> ' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare' `fvwr
> ap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citytype' `abbrev'
    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
    -----
> ----- end _get_diopts.GetDiopts -----
    - }
    - else {
        GetDiopts DUPS options, `options'
        c_local `name2' ``options''
    }
    - local DUPS : list DUPS - DEFAULT
    - gettoken DUPS : DUPS
    - if ``DUPS'' != "" {
    = if ``' != "" {
        if (strpos(`DUPS', "(")) {
            gettoken DUPS : DUPS, parse("(")
            local DUPS `DUPS'()'
        }
        di as err "duplicate {bf:`DUPS'} option not allowed"
        exit 198
    }
    - sreturn clear
    - foreach sname of local snames {
    - sreturn local `sname' = ``S_`sname''
    = sreturn local level = `95'
    - }
    - sreturn local `sname' = ``S_`sname''

```

```

= sreturn local after = `"aux"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local before = `"PCAUX "'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local log = `"log"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local deriv = `"0"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local namelist = `"cluster"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local optlist = `"CLuster"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local k_fill = `"10"'
- }
- sreturn local `sname' = `"'S_`sname'""'
= sreturn local k = `"10"'
- }

```

---

```

> ----- end _get_diopts -----
- if "`irr'"!="" {
= if ""!="" {
  local eopt "eform(IRR)"
}
- if "`e(dispers)'"=="mean" {
= if "mean"=="mean" {
- local parm alpha
- }
- else local parm delta
- version 9: ml di, level(`level') `eopt' nofootnote `diopts'
= version 9: ml di, level(95) nofootnote

```

---

```

> ----- begin ml -----
- if _caller()>=11 {
  mopt `0'
  exit
}
- version 8.2, missing
- version 6, missing
- if _caller()<6 {
  ml_5 `0'

```

```

    exit
}
- local caller : display string(_caller())
- gettoken cmd 0: 0, parse(" ,")
- local l = length(`""`cmd'`)
= local l = length(`"di"`)
- if `""`cmd'` == "technique" {
= if `"di"` == "technique" {
    ml_technique `0'
    exit
}
- if `""`cmd'` == "clear" {
= if `"di"` == "clear" {
    ml_clear `0'
    exit
}
- if `""`cmd'` == "hold" {
= if `"di"` == "hold" {
    ml_hold `0'
    exit
}
- if `""`cmd'` == "unhold" {
= if `"di"` == "unhold" {
    ml_unhold `0'
    exit
}
- if `""`cmd'` == "score" {
= if `"di"` == "score" {
    if "e(opt)" == "moptimize" {
        mopt score `0'
        exit
    }
    ml_score `0'
    exit
}
- if `""`cmd'`=="count" {
= if `"di"`=="count" {
    ml_c_d count `0'
    exit
}
- if `""`cmd'` == "trace" {
= if `"di"` == "trace" {
    ml_c_d trace `0'
    exit
}
- if `""`cmd'`==bsubstr("graph",1,max(2,`l`)) {

```

```

= if ``di''==bsubstr("graph",1,max(2,2)) {
  ml_graph `0'
  exit
}
- if ```cmd''' == "init" {
= if ``di'' == "init" {
  ml_init `0'
  exit
}
- if ```cmd''' == bsubstr("maximize",1,max(3,`l')) {
= if ``di'' == bsubstr("maximize",1,max(3,2)) {
  version `caller', missing: AllowEv ml_max `0'
  exit
}
- if ```cmd''' == bsubstr("display",1,max(2,`l')) | ```cmd''' == bsubs
> tr("mlout", 1,max(3,`l')) {
  = if ``di'' == bsubstr("display",1,max(2,2)) | ``di'' == bsubstr("mlou
> t", 1,max(3,2)) {
  - if ``e(opt)'' == "moptimize" {
  = if "moptimize" == "moptimize" {
  - mopt display `0'
  = mopt display , level(95) nofootnote
}

> ----- begin mopt -----
- local vv : di "version " string(_caller()) ":"
- version 12
- gettoken subcmd 0 : 0, parse(" ,")
- local lcmd : length local subcmd
- if ``subcmd'' == "clear" {
= if "display" == "clear" {
  Clear
  exit
}
- if ```subcmd''' == "hold" {
= if ``display'' == "hold" {
  ml_hold `0'
  exit
}
- if ```subcmd''' == "unhold" {
= if ``display'' == "unhold" {
  ml_unhold `0'
  exit
}
- if ``subcmd'' == "score" {
= if "display" == "score" {
  if _caller() > 12 {

```



```

`vv' Score `0'
}
else {
Score `0'
}
exit
}
- if "`subcmd'" == "count" {
= if "display" == "count" {
Count `0'
exit
}
- if "`subcmd'" == "trace" {
= if "display" == "trace" {
Trace `0'
exit
}
- if "`subcmd'" == bsubstr("graph",1,max(2,`lcmd')) {
= if "display" == bsubstr("graph",1,max(2,7)) {
Graph `0'
exit
}
- if "`subcmd'" == "init" {
= if "display" == "init" {
Interactive 0
`vv' Init `0'
exit
}
- if "`subcmd'" == bsubstr("maximize",1,max(3,`lcmd')) {
= if "display" == bsubstr("maximize",1,max(3,7)) {
Interactive 1
`vv' MaxMin maximize `0'
exit
}
- if "`subcmd'" == bsubstr("minimize",1,max(3,`lcmd')) {
= if "display" == bsubstr("minimize",1,max(3,7)) {
Interactive 1
`vv' MaxMin minimize `0'
exit
}
- if "`subcmd'" == bsubstr("display",1,max(2,`lcmd')) {
= if "display" == bsubstr("display",1,max(2,7)) {
- Display `0'
= Display , level(95) nofootnote

```

---

> \_\_\_\_\_ begin mopt.Display \_\_\_\_\_

```

- if !inlist("`e(opt)'" , "ml", "moptimize") {
= if !inlist("moptimize", "ml", "moptimize") {
  error 301
}
- if "`e(svyml)'" != "" | "`e(prefix)'" == "svy" {
= if "" != "" | "" == "svy" {
  _prefix_display `0'
  exit
}
- syntax [, Level(cilevel) PPlus noHeader First SHOWEQns NEQ(intege
> r -1) noFOOTnote NOSKIP Title(passthru) NOTEST * ]
- if "`plus'" != "" {
= if "" != "" {
  local footnote nofootnote
}
- if `neq' >= 0 {
= if -1 >= 0 {
  local neqopt neq(`neq')
}
- _get_diopts diopts options, `options'
= _get_diopts diopts options,

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`emptycells` `noemptycells`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel` `nofvlabel`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while "`opts'" != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }

```



```

- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `pformat(`pformat')"'
  }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
  }

```

```

- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, W0rd WIdth]
    exit 198
    }
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
    }
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas

```

```

> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abbre
> v'

```

```

    = c_local diopts
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options'''
    = c_local options `'''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

```

---

```

> _____ end _get_diopts.GetDiopts _____
    - c_local `name1' ``diopts'''
    = c_local diopts `'''
    - local snames : s(macros)
    - foreach sname of local snames {
    - local S_`sname' = s(`sname')
    = local S_level = s(level)
    - }
    - local S_`sname' = s(`sname')
    = local S_k = s(k)
    - }
    - local S_`sname' = s(`sname')
    = local S_k_fill = s(k_fill)
    - }
    - local S_`sname' = s(`sname')
    = local S_optlist = s(optlist)
    - }
    - local S_`sname' = s(`sname')
    = local S_namelist = s(namelist)
    - }
    - local S_`sname' = s(`sname')
    = local S_deriv = s(deriv)
    - }
    - local S_`sname' = s(`sname')
    = local S_log = s(log)
    - }
    - local S_`sname' = s(`sname')

```

```

= local S_before = s(before)
- }
- local S_`sname' = s(`sname')
= local S_after = s(after)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`noci` `nopvalues`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding` `compare`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {

```



```

        - exit
    > ----- end opts_exclusive -----
        - opts_exclusive "`versus' `compare'"
    = opts_exclusive " "
    > ----- begin opts_exclusive -----
        - version 8.2
        - args opts optname errcode optslab
        - local opts `opts'
    = local opts
        - local n 0
        - while ```opts'`` != "" {
    = while `""' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
    = if 0 < 2 {
            - exit
        > ----- end opts_exclusive -----
        - opts_exclusive "`abbrev' `noabbrev'"
    = opts_exclusive " "
    > ----- begin opts_exclusive -----
        - version 8.2
        - args opts optname errcode optslab
        - local opts `opts'
    = local opts
        - local n 0
        - while ```opts'`` != "" {
    = while `""' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
    = if 0 < 2 {
            - exit
        > ----- end opts_exclusive -----
        - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
        - local baselevels `baselevels' `nobaselevels'
    = local baselevels
        - local emptycells `emptycells' `noemptycells'

```

```

= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')"'
}
}

```

```

- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth("`sformat'") > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat "`sformat'"
  local sformat `sformat(`sformat')'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

> ----- end _get_diopts.Wrapon -----

- if ``fvwrap'' != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abbre
> v'

= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options''
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

- if ``name2'' == "" {
= if ``options'' == "" {
    GetDiopts DUPS, `options'

```

```

    }
    - else {
    - GetDiopts DUPS options, `options'
    = GetDiopts DUPS options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
    - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBASEl
> evels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTYce
> lls NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(string
> ) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) FVW
> RAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) NOA
> BBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASEle
> vels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTY
> cells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND c
> format(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(
> passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(
> passthru) NOABBREV ABBREV *]
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

```

---

```

> ----- end opts_exclusive -----
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts

```

```

- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```



```

        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}
> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}
> ----- end opts_exclusive -----

```

```

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted

```

```

- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'

```

```

if fmtwidth("`sformat'") > 8 {
local sformat "%8.2f"
di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat "`sformat'"
local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth("`pformat'") > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat "`pformat'"
local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wraon , `fvwraon'
= Wraon ,

```

---

> begin \_get\_diopts.Wraon

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, WOrd WIdth]
exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `""' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `bas
> elevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `coef
> legend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compare'
> `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abbre
> v'

    = c_local DUPS
    - if `K' == 2 {
    = if 2 == 2 {
        - c_local `c_opts' ``options''
        = c_local options `''
        - }
        - sreturn local coding `coding'
        = sreturn local coding
        - sreturn local versus `versus'
        = sreturn local versus
        - sreturn local compare `compare'
        = sreturn local compare
        - sreturn local level `level'
        = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

    - c_local `name2' ``options''
    = c_local options `''
    - }
    - local DUPS : list DUPS - DEFAULT
    - gettoken DUPS : DUPS

```

```

- if ``DUPS'' != "" {
= if `'' != "" {
    if (strpos(`DUPS', "(") {
        gettoken DUPS : DUPS, parse("(")
        local DUPS ``DUPS'()'
    }
    di as err "duplicate {bf:`DUPS'} option not allowed"
    exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname''
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local k = `10'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local k_fill = `10'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local optlist = `CLuster'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local namelist = `cluster'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local deriv = `0'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local log = `log'
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local before = `PCAUX '
- }
- sreturn local `sname' = ``S_`sname''
= sreturn local after = `aux'
- }

> _____ end _get_diopts _____
- _get_eformopts , eformopts(`options') soptions allowed(__all__)
= _get_eformopts , eformopts() soptions allowed(__all__)
> _____ begin _get_eformopts _____
- version 8.0
- syntax [, eformopts(string asis) soptions ALLOWed(string) ]

```

```

- local 0 , `eformopts'
= local 0 ,
- local EFALL NOHR hr NOSHR shr or IRr RRr tr TRatio
- if ""options"" != "" {
= if "options" != "" {
- syntax [, EForm1(passthru) EForm `EFALL' * ]
= syntax [, EForm1(passthru) EForm NOHR hr NOSHR shr or IRr RRr
> tr TRatio * ]
- local opts ``options'
= local opts `''
- local 0 , `eform1' `eform' `nohr' `hr' `noshr' `shr' `or' `irr
> ' `rrr' `tr' `tratio'
= local 0 ,
- }
- if "`allowed'" == "__all__" {
= if "__all__" == "__all__" {
- local allowed `EFALL'
= local allowed NOHR hr NOSHR shr or IRr RRr tr TRatio
- }
- foreach ef of local allowed {
- capture confirm name `ef'
= capture confirm name NOHR
- if _rc {
di as err "`ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr
- }
- capture confirm name `ef'
= capture confirm name hr
- if _rc {
di as err "`ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr
- }
- capture confirm name `ef'
= capture confirm name NOSHR
- if _rc {
di as err "`ef' is not a valid name"
exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr

```

```

- }
- capture confirm name `ef'
= capture confirm name shr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr
- }
- capture confirm name `ef'
= capture confirm name or
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or
- }
- capture confirm name `ef'
= capture confirm name IRr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr
- }
- capture confirm name `ef'
= capture confirm name RRr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr
- }
- capture confirm name `ef'
= capture confirm name tr
- if _rc {
    di as err "`ef' is not a valid name"
    exit 198
  }
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr
- }

```



```

- capture confirm name `ef'
= capture confirm name TRatio
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr tratio
- }
- local efopts : list uniq efopts
- syntax [, EForm1(string) EForm `allowed' ]
= syntax [, EForm1(string) EForm NOHR hr NOSHR shr or IRr RRr tr
> TRatio ]
- foreach ef of local efopts {
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- if ```eform1'''"=="Odds ratio" {
= if `""'"=="Odds ratio" {
  local eform_cons_ti = "Odds"
  local eform_cons_note = "odds ratios"
}

```

```

- else if ```eform1'``'=="Haz. ratio" {
= else if `''`'=="Haz. ratio" {
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
}
- else if ```eform1'``'=="Time ratio" {
= else if `''`'=="Time ratio" {
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
}
- else if ```eform1'``'=="IRR" {
= else if `''`'=="IRR" {
    local eform_cons_ti = "Inc. rate"
    local eform_cons_note = "incidence-rate ratios"
    local eform_cons_ti2 = "incidence rate"
}
- else if ```eform1'``'=="Risk ratio" {
= else if `''`'=="Risk ratio" {
    local eform_cons_ti = "Risk"
    local eform_cons_note = "risk ratios"
}
- else if ```eform1'``'=="Hlth ratio" {
= else if `''`'=="Hlth ratio" {
    local eform_cons_ti = "Health"
    local eform_cons_note = "health ratios"
}
- local k : list sizeof eform
- if `k' {
= if 0 {
    opts_exclusive ```eform'``
    if `:length local eform1' {
    opts_exclusive "eform() `eform'"
    }
}
- if `k' {
= if 0 {
    if ("`eform'`=="eform") local eform1 = "exp(b)"
    else if ("`eform'`=="hr") {
        local eform1 = "Haz. ratio"
        local eform_cons_ti = "Hazard"
        local eform_cons_note = "hazard ratios"
    }
    else if ("`eform'`=="shr") {
        local eform1 = "SHR"
        local eform_cons_ti = "Subhazard"
        local eform_cons_note = "subhazard ratios"
    }
}

```

```

    }
    else if ("`eform"=="tr") {
        local eform1 = "Time ratio"
        local eform_cons_ti = "Time"
        local eform_cons_note = "time ratios"
    }
    else if ("`eform"=="tratio") {
        local eform1 = "Time ratio"
        local eform_cons_ti = "Time"
        local eform_cons_note = "time ratios"
    }
    else if ("`eform"=="or") {
        local eform1 = "Odds ratio"
        if ("`e(cmd)'"=="asclogit" | "`e(cmd)'"=="cmclogit" | "`e(cmd)
> '"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
            local eform_cons_ti = "Rel. risk"
            local eform_cons_note = "relative-risk ratios"
            local eform_cons_ti2 = "relative risk"
        }
        else {
            local eform_cons_ti = "Odds"
            local eform_cons_note = "odds ratios"
        }
    }
    else if ("`eform"=="irr") {
        local eform1 = "IRR"
        local eform_cons_ti = "Inc. rate"
        local eform_cons_note = "incidence-rate ratios"
        local eform_cons_ti2 = "incidence rate"
    }
    else if ("`eform"=="rrr") {
        local eform1 = "RRR"
        local eform_cons_ti = "Rel. risk"
        local eform_cons_note = "relative-risk ratios"
        local eform_cons_ti2 = "relative risk"
    }
    else if ("`eform"!="nohr") & ("`eform"!="noshr") {
        local eform1 = upper("`eform'")
    }
}

- sreturn clear
- sreturn local options ```opts'``
= sreturn local options `""'
- sreturn local opt `eform'
= sreturn local opt
- sreturn local str `eform1'

```

```

= sreturn local str
- if ``eform1''' != "" {
= if `''' != "" {
    sreturn local eform eform(`eform1'')
  }
- else sreturn local eform ""
- sreturn local eform_cons_ti `eform_cons_ti'
= sreturn local eform_cons_ti
- sreturn local eform_cons_note `eform_cons_note'
= sreturn local eform_cons_note
- sreturn local eform_cons_ti2 `eform_cons_ti2'
= sreturn local eform_cons_ti2

```

---

```

> ----- end _get_eformopts -----
- local eform `s(eform)'
= local eform
- local diparm `s(options)'
= local diparm
- _get_diparmopts , diparmopts(`diparm') level(`level')
= _get_diparmopts , diparmopts() level(95)

```

---

```

> ----- begin _get_diparmopts -----
- if _caller() < 8.2 {
  _get_diparmopts_8 `0'
  exit
}
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom plus
> SYNTAXonly * ]
- local goptions `options'
= local goptions level(95)
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk , level(95)

```

---

```

> ----- begin _get_diparmopts.SpliceGlobals -----
- syntax name(id="macro name" name=c_diparm) [, diparm(string
> asis) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof''' == "" & ``e(df_r)''' != "" {
= if `''' == "" & `''' != "" {
    capture confirm integer number `e(df_r)'
    if !c(rc) {
      local gdof dof(`e(df_r)')
    }
    capture
  }

```

```

- else local gdof ``dof'''
= else local gdof `'''
- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)
- if (``diparm''' == '') exit
= if (``''' == '') exit

```

---

```

> ----- end _get_diparmopts.SpliceGlobals -----
- local 0 `", `diparmopts'''
= local 0 `", ''
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while ``diparm''' != '' {
= while `''' != '' {
    local ++k
    local single 1
    local after
    capture _on_colon_parse `diparm'
    if !c(rc) {
        local single 0
        local diparm ``s(before)'''
        local after ``s(after)'''
    }
    capture
    SpliceGlobals diparm , diparm(`diparm') `syntaxonly' `goptions

```

```

> '
    local diparm`k' ``diparm'''
    local diparmopts ``diparmopts' : `diparm'''
    if `single' | ``:list retok after''' == '' {
        local 0 `", `options'''
        syntax [, diparm(string asis) * ]
    }
    else {
        local diparm ``after'''
    }
}
- if ``soptions''' == '' {
= if `''' == '' {
- syntax [, diparm(string asis) ]
- }
- if (``execute'' == '') {
= if ("" == "") {
- sreturn clear

```

```

- sreturn local options ``options'''
= sreturn local options `'''
- sreturn local diparm ``diparmopts'''
= sreturn local diparm `'''
- forval i = 1/`k' {
= forval i = 1/0 {
    sreturn local diparm`i' ``diparm`i''''
    }
- sreturn local k `k'
= sreturn local k 0
- exit

```

---

```

> ----- end _get_diparmopts -----
- if ``header'""=="" {
= if ""=="" {
- _coef_table_header, `title'
= _coef_table_header,

```

---

```

> ----- begin _coef_table_header -----
- version 9
- if !c(noisily) {
    exit
    }
- syntax [, rclass noHeader obj(string) head1left(numlist max=1
> integer >=0) head1right(numlist max=1 integer >=0) head2left(numlist max=1 i
> nteger >=0) head2right(numlist max=1 integer >=0) head2start(numlist max=1 i
> nteger >=1) width(numlist max=1 integer >=1) * ]
- if (``header'"" != "") exit
= if ("" != "") exit
- if (``head2start'"" != "" & ``width'"" != "") {
= if ("" != "" & "" != "") {
    opts_exclusive "head2start() width()"
    exit 198
    }
- local has_obj = (``obj'"" != "")
= local has_obj = (``'"" != "")
- if (`has_obj' == 0) {
= if (0 == 0) {
- tempname obj
- }
- if (``rclass'"" != "") {
= if ("" != "") {
    tempname rhold
    _return hold `rhold'
    }
- `obj' = ._coef_table__header.new, `rclass' `options'

```

```

= .__000007 = .__coef_table__header.new,
-class __coef_table__header {
-     string  result          = "e"
-     string  newline         = "display"
-     double  opt_dvheader    = 0
-     double  opt_modeltest   = 0
-     double  opt_cluster     = 0
-     double  opt_rules       = 0
-     double  opt_tvar        = 0
-     double  opt_septitle    = 0
-     double  linesize        = 0
-     string  title           = ""
-     string  title2          = ""
-     string  groupvar        = ""
-     string  intmethod       = ""
-     double  intpoints       = -1
-     string  cmd             = ""
-     string  cmd2            = ""
-     string  prefix          = ""
-     double  is_prefix       = 0
-     double  is_svy          = 0
-     double  is_margins      = 0
-     double  is_gsem         = 0
-     double  is_gsem_group   = 0
-     double  is_sem          = 0
-     double  is_areg         = 0
-     double  is_demand       = 0
-     double  is_demandn      = 0
-     double  is_bacon        = 0
-     double  is_htedid       = 0
-     double  is_sum          = 0
-     double  is_tab          = 0
-     double  is_r2_p         = 1
-     double  is_multclust    = 0
-     double  is_stintcox     = 0
-     double  is_bmareg       = 0
-     double  prop_pushtitle  = 0
-     double  prop_groupvar   = 0
-     double  prop_censlimits = 0
-     double  prop_censored   = 0
-     double  prop_nouncensored = 0
-     double  prop_iselected = 0
-     double  prop_selected   = 0
-     double  prop_groups     = 0
-     double  prop_balance    = 0
-     double  ablen           = 16

```

```

-      double width      = 78
-      double c1         = 1
-      double c2         = 16
-      double c2a        = 16
-      double c3         = 49
-      double c4         = 67
-      double c2wfmt      = 10
-      double c4wfmt      = 10
-      string scheme     = "ml"
-      double lalignr     = 0
-      double ralignr     = 0
-      double lalignl     = 0
-      double ralignl     = 0
-      double lalignla    = 0
-      double ralignla    = 0
-      array left
-      array right
-}

```

---

```

> ----- begin _coef_table__header.new -----
- version 16
- syntax [, rclass noDVHeader noMODELtest TTitle(string asis) n
> ocluster noRULES noTVAR SEPTITLE linesize(int 0) ]
- if "`rclass'" != "" {
= if "" != "" {
    .result = "r"
}
- if "`dvheader'" == "" {
= if "" == "" {
    .opt_dvheader = 1
}
- if "`modeltest'" == "" {
= if "" == "" {
    .opt_modeltest = 1
}
- if "`cluster'" == "" {
= if "" == "" {
    .opt_cluster = 1
}
- if "`rules'" == "" {
= if "" == "" {
    .opt_rules = 1
}
- if "`tvar'" == "" {
= if "" == "" {
    .opt_tvar = 1
}

```



```

- }
- if "`septime'" != "" {
= if "" != "" {
    .opt_septime = 1
  }
- .linesize = `linesize'
= .linesize = 0
- .title = ``title'""
= .title = `""'
- if ``.result'"" == "e" {
= if ``e'"" == "e" {
- .New4e

```

---

```

> ----- begin _coef_table__header.New4e -----
- version 16
- .cmd = ``e(cmd)'"
= .cmd = "nbreg"
- .cmd2 = ``e(cmd2)'"
= .cmd2 = ""
- .prefix = ``e(prefix)'"
= .prefix = ""
- if ``.prefix'"" != "" {
= if `""' != "" {
    .is_prefix = 1
  }
- if ``.prefix'" == "svy" {
= if "" == "svy" {
    .is_svy = 1
  }
- if ``.cmd'" == "margins" {
= if "nbreg" == "margins" {
    .is_margins = 1
  }
- if inlist("`.cmd'", "contrast", "pwcompare") {
= if inlist("nbreg", "contrast", "pwcompare") {
    if ``.cmd2'" == "margins" {
      .is_margins = 1
    }
  }
- if ``.cmd'" == "gsem" | ``.cmd2'" == "gsem" {
= if "nbreg" == "gsem" | "" == "gsem" {
    .is_gsem = 1
  }
- if ``.cmd'" == "sem" {
= if "nbreg" == "sem" {
    .is_sem = 1

```

```

}
- if "`cmd'" == "areg" {
= if "nbreg" == "areg" {
  .is_areg = 1
}
- if ("`cmd'" == "demandsys" & ``e(doheader)'"=="yes") {
= if ("nbreg" == "demandsys" & `""'"=="yes") {
  .is_demand = 1
}
- if ("`cmd'" == "demandsys") {
= if ("nbreg" == "demandsys") {
  .is_demandn = 1
}
- if ("`cmd'" == "xthdidregress" & ``e(inestat)'"=="") {
= if ("nbreg" == "xthdidregress" & ""=="") {
  .is_htedid = 1
}
- if "`cmd'" == "stintcox" {
= if "nbreg" == "stintcox" {
  .is_stintcox = 1
}
- if "`cmd'" == "bdecomp" {
= if "nbreg" == "bdecomp" {
  .is_bacon = 1
}
- if "`cmd'" == "bmaregress" {
= if "nbreg" == "bmaregress" {
  .is_bmareg = 1
}
- is_svysum `cmd'
= is_svysum nbreg

```

```

> ----- begin is_svysum -----
- version 9
- local SvySum mean prop proportion ratio total
- if ``0'" != "" & `:list 0 in SvySum' {
= if `"nbreg'" != "" & 0 {
  return scalar is_svysum = 1
}
- else return scalar is_svysum = 0
> ----- end is_svysum -----
- .is_sum = r(is_svysum)
- if `.is_svy' {
= if 0 {
  if "`cmd'" == "tabulate" {

```

```

        .is_tab = 1
    }
}
- if `.is_gsem' {
= if 0 {
    if ``e(groupvar)'' != "" & ``.cmd2'' == "" {
        .is_gsem_group = 1
    }
}
- local clustvars `e(clustvar)'
= local clustvars
- local kclvar : list sizeof clustvars
- .is_multclust = ((`kclvar'>1) & ("`e(kcluster)'"=="matrix"

> ))

= .is_multclust = ((0>1) & ("=="matrix"))
- if `.is_multclust' {
= if 0 {
    .prop_pushtitle = 1
}
- local c2pfix 20
- if `.is_sem' | `.is_gsem_group' {
= if 0 | 0 {
    local crlen ``e(crittype)''
    local crlen : smcllength local crlen
    if `crlen' >= `c2pfix' {
        local c2pfix = `crlen' + 2
    }
}
- if `.is_sum' {
= if 0 {
    if `.linesize' > 62 {
        local offset = `.linesize' - 62
    }
    else local offset 0
    .width = 62 + `offset'
    .c2 = 18
    .c3 = 35 + `offset'
    .c4 = 51 + `offset'
    .c2wfmt = 7
    .c4wfmt = 10
    .scheme = "compact"
}
- else if `.is_prefix' | `.is_sem' | `.is_gsem_group' {
= else if 0 | 0 | 0 {
    .width = 78
    .c2 = `c2pfix'

```

```

        .c3 = 49
        .c4 = 67
        .c2wfmt = 9
        .c4wfmt = 10
        .scheme = "svy"
    }
    - if `.is_svy' {
= if 0 {
    local maxlen = `.width' - `.c2' - (`.c2wfmt' + 2) - 2 - (`.c4' - `.c3'

> ) - 2

    if `maxlen' > 19 {
    local maxlen 19
    }
    local len : display %`maxlen'.0fc e(N_pop)
    local len : list retok len
    local len : length local len
    local ++maxlen
    local ++len
    if `.c4wfmt' <= `len' & `len' <= `maxlen' {
        .c3 = `.c3' + `.c4wfmt' - `len'
        .c4 = `.c4' + `.c4wfmt' - `len'
        .c4wfmt = `len'
    }
}

> ----- end _coef_table__header.New4e -----
    - }
    - else {
        .New4r
    }
    - local props ```.Macro _coef_table_props'''

> ----- begin _coef_table__header.Macro -----
    - version 16
    - class exit `""`.result'(`0')'''
    = class exit `""'

> ----- end _coef_table__header.Macro -----
    = local props `""'
    - if `.is_sem' | `.is_gsem_group' | `:list posof "pushtitle" i
> n props' {
    = if 0 | 0 | 0 {
        .prop_pushtitle = 1
    }
    - if (`.is_demand' | `.is_htedid') {
    = if (0 | 0) {

```

```

        .prop_pushtitle = 1
    }
    - if (`is_bmareg') {
= if (0) {
        .prop_pushtitle = 1
    }
    - if `is_sem' | `is_gsem_group' | `:list posof "groupvar" in
> props' {
        = if 0 | 0 | 0 {
            .prop_groupvar = 1
        }
        - local list tobit cpoisson ivtobit metobit xttobit
        - local list2 metobit
        - if `:list posof ``.cmd''' in list' | `:list posof ``.cmd2''' i
> n list2' | `:list posof "censlimits" in props' {
            = if 0 | 0 | 0 {
                .prop_censlimits = !missing(e(N_unc))
            }
            - local list intreg tobit cpoisson ivtobit eintreg xtintreg xt
> tobit
                - local list2 meintreg metobit stintreg
                - if `:list posof ``.cmd''' in list' | `:list posof ``.cmd2''' i
> n list2' | `:list posof "censored" in props' {
                    = if 0 | 0 | 0 {
                        .prop_censored = !missing(e(N_unc))
                    }
                    - if `:list posof "nuncensored" in props' {
= if 0 {
                        .prop_nuncensored = 1
                    }
                    - local list eintreg
                    - if `:list posof ``.cmd''' in list' | `:list posof "iselected"
> in props' {
                        = if 0 | 0 {
                            .prop_iselected = 1
                        }
                        - local list eregress eprobit eoprobit
                        - if `:list posof ``.cmd''' in list' | `:list posof "selected"
> in props' {
                            = if 0 | 0 {
                                .prop_selected = 1
                            }
                            - if `:list posof "groups" in props' {
= if 0 {
                                .prop_groups = 1
                                .groupvar = ``e(repanvar)'''

```

```

        .intmethod = "`e(reintmethod)'"
        .intpoints = e(reintpoints)
    }
    - local list xtreg xtprobit xtlogit xtcloglog xtfrontier xtint
> reg xtnbreg
    - local list `list' xtpoisson xttobit xtmlogit xtrc
    = local list xtreg xtprobit xtlogit xtcloglog xtfrontier xtint
> reg xtnbreg xtpoisson xttobit xtmlogit xtrc
    - local list2 xtlogit xtprobit xtologit xtmlogit
    - if `:list posof "`cmd'" in list' | `:list posof "`cmd2'" i
> n list2' {
    = if 0 | 0 {
        .prop_groups = 1
        .groupvar = "`e(ivar)'"
        .intmethod = "`e(intmethod)'"
        .intpoints = e(n_quad)
    }
    - if "`cmd'" == "xtreg" | "`cmd2'" == "xtlogit" {
    = if "nbreg" == "xtreg" | "" == "xtlogit" {
        .is_r2_p = 0
    }
    - if `:list posof "balance" in props' {
    = if 0 {
        .prop_balance = 1
    }
    - .left = {}
    - .right = {}
}

> ----- end _coef_table__header.new -----
    - if ("`rclass'" != "") {
    = if (""" != "") {
        _return restore `rhold'
        .`obj'.New4r
    }
    - if ("`head1left'" != "") {
    = if (""" != "") {
        .`obj'.lalignla = `head1left'
    }
    - if ("`head1right'" != "") {
    = if (""" != "") {
        .`obj'.ralignla = `head1right'
    }
    - if ("`head2left'" != "") {
    = if (""" != "") {
        .`obj'.lalignr = `head2left'
    }
}

```

```

- if ("`head2right'" != "") {
= if ("'" != "") {
  .`obj'.ralignr = `head2right'
}
- if ("`width'" != "") {
= if ("'" != "") {
  .`obj'.width = `width'
}
- .`obj'.build
= .__000007.build

> ----- begin _coef_table__header.build -----
- version 16
- if ```.title'" == "" {
= if `"' == "" {
- .title = ```.Macro title'"

> ----- begin _coef_table__header.Macro -----
- version 16
- class exit ````result'(`0')'"
= class exit `"Negative binomial regression"'

> ----- end _coef_table__header.Macro -----
= .title = `"Negative binomial regression"'
- .title2 = ```.Macro title2'"

> ----- begin _coef_table__header.Macro -----
- version 16
- class exit ````result'(`0')'"
= class exit `"'

> ----- end _coef_table__header.Macro -----
= .title2 = `"'
- }
- if `.is_areg' {
= if 0 {
  local absvar = abbrev("`e(absvar)'" , 22)
  .title2 = "Absorbed variable: {res:`absvar'}"
}
- if `.prop_pushtitle' {
= if 0 {
  if `.is_svy' & ("`.cmd'" == "clogit" | "`.cmd'" == "xtmlogit"
> " | "`.cmd2'" == "xtmlogit") {
  }
  else if `.is_stintcox' | `.is_bmareg' {
    .PushLeft ````title'"

```

```

.PushLeft ```.title2'''
.title = ""
.title2 = ""
}
else {
.PushLeft ```.title'''
if `.is_areg' {
.PushLeft ```.title2'''
}
.title = ""
.title2 = ""
}
}
- if `.prop_groupvar' | `.is_gsem_group' {
= if 0 | 0 {
local groupvar ```.Macro groupvar'''
if (```.groupvar''' != "" & `.is_stintcox') {
local s = cond(`:word count `groupvar''>1, "s", "")
local grpvar = abbrev("`groupvar'",`.ablen')
.PushLeft "Strata variable`s': {res}`grpvar'"
}
else if ```.groupvar''' != "" {
local grpvar = abbrev("`groupvar'",`.ablen')
.PushLeft "Grouping variable: {res}`grpvar'"
}
}
- if (`.is_stintcox') {
= if (0) {
.PushLeft ""
local ndep = e(k_dv)
if (`ndep'==2) {
tokenize `e(depvar)'
local ltime `1'
local rtime `2'
.PushLeft "Event-time interval:"
.PushLeft "  Lower endpoint: {res}`ltime'"
.PushLeft "  Upper endpoint: {res}`rtime'"
if ``e(strata)''' != "" {
.PushLeft ""
}
}
}
else {
.PushLeft "ID variable: {res}`e(id)'"
.PushLeft "Examination time: {res}`e(depvar)'"
.PushLeft "Status indicator: {res}`e(status)'"
}
}

```



```

    }
    - if `.is_svy' {
= if 0 {
    .BuildLeft4svy
    }
    - else if `.is_prefix' {
= else if 0 {
    if !`.is_margins' {
    .Build4strata
    }
    }
    - if (`.is_tab' | `.is_sum') & !missing(`.result'(N_stdize)) {
= if (0 | 0) & !missing(e(N_stdize)) {
    .PushLeft "N. of std strata" " = " "as res" %`.c2wfmt'.0fc `
> .result'(N_stdize)
    }
    - .Build4tvar
    _____
> _____ begin _coef_table__header.Build4tvar _____
    - version 16
    - if `.opt_tvar' == 0 {
= if 1 == 0 {
    exit
    }
    - if "`Macro tvar'" == "" {
    _____
> _____ begin _coef_table__header.Macro _____
    - version 16
    - class exit ```.result'(`0')'''
= class exit `'''
    _____
> _____ end _coef_table__header.Macro _____
    = if "" == "" {
    - exit
    _____
> _____ end _coef_table__header.Build4tvar _____
    - if (!missing(`.Scalar N') & `.is_demandn'==0 & !`.is_bmareg'
> ) {
    _____
> _____ begin _coef_table__header.Scalar _____
    - version 16
    - class exit ```= `.result'(`0')'''
= class exit `"221"'
    _____
> _____ end _coef_table__header.Scalar _____
    = if (!missing(221) & 0==0 & !0) {

```

```

- .PushRight "Number of obs" " = " "as res" %`.c4wfmt'.0fc `.S
> calar N'
_____
> _____ begin _coef_table__header.Scalar _____
- version 16
- class exit ```= `.result'(`0')''
= class exit ``221''
_____
> _____ end _coef_table__header.Scalar _____
= .PushRight "Number of obs" " = " "as res" %10.0fc 221
_____
> _____ begin _coef_table__header.PushRight _____
- version 16
- gettoken tok rest : 0
- gettoken eq rest : rest
- gettoken dias rest : rest
- if `:length local rest' {
= if 12 {
- local direst = ustrltrim("` : di `rest'")
= local direst = ustrltrim("      221")
- .ralignr = max(`.ralignr', udstrlen("`direst'"))
= .ralignr = max(0, udstrlen("221"))
- local len : di ``tok''
= local len : di `"Number of obs"
- local len : smcllength local len
- .lalignr = max(`.lalignr', `len')
= .lalignr = max(0, 13)
- .right.Arrpush \`.C3' ``{lalign \`.lalignr':`tok}'" \`.C4
> ' ``eq' ``dias' "{ralign \`.ralignr':`direst}"
= .right.Arrpush \`.C3' ``{lalign \`.lalignr':Number of obs}"
> \`.C4' " = " as res "{ralign \`.ralignr':221}"
- }
- else if `:length local tok' {
= else if 13 {
. right.Arrpush \`.C3' ``tok''
}
- else {
. right.Arrpush
}
_____
> _____ end _coef_table__header.PushRight _____
- }
- if `.is_demandn' {
= if 0 {
if (e(N_estat_dsys)==.|``e(N_estat_dsys)''==) {
.PushRight "Number of obs" " = " "as res" %`.c4wfmt'.0fc `.S

```

```

> calar N'
    }
    else {
      .PushRight "Number of obs" " = " "as res" %`.c4wfmt'.0fc e(N
> _estat_dsys)
    }
    }
    - if `.is_demand' {
    = if 0 {
      .PushRight "Number of goods" " = " "as res" %`.c4wfmt'.0fc e
> (n_goods)
      .PushLeft "Expenditure variable: {result:`e(expenditures)'}"
    }
    - if `.is_demand' {
    = if 0 {
      if (`"`e(piconstant)'"'!="") {
      .PushRight "Price index constant" " = " "as res" %`.c2wfmt'.
> 4gc e(piconstant)
      .PushLeft ""
    }
    if (e(n_demos)>0) {
      local scaling = strproper("`e(demo_type)'"')
      .PushLeft `"Demographic method: {text:`scaling'}"'
      .PushRight "Number of demographics" " = " "as res" %`.c2wfmt
> '.0fc e(n_demos)
    }
    .PushLeft ""
    local rsquare = strproper("`e(r2_type)'"')
    .PushLeft `""rsquare' R2 of model for"'
    local kdem = e(k_eq_old) + 1
    local labels `""e(labels)'"'
    forvalues i=1/`kdem' {
      if (`""labels'"'=="") {
      .PushLeft " Good `i'" " = " "as res" %`.c2wfmt'.4f e(r2_`i'
> )
    }
    else {
      local xdm: word `i' of `labels'
      .PushLeft " `xdm'" " = " "as res" %`.c2wfmt'.4f e(r2_`i')
    }
    }
    .PushLeft ""
    }
    - if `.is_bacon' {
    = if 0 {
      .PushLeft "ATET" " = " "as res" %`.c2wfmt'.0g `e(atet)'

```

```

        .PushRight "Number of groups" " = " "as res" %`.c4wfmt'.0fc
> `e(N_groups)'
        .PushRight "Number of cohorts" " = " "as res" %`.c4wfmt'.0fc
> `e(N_timing)'
    }
    - if `.is_bmareg' {
    = if 0 {
        local heredlab = proper("`e(heredity)')")
        .PushLeft "`e(sampling_lab)'"
        if "`heredlab'" != "" {
            .PushLeft "Heredity: {txt}`heredlab'"
            if !missing(e(burnin)) {
                .PushLeft ""
            }
        }
        else if !missing(e(burnin)) {
            .PushLeft ""
            .PushLeft ""
        }
        .PushLeft ""
        .PushLeft "Priors:"
        .PushLeft "  Models: {txt}`e(mprior_lab)'"
        .PushLeft "  Cons.: Noninformative"
        .PushLeft "  Coef.: Zellner's g"
        .PushLeft "      g: {txt}`e(gprior_lab)'"
        .PushLeft "  sigma2: Noninformative"
        if !missing(e(burnin)) {
            .PushLeft
            .PushLeftRalign "{help j_bmaregress_sampcorr:Sampling correl
> ation}" " = " "as res" %`.c4wfmt'.4f e(sampcorr)
        }
        .PushRight "No. of obs" " = " "as res" %`.c4wfmt'.0fc `.Scal
> ar N'
        .PushRight "No. of predictors" " = " "as res" %`.c4wfmt'.0fc
> e(p)
        .PushRightRalign "Groups" " = " "as res" %`.c4wfmt'.0fc e(p_
> groups)
        .PushRightRalign "Always" " = " "as res" %`.c4wfmt'.0fc e(p_
> always)
        .PushRight "No. of models" " = " "as res" %`.c4wfmt'.0fc e(k
> _models)
        .PushRightRalign "For CPMP >= .9" " = " "as res" %`.c4wfmt'.
> 0fc e(k_models_cpmp)
        .PushRight "Mean model size" " = " "as res" %`.c4wfmt'.3f e(
> msize_mean)
        if !missing(e(burnin)) {

```

```

        .PushRight "Burn-in" " = " "as res" %`.c4wfmt'.0fc e(burnin)
        .PushRight "MCMC sample size" " = " "as res" %`.c4wfmt'.0fc
> e(mcmcsize)
        .PushRight "Acceptance rate" " = " "as res" %`.c4wfmt'.4f e(
> arate)
    }
    else {
    if "`heredlab'" != "" {
        .PushRight
    }
        .PushRight
    }
    if !missing(e(shrinkage)) {
        .PushRight "Shrinkage, g/(1+g)" " = " "as res" %`.c4wfmt'.4f
> e(shrinkage)
    }
    else {
        .PushRight ""
    }
    if (e(sigma2) < 1e-3) {
        local sigfmt 8.4g
    }
    else {
        local sigfmt `.c4wfmt'.3f
    }
        .PushRight "Mean sigma2" " = " "as res" %`sigfmt' e(sigma2)
    }
    - if `.is_htedid' {
    = if 0 {
        .PushRight "Number of panels" " = " "as res" %`.c4wfmt'.0fc
> e(N_panels)
    }
    - if !missing(`.Scalar N_sub') & `.is_stintcox' {


---


> 

---

 begin _coef_table__header.Scalar 

---


    - version 16
    - class exit ``'= `.result'(`0')'''
    = class exit `",'''


---


> 

---

 end _coef_table__header.Scalar 

---


    = if !missing(.) & 0 {
        .PushRight "Number of subjects" " = " "as res" %`.c4wfmt'.0f
> c `.Scalar N_sub'
    }
    - if `.is_areg' {
    = if 0 {

```

```

        .PushRight "No. of categories" " = " "as res" %`.c4wfmt'.0fc
> e(k_absorb)
    }
    - if `.is_sem' | `.is_gsem_group' {
    = if 0 | 0 {
        if `""Macro groupvar"" != "" {
        .PushRight "Number of groups" " = " "as res" %`.c4wfmt'.0fc
> `.Scalar N_groups'
    }
    }
    - if `.is_sem' & !`.is_svy' {
    = if 0 & !0 {
        .PushLeft `"Estimation method: {res:`.Macro method'}""
        }
    - if `.prop_selected' {
    = if 0 {
        .Build4selected
        }
    - if `.is_svy' {
    = if 0 {
        .BuildRight4svy
        if `.prop_censored' {
        .PushRight
        .Build4censobs "header"
        .PushRight
        }
        }
    - else {
    - if `.is_margins' {
    = if 0 {
        .Build4subpop
        }
    - if `.prop_iselected' {
    = if 0 {
        .Build4title
        .Build4iselected
        }
    - if `.prop_censored' {
    = if 0 {
        .Build4title
        .Build4censobs
        if !`.is_prefix' {
        .PushRight
        }
        }
    - .Build4cluster

```

```

> ----- begin _coef_table__header.Build4cluster -----
- version 16
- if `.is_multclust' {
= if 0 {
  .Build4mclust
  exit
}
- if missing(`.Scalar N_clust') {

> ----- begin _coef_table__header.Scalar -----
- version 16
- class exit ```= `.result'(`0')''''
= class exit `".'"'

> ----- end _coef_table__header.Scalar -----
= if missing(.) {
- exit

> ----- end _coef_table__header.Build4cluster -----
- .Build4reps

> ----- begin _coef_table__header.Build4reps -----
- version 16
- if !missing(`.Scalar N_reps') {

> ----- begin _coef_table__header.Scalar -----
- version 16
- class exit ```= `.result'(`0')''''
= class exit `".'"'

> ----- end _coef_table__header.Scalar -----
= if !missing(.) {
  .PushRight "Replications" " = " "as res" %`.c4wfmt'.0fc `.`
> Scalar N_reps'
  if `.prop_censored' & !`.is_svy' {
    .PushRight
  }
}

> ----- end _coef_table__header.Build4reps -----
- }
- if `.prop_groups' {
= if 0 {
  .Build4groups
}

```

```

- if `.is_stintcox' & "`e(strata)'" != "" {
= if 0 & "" != "" {
  if `ndep' == 2 {
    .PushRight
  }
}
- .Build4modeltest

```

---

```

> ----- begin _coef_table__header.Build4modeltest -----
- version 16
- if ``.result'" != "e" {
= if "e" != "e" {
  exit
}
- if `.opt_modeltest' == 0 {
= if 1 == 0 {
  exit
}
- if ```.Macro k_eq_model'"' == "0" {

```

---

```

> ----- begin _coef_table__header.Macro -----
- version 16
- class exit `""`.result'(`0')'"'
= class exit `1'"'

```

---

```

> ----- end _coef_table__header.Macro -----
= if `1'"' == "0" {
  exit
}
- if missing(`.Scalar df_m') {

```

---

```

> ----- begin _coef_table__header.Scalar -----
- version 16
- class exit ```= `.result'(`0')'"'
= class exit `8'"'

```

---

```

> ----- end _coef_table__header.Scalar -----
= if missing(8) {
  exit
}
- if `""`.Macro chi2'"' != "" | `""`.Macro df_r'"' == "" {

```

---

```

> ----- begin _coef_table__header.Macro -----
- version 16
- class exit `""`.result'(`0')'"'
= class exit `252.0138019548383'"'

```



```

> _____ end _coef_table__header.Macro _____
> _____ begin _coef_table__header.Macro _____
    - version 16
    - class exit ```.result'(`0')'''
    = class exit `'''
> _____ end _coef_table__header.Macro _____
    = if `"252.0138019548383"' != "" | `''' == "" {
    - .Build4chi2test
> _____ begin _coef_table__header.Build4chi2test _____
    - version 16
    - local type `e(chi2type)'
    = local type Wald
    - if ``type''' == "" {
    = if `"Wald"' == "" {
        local type Wald
    }
    - local didf = strtrim("` : di e(df_m)')")
    = local didf = strtrim("8")
    - if !missing(e(chi2)) {
    - .PushRight ``type' chi2({res:`didf'})" " = " "as res" %`
> .c4wfmt'.2f e(chi2)
    = .PushRight "Wald chi2({res:8})" " = " "as res" %10.2f e(
> chi2)
> _____ begin _coef_table__header.PushRight _____
    - version 16
    - gettoken tok rest : 0
    - gettoken eq rest : rest
    - gettoken dias rest : rest
    - if `:length local rest' {
    = if 15 {
    - local direst = ustrltrim("` : di `rest'")
    = local direst = ustrltrim("    252.01")
    - .ralignr = max(`.ralignr', udstrlen("`direst'))
    = .ralignr = max(3, udstrlen("252.01"))
    - local len : di ``tok'''
    = local len : di `"Wald chi2({res:8})"'
    - local len : smcllength local len
    - .lalignr = max(`.lalignr', `len')
    = .lalignr = max(13, 12)
    - .right.Arrpush `\.C3' ``{lalign `\.lalignr':`tok}'" \
> `\.C4' ``eq' `dias' ``{ralign `\.ralignr':`direst}'"

```

```

= .right.Arrpush \.C3' ``{lalign \.lalignr':Wald chi2({r
> es:8}})}```.C4' " = " as res "{ralign \.ralignr':252.01}"
- }
- else if `:length local tok' {
= else if 18 {
  .right.Arrpush \.C3' ``tok'``
  }
- else {
  .right.Arrpush
  }
}

> ----- end _coef_table__header.PushRight -----
- .PushRight "Prob > chi2" " = " "as res" %`.c4wfmt'.4f ch
> i2tail(e(df_m),e(chi2))
= .PushRight "Prob > chi2" " = " "as res" %10.4f chi2tail(
> e(df_m),e(chi2))

> ----- begin _coef_table__header.PushRight -----
- version 16
- gettoken tok rest : 0
- gettoken eq rest : rest
- gettoken dias rest : rest
- if `:length local rest' {
= if 33 {
- local direst = ustrltrim("` : di `rest'")
= local direst = ustrltrim(" 0.0000")
- .ralignr = max(`.ralignr', udstrlen("`direst'"))
= .ralignr = max(6, udstrlen("0.0000"))
- local len : di ``tok'``
= local len : di `"Prob > chi2"`
- local len : smcllength local len
- .lalignr = max(`.lalignr', `len')
= .lalignr = max(13, 11)
- .right.Arrpush \.C3' ``{lalign \.lalignr':`tok'}``` \
> \.C4' ``eq'`` `dias' "{ralign \.ralignr':`direst'}`"
= .right.Arrpush \.C3' ``{lalign \.lalignr':Prob > chi2}
> ``.C4' " = " as res "{ralign \.ralignr':0.0000}"
- }
- else if `:length local tok' {
= else if 11 {
  .right.Arrpush \.C3' ``tok'``
  }
- else {
  .right.Arrpush
  }
}

```

```

> ----- end _coef_table__header.PushRight -----
      - }
      - else {
        local j_robust "{help j_robustsingular##|_new:`type' chi
> 2(`didf'))}"
        .PushRight "`j_robust'" " = " "as res" %`.c4wfmt's "
        .PushRight "Prob > chi2" " = " "as res" %`.c4wfmt's "
      }
    -----

> ----- end _coef_table__header.Build4chi2test -----
      - }
      - else {
        .Build4Ftest
      }
    -----

> ----- end _coef_table__header.Build4modeltest -----
      - .Build4r2
    -----

> ----- begin _coef_table__header.Build4r2 -----
      - version 16
      - if "`'.result'" != "e" {
      = if "e" != "e" {
        exit
      }
      - if !missing(e(r2)) {
        .PushRight "R-squared" " = " "as res" %`.c4wfmt'.4f e(r2)
      }
      - if !missing(e(r2_p)) & `.is_r2_p' {
      = if !missing(e(r2_p)) & 1 {
      - .PushRight "Pseudo R2" " = " "as res" %`.c4wfmt'.4f e(r2_p
> )
      = .PushRight "Pseudo R2" " = " "as res" %10.4f e(r2_p)
    -----

> ----- begin _coef_table__header.PushRight -----
      - version 16
      - gettoken tok rest : 0
      - gettoken eq rest : rest
      - gettoken dias rest : rest
      - if `:length local rest' {
      = if 15 {
      - local direst = ustrltrim("` : di `rest'")
      = local direst = ustrltrim(" 0.1289")
      - .ralignr = max(`.ralignr', udstrlen("`direst'"))
      = .ralignr = max(6, udstrlen("0.1289"))
      - local len : di ``tok'""
      = local len : di `"Pseudo R2"'

```

```

- local len : smcllength local len
- .lalignr = max(`.lalignr', `len')
= .lalignr = max(13, 9)
- .right.Arrpush `'.C3' ``{lalign `'.lalignr':`tok}'``' `'.
> C4' ``eq'' `dias' "{ralign `'.ralignr':`direct'}"
= .right.Arrpush `'.C3' ``{lalign `'.lalignr':Pseudo R2}'``' `
> .C4' " = " as res "{ralign `'.ralignr':0.1289}"
- }
- else if `:length local tok' {
= else if 9 {
. right.Arrpush `'.C3' ```tok'``'
}
- else {
. right.Arrpush
}
}

> ----- end _coef_table__header.PushRight -----
- }
- if !missing(e(r2_a)) {
. PushRight "Adj R-squared" " = " "as res" %`.c4wfmt'.4f e(
> r2_a)
}
- if !missing(e(rmse)) {
. PushRight "Root MSE" " = " "as res" %`.c4wfmt'.4f e(rmse)
}
}

> ----- end _coef_table__header.Build4r2 -----
- .Build4title
}

> ----- begin _coef_table__header.Build4title -----
- version 16
- if `.opt_septitle' {
= if 0 {
exit
}
- if ```.title''' == "" {
= if ``Negative binomial regression''' == "" {
exit
}
- if `.left.arnels' {
= if 0 {
exit
}
- local title `.title'
= local title Negative binomial regression
- local title2 `.title2'

```

```

= local title2
- local len : smcllength local title
- local len2 : smcllength local title2
- if `len2' > `len' {
= if 0 > 28 {
    local len = `len2'
  }
- if `len' < (`c3' - 2) - (`c1' - 1) {
= if 28 < (49 - 2) - (1 - 1) {
- PushLeft ```.title'''
= PushLeft `"Negative binomial regression"'

```

---

```

> ----- begin _coef_table__header.PushLeft -----
- version 16
- gettoken tok rest : 0
- gettoken eq rest : rest
- gettoken dias rest : rest
- if `:length local rest' {
= if 0 {
    local direst : di `rest'
    local direst = strtrim("`direst'")
    local len : smcllength local direst
    .ralignl = max(`.ralignl', `len')
    local len : di ``tok'''
    local len : smcllength local len
    .lalignl = max(`.lalignl', `len')
    .left.Arrpush \`.C1' `{lalign \.lalignl':`tok'}" \`.C
> 2' ``eq' ``dias' "{ralign \.ralignl':`direst'}"
    }
- else if `:length local tok' {
= else if 28 {
- .left.Arrpush \`.C1' ``tok'''
= .left.Arrpush \`.C1' `"Negative binomial regression"'
- }
- else {
    .left.Arrpush
  }

```

---

```

> ----- end _coef_table__header.PushLeft -----
- .title = ""
- if `len2' {
= if 0 {
    PushLeft ```.title2'''
    .title2 = ""
  }
- }

```

```

- .opt_septitle = 1


---


> 

---

 end _coef_table__header.Build4title 

---


- if `.prop_censlimits' {
= if 0 {
  .Build4censlimits 1
}
- .c2 = `.lalignl' + 1
= .c2 = 0 + 1
- .c3 = `.width' - `.lalignr' - `.ralignr' - 3 + 1
= .c3 = 78 - 13 - 6 - 3 + 1
- .c4 = `.c3' + `.lalignr'
= .c4 = 57 + 13
- if `.is_gsem' {
= if 0 {
  if `.opt_dvheader' {
    gsem_depvar_header, left(`.left.objkey') right(`.right.objke
> y') c3(`.c3') c4(`.c4') c4wfmt(`.c4wfmt')
  }
}
- .Build4crit


---


> 

---

 begin _coef_table__header.Build4crit 

---


- version 16
- if ``.result'' != "e" {
= if "e" != "e" {
  exit
}
- local offset 1
- local hascrit = !missing("`e(ll)''') & !missing("`e(crit
> type)''')
= local hascrit = !missing("`-360.8365614162979''') & !missin
> g(`"log pseudolikelihood"')
- if inlist("`e(dispers)'", "mean", "constant") {
= if inlist("mean", "mean", "constant") {
- if `hascrit' {
= if 1 {
- local ++offset
- local crtype ``e(crittype)''
= local crtype `"log pseudolikelihood"
- local clen : length local crtype
- .c2 = `clen' + 2
= .c2 = 20 + 2
- }
- local didisp = strtrim("` : di e(dispers)''")
= local didisp = strtrim("mean")

```

```

- local DISPERS `""Dispersion: {res:`didisp'}""'
= local DISPERS `""Dispersion: {res:mean}""'
- }
- .FillLeft `offset'
= .FillLeft 2

> ----- begin _coef_table__header.FillLeft -----
- version 16
- args offset
- local kr = `.right.arnels'
= local kr = 4
- local kl = `.left.arnels'
= local kl = 1
- local diff = `kr' - `kl' - 0`offset'
= local diff = 4 - 1 - 02
- forval i = 1/`diff' {
= forval i = 1/1 {
- .left.Arrpush ""
- }

> ----- end _coef_table__header.FillLeft -----
- if `""DISPERS""' != "" {
= if `""Dispersion: {res:mean}""' != "" {
- .PushLeft `DISPERS'
= .PushLeft "Dispersion: {res:mean}"

> ----- begin _coef_table__header.PushLeft -----
- version 16
- gettoken tok rest : 0
- gettoken eq rest : rest
- gettoken dias rest : rest
- if `:length local rest' {
= if 0 {
local direst : di `rest'
local direst = strtrim("`direst'")
local len : smcllength local direst
.ralignl = max(`.ralignl', `len')
local len : di `""tok'""'
local len : smcllength local len
.lalignl = max(`.lalignl', `len')
.left.Arrpush `\.C1' `"{lalign \\.lalignl':`tok'}" `\.C
> 2' ""eq"" `dias' "{ralign \\.ralignl':`direst'}"
}
- else if `:length local tok' {
= else if 22 {
- .left.Arrpush `\.C1' `""tok'""'

```





```

- local len : smcllength local len
- .lalignla = max(`.lalignla', `len')
= .lalignla = max(0, 20)
- .left.Arrpush `'.C1' ``{lalign `'.lalignla':`tok}'" `'.
> C2a' ``eq'" `dias' "{ralign `'.ralignla':`direst'}"
= .left.Arrpush `'.C1' ``{lalign `'.lalignla':Log pseudolike
> lihood}'" `'.C2a' " = " as res "{ralign `'.ralignla':-360.83656}"
- }
- else if `:length local tok' {
= else if 20 {
. left.Arrpush `'.C1' ``"tok'"
}
- else {
. left.Arrpush
}

> _____ end _coef_table__header.PushLeftRalign _____
- }
- }

> _____ end _coef_table__header.Build4crit _____
- .c2 = `.lalignl' + 1
= .c2 = 0 + 1
- .c2a = `.lalignla' + 1
= .c2a = 20 + 1
- .c3 = `.width' - `.lalignr' - `.ralignr' - 3 + 1
= .c3 = 78 - 13 - 6 - 3 + 1
- .c4 = `.c3' + `.lalignr'
= .c4 = 57 + 13

> _____ end _coef_table__header.build _____
- if ("`head2start'" != "") {
= if (" " != " ") {
. `obj'.c3 = `head2start'
. `obj'.c4 = `head2start' + `.`obj'.lalignr'
. `obj'.width = `head2start' + `.`obj'.lalignr' + `.`obj'.ralig
> nr' + 2
}
- if (`has_obj' == 0) {
= if (0 == 0) {
- .`obj'.display
= .__000007.display

> _____ begin _coef_table__header.display _____
- version 16
- .ShowRules

```

```

> ----- begin _coef_table__header.ShowRules -----
- version 16
- if ``.result'' != "e" {
= if "e" != "e" {
  exit
}
- if `.opt_rules' == 0 {
= if 1 == 0 {
  exit
}
- if !inlist("`.cmd'", "logistic", "logit", "probit") {
= if !inlist("nbreg", "logistic", "logit", "probit") {
- exit
}

> ----- end _coef_table__header.ShowRules -----
- local nl = `.left.arnels'
= local nl = 4
- local nr = `.right.arnels'
= local nr = 4
- local K = max(`nl', `nr')
= local K = max(4,4)
- `.newline'
= display

- if ``.title'' != "" {
= if `""' != "" {
  display as txt ``.title''
  if ``.title2'' != "" {
    display as txt ``.title2''
  }
  if `K' {
    display
  }
}
- forval i = 1/`K' {
= forval i = 1/4 {
- display as smcl as txt `.left['i']' as txt `.right['i']'
}

> ----- begin _coef_table__header.C1 -----
- version 16
- class exit `"_col(`.c1')"'
= class exit `"_col(1)"'

> ----- end _coef_table__header.C1 -----

```

```

> ----- begin _coef_table__header.C3 -----
      - version 16
      - class exit `"_col(`.c3')"'
      = class exit `"_col(57)"'
-----

> ----- end _coef_table__header.C3 -----
-----

> ----- begin _coef_table__header.C4 -----
      - version 16
      - class exit `"_col(`.c4')"'
      = class exit `"_col(70)"'
-----

> ----- end _coef_table__header.C4 -----
      = display as smcl as txt _col(1) `Negative binomial regressio
> n"' as txt _col(57) `{lalign 13:Number of obs}' _col(70) " = " as res "{ra
> lign 6:221}"
Negative binomial regression                                Number of obs =    221
      - }
      - display as smcl as txt `.left['i']' as txt `.right['i']'
-----

> ----- begin _coef_table__header.C3 -----
      - version 16
      - class exit `"_col(`.c3')"'
      = class exit `"_col(57)"'
-----

> ----- end _coef_table__header.C3 -----
-----

> ----- begin _coef_table__header.C4 -----
      - version 16
      - class exit `"_col(`.c4')"'
      = class exit `"_col(70)"'
-----

> ----- end _coef_table__header.C4 -----
      = display as smcl as txt as txt _col(57) `{lalign 13:Wald ch
> i2({res:8}))"' _col(70) " = " as res "{ralign 6:252.01}"
                                                                Wald chi2(8) = 252.01
      - }
      - display as smcl as txt `.left['i']' as txt `.right['i']'
-----

> ----- begin _coef_table__header.C1 -----
      - version 16
      - class exit `"_col(`.c1')"'
      = class exit `"_col(1)"'
-----

> ----- end _coef_table__header.C1 -----
-----

```

```

> ----- begin _coef_table__header.C3 -----
      - version 16
      - class exit `"_col(`.c3')"'
      = class exit `"_col(57)"'
-----

> ----- end _coef_table__header.C3 -----
-----

> ----- begin _coef_table__header.C4 -----
      - version 16
      - class exit `"_col(`.c4')"'
      = class exit `"_col(70)"'
-----

> ----- end _coef_table__header.C4 -----
      = display as smcl as txt _col(1) `"Dispersion: {res:mean}"' as
> txt _col(57) `"{lalign 13:Prob > chi2}"' _col(70) " = " as res "{ralign 6:0
> .0000}"
Dispersion: mean                               Prob > chi2   = 0.0000
      - }
      - display as smcl as txt `.left['i']' as txt `.right['i']'
-----

> ----- begin _coef_table__header.C1 -----
      - version 16
      - class exit `"_col(`.c1')"'
      = class exit `"_col(1)"'
-----

> ----- end _coef_table__header.C1 -----
-----

> ----- begin _coef_table__header.C2a -----
      - version 16
      - class exit `"_col(`.c2a')"'
      = class exit `"_col(21)"'
-----

> ----- end _coef_table__header.C2a -----
-----

> ----- begin _coef_table__header.C3 -----
      - version 16
      - class exit `"_col(`.c3')"'
      = class exit `"_col(57)"'
-----

> ----- end _coef_table__header.C3 -----
-----

> ----- begin _coef_table__header.C4 -----
      - version 16
      - class exit `"_col(`.c4')"'
      = class exit `"_col(70)"'
-----

```

```

> ----- end _coef_table__header.C4 -----
      = display as smcl as txt _col(1) "{lalign 20:Log pseudolikeli
> hood}" _col(21) " = " as res "{ralign 10:-360.83656}" as txt _col(57) "{la
> lign 13:Pseudo R2}" _col(70) " = " as res "{ralign 6:0.1289}"
Log pseudolikelihood = -360.83656           Pseudo R2       = 0.1289
      - }
> ----- end _coef_table__header.display -----
      - }
      - sreturn local width_header `.`obj'.width'
      = sreturn local width_header 78
      - sreturn local head2_right `.`obj'.ralignr'
      = sreturn local head2_right 6
      - sreturn local head2_left `.`obj'.lalignr'
      = sreturn local head2_left 13
      - sreturn local head2_start `.`obj'.c3'
      = sreturn local head2_start 57
      - sreturn local head1_right `.`obj'.ralignla'
      = sreturn local head1_right 10
      - sreturn local head1_left `.`obj'.lalignla'
      = sreturn local head1_left 20
      - sreturn local head1_start `.`obj'.c1'
      = sreturn local head1_start 1
> ----- end _coef_table__header -----
      - di
      - }
      - _coef_table, `first' level(`level') `neqopt' `plus' `noskip' `sh
> oweqns' `diopts' `options' `notest'
      = _coef_table, level(95)
> ----- begin _coef_table -----
      - version 11
      - local vv : di "version " string(max(11,_caller())) ", missing:
> "
      - if (!c(noisily) & c(coeftabresults) == "off") {
        exit
      }
      - _check_eclass
      - syntax [, BMATrix(passthru) VMATrix(passthru) * ]
      - if "`e(b)'" == "" & "`e(V)'" == "" & "`bmatrix'`vmatrix'" == ""
> {
      = if "matrix" == "" & "matrix" == "" & "" == "" {
        exit
      }

```

```

- local mc_cmds contrast margins pwcompare pwmean
- local cmd ``e(cmd)''
= local cmd "nbreg"
- local keepmc : list cmd in mc_cmds
- if inlist("`cmd'", "pwcompare", "pwmean") {
= if inlist("nbreg", "pwcompare", "pwmean") {
    local groups GROUPS
    }
- if "`cmd'" == "regress" {
= if "nbreg" == "regress" {
    local beta Beta
    }
- if "`cmd'" == "gsem" {
= if "nbreg" == "gsem" {
    local fvignore fvignore(int 0)
    local flignore flignore
    }
- if "`cmd'" == "sem" {
= if "nbreg" == "sem" {
    local standardized STANDARDIZED
    local showginvariant SHOWGINvariant
    local nolabel NOLABel LABel NOFVLABel FVLABel
    local nofootnote NOFOOTnote
    local wrap wrap(numlist max=1) fwwrap(passthru)
    }
- if ``e(mi)''=="mi" {
= if ""=="mi" {
    local dftable DFTable
    local dfonly DFONLY
    local noclustreport NOCLUSTReport
    local pisematrix PISEMATrix(string)
    }
- syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st
> ring) DFMATrix(string) EQMATrix(string) ROWMATrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMATrix(string) MEMATrix(string) BSTDMATrix(string) CNSMATrix(string) P
> CLASSMATrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) `beta' `standardized' NOCI NOPValues `dftable' `dfonly' dfci DFPVal
> ues dfmissing `groups' COEFLegend selegend `fvignore' `flignore' `showginvar
> iant' `nolabel' `nofootnote' `wrap' sort `pisematrix' depname(string) COEFTi
> tle(string) coeftitle2(string) ptitle(string) cititle(string) NOMCLEgend noC
> NSReport FULLCNSReport `noclustreport' cformat(passthru) sformat(passthru) p
> format(passthru) NOFirst First SHOWEQns neq(integer -1) NODIPARM NOTEST SEP
> rator(integer 0) NOSKIP OFFSETONLY1 PPlus NOEQCHECK EFORMALL CITYPE(string) e
> qselect(string) noEFORMNOTE * ]
= syntax [, cmdextras BMATrix(string) VMATrix(string) EMATrix(st

```

```

> ring) DFMAtrix(string) EQMAtrix(string) ROWMAtrix(string) CIMATrix(string) R
> OWCFormat(string) ROWPFormat(string) ROWSFormat(string) NOROWCI MMATrix(stri
> ng) MVMAtrix(string) MEMATrix(string) BSTDMAtrix(string) CNSMATrix(string) P
> CLASSMAtrix(string) Level(cilevel) prefix(name) suffix(name) OFFSETlist(stri
> ng asis) NOCI NOPValues dfci DFPValues dfmissing COEFLegend selegend
> sort depname(string) COEFTitle(string) coeftitle2(string) ptitle(string
> ) cititle(string) NOMCLEgend noCNSReport FULLCNSReport cformat(passthru) sf
> ormat(passthru) pformat(passthru) NOFirst First SHOWEQns neq(integer -1) NOD
> IPARM NOTEST SEParator(integer 0) NOSKIP OFFSETONLY1 Plus NOEQCHECK EFORMALL
> CITYPE(string) eqselect(string) noEFORMNOTE * ]

```

```

- if "`wrap'" != "" {
= if "" != "" {
  opts_exclusive `"wrap(`wrap') `fvwrap'"
  local fvwrap fvwrap(`wrap')
}
- if "`nolabel'`label'" != "" {
= if "" != "" {
  opts_exclusive "`nolabel' `label'"
  opts_exclusive "`nofvlabel' `fvlabel'"
  opts_exclusive "`nolabel' `fvlabel'"
  opts_exclusive "`nofvlabel' `label'"
  if "`nolabel'" != "" {
    local fvlabel nofvlabel
  }
  else {
    local fvlabel fvlabel
  }
}
- else local fvlabel `nofvlabel' `fvlabel'
= else local fvlabel
- local type `coeflegend' `selegend'
= local type
- opts_exclusive "`type' `standardized'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- if (`rowmatrix'=="" & "`norowci"!="") {
= if (`"'=="" & "'!="") {
di as err "{bf:norowci} requires {bf:rowmatrix()}"
exit 198
}
- if (`dftable'!="" & "`norowci"!="") {
= if ("!=" & "'!="") {
di as err "{bf:norowci} is not allowed with {bf:`dftable'}"
exit 198
}
- if (`dfonly'!="" & "`norowci"!="") {
= if ("!=" & "'!="") {
di as err "{bf:norowci} is not allowed with {bf:`dfonly'}"
exit 198
}
- if (`noci'!="" & "`norowci"!="") {
= if ("!=" & "'!="") {
di as err "{bf:norowci} is not allowed with {bf:`noci'}"
exit 198
}
- if "`standardized' != "" & "`bstdmatrix' != "" {
= if "" != "" & "" != "" {
local beta beta
local standardized
}
- if `:length local type' == 0 {
= if 0 == 0 {
- local type `beta' `noci' `nopvalues' `dftable' `dfonly' `group
> s' `dfci' `dfpvalues'
= local type
- }
- opts_exclusive "`type'"
= opts_exclusive ""

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```



```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`first' `nofirst'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`first' `showeqns'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
    - local cnsreport `cnsreport' `fullcnsreport'
    = local cnsreport
    - opts_exclusive "`cnsreport'"
    = opts_exclusive ""

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'"" != "" {
    = while `""' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - if ```showeqns'"" != "" {
    = if `""' != "" {
        local nofirst nofirst
    }
    - if ```ematrix'`noeqcheck'"" == "" {
    = if `""' == "" {
    - if ```e(error)'"" == "matrix" {
    = if `""' == "matrix" {
        local ematrix e(error)
    }
    - }
    - _get_mcompare, `options'
    = _get_mcompare,

> ----- begin _get_mcompare -----
    - version 11
    - local cmd ```e(cmd)'""
    = local cmd ``nbreg'"
    - syntax [name(name=caller)] [, *]
    - local is_pwc = inlist("pwcompare", "`cmd'", "`caller'") | in
> list("pwmean", "`cmd'", "`caller'")
    = local is_pwc = inlist("pwcompare", "nbreg", "") | inlist("pw
> mean", "nbreg", "")
    - local is_marg = inlist("margins", "`cmd'", "`e(cmd2)'", "`ca

```

```

> ller'')
    = local is_marg = inlist("margins", "nbreg", "", "")
    - local OPTIONS NOADJust BONferroni DUNCAn DUNNett SCHeffe SID
> ak SNK TUKey ADJustall
    - if `is_pwc' {
    = if 0 {
        syntax [name] [, `OPTIONS' MCOMPare(string) *]
    }
    - else {
    - syntax [name] [, MCOMPare(string) *]
    - local suboption mcompare()
    - }
    - sreturn clear
    - sreturn local options ``options'''
    = sreturn local options `'''
    - if `is_pwc' {
    = if 0 {
        local METHOD `noadjust' `bonferroni' `duncan' `dunnett' `sch
> effe' `sidak' `snk' `tukey'
        local METHOD : list uniq METHOD
        opts_exclusive ``METHOD'''
        local ALL `adjustall'
        opts_exclusive ``noadjust' `ALL'''
    }
    - local 0 `", `mcompare'''
    = local 0 `", ""
    - capture syntax [, `OPTIONS']
    = capture syntax [, NOADJust BONferroni DUNCAn DUNNett SCHeffe
> SIDak SNK TUKey ADJustall]
    - if c(rc) {
        di as err "invalid mcompare() option;"
        di as err "method ``mcompare'' not allowed"
        error 198
    }
    - local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `dunn
> ett' `scheffe' `sidak' `snk' `tukey'
    = local METHOD
    - local METHOD : list uniq METHOD
    - local ALL `ALL' `adjustall'
    = local ALL
    - local ALL : list uniq ALL
    - opts_exclusive ``METHOD''' `suboption'
    = opts_exclusive `` mcompare()'

> _____ begin opts_exclusive _____
    - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- if "`METHOD'" == "" {
= if "" == "" {
- if "`ALL'" != "" {
= if "" != "" {
    di as err "multiple comparison method required with the adju
> stall option"
    exit 198
}
- local METHOD noadjust
- local noadjust noadjust
- }
- opts_exclusive "`ALL' `noadjust'" `suboption'
= opts_exclusive " noadjust" mcompare()

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts noadjust
- local n 0
- while ``opts' != "" {
= while `"noadjust"' != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

= if 1 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local EXCLUDE duncan dunnett snk tukey
- if `:list METHOD in EXCLUDE' {
= if 0 {
  if `is_pwc' {
    local opt : copy local METHOD
  }
  else {
    local opt "mcompare(`METHOD')"
  }
  local cmdlist anova manova regress mvreg pwmean
  if "`cmd'" == "pwcompare" {
    local cmd ``e(cmd_est)'''
  }
  if `is_marg' {
    local cmd margins
  }
  if !`:list cmd in cmdlist' | !`is_pwc' {
    di as err ``method `opt' is not allowed with results from `c
> md'''

    exit 322
  }
  if ``e(prefix)''' == "svy" {
    di as err ``method `opt' is not allowed with results using t
> he svy prefix'''

    exit 322
  }
  local vce ``e(vce)'''
  if !inlist("`vce'", "", "ols") {
    di as err ``method `opt' is not allowed with results using v
> ce(`vce')'''

    exit 322
  }
}

- local EXCLUDE duncan dunnett snk tukey scheffe
- if `:list METHOD in EXCLUDE' {
= if 0 {
  if ""ALL"" != "" {
    di as err ``option adjustall is not allowed with method `MET
> HOD'''

    exit 198
  }
}

```

```

- sreturn local method ``METHOD'''
= sreturn local method "noadjust"
- sreturn local adjustall ``ALL'''
= sreturn local adjustall `'''

> ----- end _get_mcompare -----
- local method ``s(method)'''
= local method "noadjust"
- local all ``s(adjustall)'''
= local all `'''
- local options ``s(options)'''
= local options `'''
- if ``method'" != "noadjust" {
= if "noadjust" != "noadjust" {
  local keepmc 1
}
- opts_exclusive "all" `groups'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- if ``method'" == "dunnett" {
= if "noadjust" == "dunnett" {
  opts_exclusive "method" `groups'"
}
- _get_diopts diopts options, `options' `fvwrap' `fvlabel'
= _get_diopts diopts options,

> ----- begin _get_diopts -----
- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist

```

```

- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``baselevels' `nobaselevels'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``emptycells' `noemptycells'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```



```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'

```

```

= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {

```

```

    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat ``sformat''
  local sformat `sformat(`sformat')`
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`pformat') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat ``pformat''
  local pformat `pformat(`pformat')`
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> _____ begin _get_diopts.Wrapon _____
- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive ``word' `width' fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> _____ begin opts_exclusive _____
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `""' != "" {
  local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }
}
> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'
    = c_local diopts
    - if `K' == 2 {
    = if 2 == 2 {
    - c_local `c_opts' ``options''
    = c_local options ``''
    - }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
}
> ----- end _get_diopts.GetDiopts -----
    - c_local `name1' ``diopts''
    = c_local diopts ``''
    - local snames : s(macros)
    - foreach sname of local snames {
    - local S_`sname' = s(`sname')
    = local S_level = s(level)

```

```

- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```



```

- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``baselevels' `nobaselevels'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``emptycells' `noemptycells'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts'' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts'' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`coeflegend' `selegend'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts'' != "" {
    = while `"' != "" {

```

```

        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts'" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts'" != "" {
= while `"" != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells

```

```

- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
  local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
  local baselevels baselevels
}
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
  local emptycells noemptycells
}
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
  local omitted noomitted
}
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
  local lstretch nolstretch
}
- }
- if `:length local cformat' {
= if 0 {
  confirm numeric format `cformat'
  if fmtwidth("`cformat'") > 9 {
    local cformat "%9.0g"
    di "{txt}note: invalid cformat(), using default"
  }
  sreturn local cformat "`cformat'"
  local cformat `cformat(`cformat')'
}
- if `:length local sformat' {

```

```

= if 0 {
    confirm numeric format `sformat'
    if fmtwidth(`""sformat'""') > 8 {
        local sformat "%8.2f"
        di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat `""sformat'""'
    local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`""pformat'""') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat `""pformat'""'
    local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, W0rd WIdth]
    exit 198
}
- opts_exclusive ""`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
    c_local fvwrapon fvwrapon(`fvwrapon')
}

> ----- end _get_diopts.Wrapon -----

- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DEFAULT
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options''
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

- if ``name2'' == "" {
= if `options' == "" {
    GetDiopts DUPS, `options'
}

```



```

- else {
- GetDiopts DUPS options, `options'
= GetDiopts DUPS options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``baselevels' `nobaselevels'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive ``emptycells' `noemptycells'
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab

```

```

- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"

```

```

= opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive ``abbrev' `noabbrev'
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `'' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'
    = local omitted
    - local lstretch `lstretch' `nolstretch'

```

```

= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {

```

```

    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat ``sformat''
    local sformat `sformat(`sformat')''
    }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth(`pformat') > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
        }
    sreturn local pformat ``pformat''
    local pformat `pformat(`pformat')''
    }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
    }
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> \_\_\_\_\_ begin \_get\_diopts.Wrapon \_\_\_\_\_

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
    }
- opts_exclusive ``word' `width' fvwrapon
= opts_exclusive " " fvwrapon

```

---

> \_\_\_\_\_ begin opts\_exclusive \_\_\_\_\_

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `""' != "" {
    local ++n

```



```

        gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----

        - local fvwrapon `word' `width'
        = local fvwrapon
        - if `:length local fvwrapon' {
        = if 0 {
            c_local fvwrapon fvwrapon(`fvwrapon')
        }
    }
}
> ----- end _get_diopts.Wrapon -----

        - if "`fvwrap'" != "" {
        = if "" != "" {
            FVWrap, `fvwrap'
        }
        - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

        = c_local DUPS
        - if `K' == 2 {
        = if 2 == 2 {
        - c_local `c_opts' ``options'''
        = c_local options `'''
        - }
        - sreturn local coding `coding'
        = sreturn local coding
        - sreturn local versus `versus'
        = sreturn local versus
        - sreturn local compare `compare'
        = sreturn local compare
        - sreturn local level `level'
        = sreturn local level 95
    }
}
> ----- end _get_diopts.GetDiopts -----

        - c_local `name2' ``options'''
        = c_local options `'''
        - }
        - local DUPS : list DUPS - DEFAULT
        - gettoken DUPS : DUPS
        - if ``DUPS''' != "" {

```

```

= if `""` != "" {
  if (strpos(`""DUPS""`, "(")) {
    gettoken DUPS : DUPS, parse("(")
    local DUPS `""DUPS'("`"
  }
  di as err "duplicate {bf:`DUPS'}` option not allowed"
  exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = `""S_`sname'""'
= sreturn local level = `95'
- }
- sreturn local `sname' = `""S_`sname'""'
= sreturn local method = `noadjust'
- }
}

> ----- end _get_diopts -----

- local legend coeflegend selegend
- local legend : list legend & diopts
- local diopts : list diopts - legend
- local lstretch lstretch nolstretch
- local lstretch : list lstretch & diopts
- local diopts : list diopts - lstretch
- local markdown markdown
- local markdown : list markdown & diopts
- local diopts : list diopts - markdown
- _get_diopts ignore, `diopts' `cformat' `sformat' `pformat'
= _get_diopts ignore,

> ----- begin _get_diopts -----

- version 11
- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]

```

```

= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "'allbaselevels' `noallbaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "'`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "'allbaselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "'`opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "'baselevels' `nobaselevels'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted` `noomitted`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch` `nolstretch`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive ``coding' `compare'
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive ``versus' `compare'
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
= local baselevels
    - local emptycells `emptycells' `noemptycells'
= local emptycells
    - local omitted `omitted' `noomitted'
= local omitted
    - local lstretch `lstretch' `nolstretch'
= local lstretch
    - local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
    - local abbrev `abbrev' `noabbrev'
= local abbrev
    - if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
        - if c(showbaselevels) == "all" {
            local allbaselevels allbaselevels
        }
        - else if c(showbaselevels) == "on" {
            local baselevels baselevels

```



```

    }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
  }

```

```

    }
    sreturn local pformat ``pformat'''
    local pformat `pformat(pformat)'''
  }
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd Width]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, WOrd Width]
  exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon
- if `:length local fvwrapon' {
= if 0 {
  c_local fvwrapon fvwrapon(`fvwrapon')

```

```

    }
}

> ----- end _get_diopts.Wrapon -----
- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local diopts
- if `K' == 2 {
= if 2 == 2 {
- c_local `c_opts' ``options''
= c_local options ``''
- }
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

> ----- end _get_diopts.GetDiopts -----
- c_local `name1' ``diopts''
= c_local ignore ``''
- local snames : s(macros)
- foreach sname of local snames {
- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_method = s(method)
- }
- sreturn clear
- GetDiopts DEFAULT

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(string)
> sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F

```

```

> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
    - syntax namelist(max=2) [, `DIOPTS' *]
    = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *)
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'` != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---


> 

---

 end opts_exclusive 

---


    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "


---


> 

---

 begin opts_exclusive 

---


    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ```opts'` != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit


---



```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`baselevels' `nobaselevels'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`emptycells' `noemptycells'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
    = opts_exclusive " "

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive ``coding' `compare'
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive ``versus' `compare'
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```



```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while `""opts""` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels

```

```

    }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'

```

```

        if fmtwidth("`pformat'") > 5 {
        local pformat "%5.3f"
        di "{txt}note: invalid pformat(), using default"
        }
        sreturn local pformat "`pformat'"
        local pformat `pformat(`pformat')'
    }
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
    }
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
- Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, WOrd WIdth]
- if c(rc) {
    di as err "invalid fvwrapon() option;"
    syntax [, WOrd WIdth]
    exit 198
    }
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
    - exit

> ----- end opts_exclusive -----
- local fvwrapon `word' `width'
= local fvwrapon

```

```

        - if `:length local fvwrapon' {
        = if 0 {
            c_local fvwrapon fvwrapon(`fvwrapon')
        }
    }

> ----- end _get_diopts.Wrapon -----

        - if "`fvwrap'" != "" {
        = if "" != "" {
            FVWrap, `fvwrap'
        }
        - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvrapon' `fvlabel' `noci' `nopvalues' `markdown' `citype' `abb
> rev'

        = c_local DEFAULT
        - if `K' == 2 {
        = if 1 == 2 {
            c_local `c_opts' ``options''
        }
        - sreturn local coding `coding'
        = sreturn local coding
        - sreturn local versus `versus'
        = sreturn local versus
        - sreturn local compare `compare'
        = sreturn local compare
        - sreturn local level `level'
        = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

        - if ``name2'' == "" {
        = if ``' == "" {
        - GetDiopts DUPS, `options'
        = GetDiopts DUPS,

> ----- begin _get_diopts.GetDiopts -----

        - local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV

        - syntax namelist(max=2) [, `DIOPTS' *]
        = syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND

```

```

> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *)
    - opts_exclusive "`allbaselevels' `noallbaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`allbaselevels' `nobaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    _____
> _____ end opts_exclusive _____
    - opts_exclusive "`baselevels' `nobaselevels'"
    = opts_exclusive " "
    _____
> _____ begin opts_exclusive _____
    - version 8.2

```

```

- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}

```

```

    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ``coeflegend' `selegend'
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive ``noci' `nopvalues'
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```



```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding` `compare`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus` `compare`"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""opts` != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {

```

```

- exit
> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
}
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
}
- }
- if "`emptycells'" == "" {

```

```

= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')"'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"
      di "{txt}note: invalid sformat(), using default"
    }
    sreturn local sformat "`sformat'"
    local sformat `sformat(`sformat')"'
  }
- if `:length local pformat' {
= if 0 {
    confirm numeric format `pformat'
    if fmtwidth("`pformat'") > 5 {
      local pformat "%5.3f"
      di "{txt}note: invalid pformat(), using default"
    }
    sreturn local pformat "`pformat'"
    local pformat `pformat(`pformat')"'
  }

```

```

    }
    - if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
    syntax namelist(max=2) [, `DIOPTS']
    }
    - if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
    local levelopt level(`level')
    }
    - Wrapon , `fvwrapon'
= Wrapon ,

> ----- begin _get_diopts.Wrapon -----
    - capture syntax [, W0rd WIdth]
    - if c(rc) {
        di as err "invalid fvwrapon() option;"
        syntax [, W0rd WIdth]
        exit 198
    }
    - opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
= local fvwrapon
    - if `:length local fvwrapon' {
= if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----

```

```

- if "`fvwrap'" != "" {
= if "" != "" {
    FVWrap, `fvwrap'
}
- c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

= c_local DUPS
- if `K' == 2 {
= if 1 == 2 {
    c_local `c_opts' ``options''
}
- sreturn local coding `coding'
= sreturn local coding
- sreturn local versus `versus'
= sreturn local versus
- sreturn local compare `compare'
= sreturn local compare
- sreturn local level `level'
= sreturn local level 95

```

---

```

> ----- end _get_diopts.GetDiopts -----
- }
- else {
    GetDiopts DUPS options, `options'
    c_local `name2' ``options''
}
- local DUPS : list DUPS - DEFAULT
- gettoken DUPS : DUPS
- if ``DUPS'' != "" {
= if ``' != "" {
    if (strpos(`DUPS', "(")) {
        gettoken DUPS : DUPS, parse("(")
        local DUPS `DUPS'()'
    }
    di as err "duplicate {bf:`DUPS'} option not allowed"
    exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = ``S_`sname''
= sreturn local level = `95'
- }
- sreturn local `sname' = ``S_`sname''

```

```

        = sreturn local method = `noadjust'
        - }
    _____
> _____ end _get_diopts _____
        - local cformat ``s(cformat)'''
        = local cformat `'''
        - local sformat ``s(sformat)'''
        = local sformat `'''
        - local pformat ``s(pformat)'''
        = local pformat `'''
        - _get_eformopts , eformopts(`options') allowed(__all__) soption
> s
        = _get_eformopts , eformopts() allowed(__all__) options
    _____
> _____ begin _get_eformopts _____
        - version 8.0
        - syntax [, eformopts(string asis) soptions ALLOWed(string) ]
        - local 0 , `eformopts'
        = local 0 ,
        - local EFALL NOHR hr NOSHR shr or IRr RRr tr TRatio
        - if ``soptions'" != "" {
        = if "soptions" != "" {
        - syntax [, EForm1(passthru) EForm `EFALL' * ]
        = syntax [, EForm1(passthru) EForm NOHR hr NOSHR shr or IRr RR
> r tr TRatio * ]
        - local opts ``options'''
        = local opts `'''
        - local 0 , `eform1' `eform' `nohr' `hr' `noshr' `shr' `or' `i
> rr' `rrr' `tr' `tratio'
        = local 0 ,
        - }
        - if ``allowed'" == "__all__" {
        = if "__all__" == "__all__" {
        - local allowed `EFALL'
        = local allowed NOHR hr NOSHR shr or IRr RRr tr TRatio
        - }
        - foreach ef of local allowed {
        - capture confirm name `ef'
        = capture confirm name NOHR
        - if _rc {
            di as err "`ef' is not a valid name"
            exit 198
        }
        - local efopts `efopts' `=lower("`ef'"')'
        = local efopts nohr
        - }

```

```

- capture confirm name `ef'
= capture confirm name hr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr
- }
- capture confirm name `ef'
= capture confirm name NOSHR
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr
- }
- capture confirm name `ef'
= capture confirm name shr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr
- }
- capture confirm name `ef'
= capture confirm name or
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or
- }
- capture confirm name `ef'
= capture confirm name IRr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr
- }
- capture confirm name `ef'

```

```

= capture confirm name RRr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr
- }
- capture confirm name `ef'
= capture confirm name tr
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr
- }
- capture confirm name `ef'
= capture confirm name TRatio
- if _rc {
  di as err "`ef' is not a valid name"
  exit 198
}
- local efopts `efopts' `=lower("`ef'")'
= local efopts nohr hr noshr shr or irr rrr tr tratio
- }
- local efopts : list uniq efopts
- syntax [, EForm1(string) EForm `allowed' ]
= syntax [, EForm1(string) EForm NOHR hr NOSHR shr or IRr RRr

> tr TRatio ]

- foreach ef of local efopts {
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }

```



```

- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- local eform `eform' ``ef''
= local eform
- }
- if ```eform1'``=="Odds ratio" {
= if ````=="Odds ratio" {
    local eform_cons_ti = "Odds"
    local eform_cons_note = "odds ratios"
}
- else if ```eform1'``=="Haz. ratio" {
= else if ````=="Haz. ratio" {
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
}
- else if ```eform1'``=="Time ratio" {
= else if ````=="Time ratio" {
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
}
- else if ```eform1'``=="IRR" {
= else if ````=="IRR" {
    local eform_cons_ti = "Inc. rate"
    local eform_cons_note = "incidence-rate ratios"
    local eform_cons_ti2 = "incidence rate"
}
- else if ```eform1'``=="Risk ratio" {
= else if ````=="Risk ratio" {
    local eform_cons_ti = "Risk"
    local eform_cons_note = "risk ratios"
}
- else if ```eform1'``=="Hlth ratio" {
= else if ````=="Hlth ratio" {
    local eform_cons_ti = "Health"
    local eform_cons_note = "health ratios"
}
- local k : list sizeof eform
- if `k' {
= if 0 {

```

```

opts_exclusive "`eform'"
if `:length local eform1' {
  opts_exclusive "eform()" `eform'"
}
}
- if `k' {
= if 0 {
  if ("`eform'"=="eform") local eform1 = "exp(b)"
  else if ("`eform'"=="hr") {
    local eform1 = "Haz. ratio"
    local eform_cons_ti = "Hazard"
    local eform_cons_note = "hazard ratios"
  }
  else if ("`eform'"=="shr") {
    local eform1 = "SHR"
    local eform_cons_ti = "Subhazard"
    local eform_cons_note = "subhazard ratios"
  }
  else if ("`eform'"=="tr") {
    local eform1 = "Time ratio"
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
  }
  else if ("`eform'"=="tratio") {
    local eform1 = "Time ratio"
    local eform_cons_ti = "Time"
    local eform_cons_note = "time ratios"
  }
  else if ("`eform'"=="or") {
    local eform1 = "Odds ratio"
    if ("`e(cmd)'"=="asclogit" | "`e(cmd)'"=="cmclogit" | "`e(cm
> d)'"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
      local eform_cons_ti = "Rel. risk"
      local eform_cons_note = "relative-risk ratios"
      local eform_cons_ti2 = "relative risk"
    }
    else {
      local eform_cons_ti = "Odds"
      local eform_cons_note = "odds ratios"
    }
  }
  else if ("`eform'"=="irr") {
    local eform1 = "IRR"
    local eform_cons_ti = "Inc. rate"
    local eform_cons_note = "incidence-rate ratios"
    local eform_cons_ti2 = "incidence rate"
  }
}

```

```

}
else if ("`eform`"=="rrr") {
  local eform1 = "RRR"
  local eform_cons_ti = "Rel. risk"
  local eform_cons_note = "relative-risk ratios"
  local eform_cons_ti2 = "relative risk"
}
else if ("`eform`"!="nohr") & ("`eform`"!="noshr") {
  local eform1 = upper("`eform`")
}
}

- sreturn clear
- sreturn local options ```opts`''
= sreturn local options ````
- sreturn local opt `eform'
= sreturn local opt
- sreturn local str `eform1'
= sreturn local str
- if ```eform1`'' != "" {
= if ```` != "" {
  sreturn local eform eform("`eform1`")
}
- else sreturn local eform ""
- sreturn local eform_cons_ti `eform_cons_ti'
= sreturn local eform_cons_ti
- sreturn local eform_cons_note `eform_cons_note'
= sreturn local eform_cons_note
- sreturn local eform_cons_ti2 `eform_cons_ti2'
= sreturn local eform_cons_ti2

```

---

```

> ----- end _get_eformopts -----
- local tr_item = ```s(eform_cons_note)`''
= local tr_item = ````
- if inlist("`bmatrix'", "", "e(b)") & "`e(k_eform)'" == "0" {
= if inlist("", "", "e(b)") & "" == "0" {
  local eform
}
- else local eform ```s(str)`''
= else local eform ````
- local eform_cons_ti ```s(eform_cons_ti)`''
= local eform_cons_ti ````
- local eform_cons_ti2 ```s(eform_cons_ti2)`''
= local eform_cons_ti2 ````
- if ("`e(consonly)'"!="1" | ```eform_cons_ti`''=="") {
= if ("0!="1" | ````=="") {
- local eformdi ```s(str)`''

```

```

= local eformdi `'''
- }
- else {
  local eformdi ``eform_cons_ti'''
  local tr_item = cond(missing("`eform_cons_ti2'"), strlower("`
> eform_cons_ti'''), ``eform_cons_ti2''')
  }
- local coefttl = cond("`eform''==`''', ``coeftitle''', ``ef
> ormdi''')
= local coefttl = cond(`''==`''', `''', `''')
- local options ``s(options)'''
= local options `'''
- _get_diparmopts, diparmopts(`options') level(`level')
= _get_diparmopts, diparmopts() level(95)

> ----- begin _get_diparmopts -----
- if _caller() < 8.2 {
  _get_diparmopts_8 `0'
  exit
  }
- version 9
- local version : di "version " string(_caller()) ":"
- syntax [, diparmopts(string asis) EXECute soptions bottom pl
> us SYNTAXonly * ]
- local goptions `options'
= local goptions level(95)
- SpliceGlobals junk , `goptions'
= SpliceGlobals junk , level(95)

> ----- begin _get_diparmopts.SpliceGlobals -----
- syntax name(id="macro name" name=c_diparm) [, diparm(strin
> g asis) Level(cilevel) dof(passthru) syntaxonly ]
- if ``dof''' == "" & ``e(df_r)''' != "" {
= if `'' == "" & `'' != "" {
  capture confirm integer number `e(df_r)'
  if !c(rc) {
    local gdof dof(`e(df_r)')
  }
  capture
  }
- else local gdof ``dof'''
= else local gdof `'''
- local glevel `level'
= local glevel 95
- local lopt level(`glevel')
= local lopt level(95)

```

```

- if (``diparm'' == "") exit
= if (``'' == "") exit

> ----- end _get_diparmopts.SpliceGlobals -----
- local 0 `", `diparmopts''
= local 0 `", ''
- local diparmopts
- local k 0
- syntax [, diparm(string asis) * ]
- while ``diparm'' != "" {
= while ``'' != "" {
    local ++k
    local single 1
    local after
    capture _on_colon_parse `diparm'
    if !c(rc) {
        local single 0
        local diparm ``s(before)''
        local after ``s(after)''
    }
    capture
    SpliceGlobals diparm , diparm(`diparm') `syntaxonly' `goptio

> ns'

    local diparm`k' ``diparm''
    local diparmopts ``diparmopts' : `diparm''
    if `single' | ``:list retok after'' == "" {
        local 0 `", `options''
        syntax [, diparm(string asis) * ]
    }
    else {
        local diparm ``after''
    }
}
- if ``soptions'' == "" {
= if ``'' == "" {
- syntax [, diparm(string asis) ]
- }
- if ("`execute" == "") {
= if (" == "") {
- sreturn clear
- sreturn local options ``options''
= sreturn local options ``''
- sreturn local diparm ``diparmopts''
= sreturn local diparm ``''
- forval i = 1/`k' {
= forval i = 1/0 {

```

```

        sreturn local diparm`i' ``diparm`i'`
    }
    - sreturn local k `k'
    = sreturn local k 0
    - exit

```

---

```

> ----- end _get_diparmopts -----
    - local NODIPARM : length local nodiparm
    - if `NODIPARM' {
    = if 0 {
        local options
    }
    - local GTOPTS parse(":") bind quotes
    - local k 0
    - if !`NODIPARM' {
    = if !0 {
        - if ``e(diparm)`` != "" {
    = if `"" != "" {
        local i 0
        local diparm ``e(diparm)``
    }
    - else if ``e(diparm1)`` != "" {
    = else if ``lnalpha, exp label("alpha")`` != "" {
        - local i 1
        - local diparm ``e(diparm1)``
    = local diparm ``lnalpha, exp label("alpha")``
        - }
    - while `:length local diparm' {
    = while 27 {
        - gettoken diparm rest : diparm, `GTOPTS'
    = gettoken diparm rest : diparm, parse(":") bind quotes
        - while `:length local rest' {
    = while 0 {
            local ++k
            local diparm`k' : copy local diparm
            gettoken COLON rest : rest, `GTOPTS'
            gettoken diparm rest : rest, `GTOPTS'
        }
        - local ++k
        - local diparm`k' : copy local diparm
    = local diparm1 : copy local diparm
        - local ++i
        - local diparm ``e(diparm`i')``
    = local diparm `""
        - }
    - while `:length local diparm' {

```

```

= while 0 {
  gettoken diparm rest : diparm, `GTPTS'
  while `:length local rest' {
    local ++k
    local diparm`k' : copy local diparm
    gettoken COLON rest : rest, `GTPTS'
    gettoken diparm rest : rest, `GTPTS'
  }
  local ++k
  local diparm`k' : copy local diparm
  local ++i
  local diparm ``e(diparm`i)'''
}
- }
- if `:length local options' {
= if 0 {
  local 0 `", `options'''
  syntax [, diparm(string asis) *]
  if `k' & `:length local diparm' {
    local ++k
    local diparm`k' __sep__
  }
  while `:length local diparm' {
    gettoken diparm rest : diparm, `GTPTS'
    while `:length local rest' {
      local ++k
      local diparm`k' : copy local diparm
      gettoken COLON rest : rest, `GTPTS'
      gettoken diparm rest : rest, `GTPTS'
    }
    local ++k
    local diparm`k' : copy local diparm
    local 0 `", `options'''
    syntax [, diparm(string asis) *]
  }
}
- if "`markdown'" != "" {
= if "" != "" {
  local lsize = c(linesize)
  set linesize 255
}
- `vv' mata: _coef_table()
= version 12, missing: mata: _coef_table()

```

---

```

> ----- begin _get_diopts -----
- version 11

```

```

- syntax namelist(max=2) [, *]
- gettoken name1 namelist : namelist
- gettoken name2 namelist : namelist
- GetDiopts diopts options, `options'
= GetDiopts diopts options,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```



```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

> ----- end opts_exclusive -----

    - local K : list sizeof namelist
    - gettoken c_diopts c_opts : namelist
    - opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
= local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}

```

```

> ----- end opts_exclusive -----
    - opts_exclusive "`versus' `compare'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - opts_exclusive "`abbrev' `noabbrev'"
    = opts_exclusive " "
    -----

> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
    = while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
    = if 0 < 2 {
    - exit
    -----

> ----- end opts_exclusive -----
    - local allbaselevels `allbaselevels' `noallbaselevels'
    = local allbaselevels
    - local baselevels `baselevels' `nobaselevels'
    = local baselevels
    - local emptycells `emptycells' `noemptycells'
    = local emptycells
    - local omitted `omitted' `noomitted'

```

```

= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {

```

```

confirm numeric format `sformat'
if fmtwidth("`sformat'") > 8 {
  local sformat "%8.2f"
  di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat "`sformat'"
local sformat `sformat(`sformat')"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth("`pformat'") > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat "`pformat'"
  local pformat `pformat(`pformat')"'
}
- if `K' == 1 & `:length local options' {
= if 2 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> begin \_get\_diopts.Wrapon

```

- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "`word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> begin opts\_exclusive

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

        = while `""` != "" {
            local ++n
            gettoken item`n` opts : opts, bind
        }
        - if `n` < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----

    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----

    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local diopts
    - if `K' == 2 {
    = if 2 == 2 {
        - c_local `c_opts' ``options'
    = c_local options `""'
        - }
        - sreturn local coding `coding'
    = sreturn local coding
        - sreturn local versus `versus'
    = sreturn local versus
        - sreturn local compare `compare'
    = sreturn local compare
        - sreturn local level `level'
    = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----

    - c_local `name1' ``diopts'
    = c_local __000007 `""'
    - local snames : s(macros)
    - foreach sname of local snames {

```



```

- local S_`sname' = s(`sname')
= local S_level = s(level)
- }
- local S_`sname' = s(`sname')
= local S_k = s(k)
- }
- sreturn clear
- GetDiopts DEFAULT

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND cformat(strin
> g) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFlegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
local ++n
gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----

```

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels' `nobaselevels'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells' `noemptycells'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n

```

```

        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
    - exit
}
> ----- end opts_exclusive -----
    - opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "
}
> ----- begin opts_exclusive -----
    - version 8.2
    - args opts optname errcode optslab
    - local opts `opts'
    = local opts
    - local n 0
    - while ``opts' != "" {
= while `"' != "" {
        local ++n
        gettoken item`n' opts : opts, bind
    }
    - if `n' < 2 {
= if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - opts_exclusive "`lstretch' `nolstretch'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0

```

```

- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`noci' `nopvalues'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels

```

```

- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch
- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'

```

```

}
- if `:length local sformat' {
= if 0 {
  confirm numeric format `sformat'
  if fmtwidth(`"'sformat'"') > 8 {
    local sformat "%8.2f"
    di "{txt}note: invalid sformat(), using default"
  }
  sreturn local sformat `"'sformat'"'
  local sformat `"'sformat(`sformat')'"'
}
- if `:length local pformat' {
= if 0 {
  confirm numeric format `pformat'
  if fmtwidth(`"'pformat'"') > 5 {
    local pformat "%5.3f"
    di "{txt}note: invalid pformat(), using default"
  }
  sreturn local pformat `"'pformat'"'
  local pformat `"'pformat(`pformat')'"'
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
  syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
  local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

```

> ----- begin _get_diopts.Wrapon -----
- capture syntax [, W0rd WIdth]
- if c(rc) {
  di as err "invalid fvwrapon() option;"
  syntax [, W0rd WIdth]
  exit 198
}
- opts_exclusive "'word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```



```

        = local opts
        - local n 0
        - while ``opts'' != "" {
        = while `'' != "" {
            local ++n
            gettoken item`n' opts : opts, bind
        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }

> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }

> ----- end _get_diopts.Wrapon -----
    - if ``fvwrap'' != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'

    = c_local DEFAULT
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options''
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95

> ----- end _get_diopts.GetDiopts -----
    - if ``name2'' == "" {
    = if `'' == "" {

```

```

- GetDiopts DUPS, `options'
= GetDiopts DUPS,

```

---

```

> ----- begin _get_diopts.GetDiopts -----
- local DIOPTS Level(cilevel) vsquish NOALLBASElevels ALLBAS
> Elevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMPTYcells EMPTY
> cells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND cformat(stri
> ng) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRAP(passthru) F
> VWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYPE(passthru) N
> OABBREV ABBREV
- syntax namelist(max=2) [, `DIOPTS' *]
= syntax namelist(max=2) [, Level(cilevel) vsquish NOALLBASE
> levels ALLBASElevels NOBASElevels BASElevels noCNSReport FULLCNSReport NOEMP
> TYcells EMPTYcells NOOMITted OMITted NOLSTRETCH LSTRETCH COEFLegend SELEGEND
> cformat(string) sformat(string) pformat(string) CODING VERSUS COMPARE FVWRA
> P(passthru) FVWRAPON(string) NOFVLABEL FVLABEL NOCI NOPValues MARKDOWN CITYP
> E(passthru) NOABBREV ABBREV *]
- opts_exclusive "`allbaselevels' `noallbaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

```

---

```

> ----- end opts_exclusive -----
- opts_exclusive "`allbaselevels' `nobaselevels'"
= opts_exclusive " "

```

---

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {

```

```

= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`baselevels` `nobaselevels`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`emptycells` `noemptycells`"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts`
= local opts
- local n 0
- while `""`opts`"" != "" {
= while `""` != "" {
    local ++n
    gettoken item`n` opts : opts, bind
}
- if `n` < 2 {
= if 0 < 2 {
- exit
}

```

```

> ----- end opts_exclusive -----
- opts_exclusive "`fvlabel' `nofvlabel'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`omitted' `noomitted'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`lstretch' `nolstretch'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'

```

```

= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`cnsreport' `fullcnsreport'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----
- opts_exclusive "`coeflegend' `selegend'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts'' != "" {
= while `'' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {

```

```

= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`nocl' `nopvalues'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- local K : list sizeof namelist
- gettoken c_diopts c_opts : namelist
- opts_exclusive "`coding' `compare'"
= opts_exclusive " "

> ----- begin opts_exclusive -----

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while "`opts'" != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit
}

> ----- end opts_exclusive -----

- opts_exclusive "`versus' `compare'"
= opts_exclusive " "

```

```

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- opts_exclusive "`abbrev' `noabbrev'"
= opts_exclusive " "

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local allbaselevels `allbaselevels' `noallbaselevels'
= local allbaselevels
- local baselevels `baselevels' `nobaselevels'
= local baselevels
- local emptycells `emptycells' `noemptycells'
= local emptycells
- local omitted `omitted' `noomitted'
= local omitted
- local lstretch `lstretch' `nolstretch'
= local lstretch

```

```

- local fvlabel `fvlabel' `nofvlabel'
= local fvlabel
- local abbrev `abbrev' `noabbrev'
= local abbrev
- if "`allbaselevels'`baselevels'" == "" {
= if "" == "" {
- if c(showbaselevels) == "all" {
    local allbaselevels allbaselevels
  }
- else if c(showbaselevels) == "on" {
    local baselevels baselevels
  }
- }
- if "`emptycells'" == "" {
= if "" == "" {
- if c(showemptycells) == "off" {
    local emptycells noemptycells
  }
- }
- if "`omitted'" == "" {
= if "" == "" {
- if c(showomitted) == "off" {
    local omitted noomitted
  }
- }
- if "`lstretch'" == "" {
= if "" == "" {
- if c(lstretch) == "off" {
    local lstretch nolstretch
  }
- }
- if `:length local cformat' {
= if 0 {
    confirm numeric format `cformat'
    if fmtwidth("`cformat'") > 9 {
      local cformat "%9.0g"
      di "{txt}note: invalid cformat(), using default"
    }
    sreturn local cformat "`cformat'"
    local cformat `cformat(`cformat')'
  }
- if `:length local sformat' {
= if 0 {
    confirm numeric format `sformat'
    if fmtwidth("`sformat'") > 8 {
      local sformat "%8.2f"

```



```

di "{txt}note: invalid sformat(), using default"
}
sreturn local sformat ``sformat'''
local sformat `sformat(`sformat')'''
}
- if `:length local pformat' {
= if 0 {
confirm numeric format `pformat'
if fmtwidth(`pformat') > 5 {
local pformat "%5.3f"
di "{txt}note: invalid pformat(), using default"
}
sreturn local pformat ``pformat'''
local pformat `pformat(`pformat')'''
}
- if `K' == 1 & `:length local options' {
= if 1 == 1 & 0 {
syntax namelist(max=2) [, `DIOPTS']
}
- if reldif(`level', c(level)) > 1e-3 {
= if reldif(95, c(level)) > 1e-3 {
local levelopt level(`level')
}
- Wrapon , `fvwrapon'
= Wrapon ,

```

---

> 

---

 begin \_get\_diopts.Wrapon 

---

```

- capture syntax [, WOrd WIdth]
- if c(rc) {
di as err "invalid fvwrapon() option;"
syntax [, WOrd WIdth]
exit 198
}
- opts_exclusive ``word' `width'" fvwrapon
= opts_exclusive " " fvwrapon

```

---

> 

---

 begin opts\_exclusive 

---

```

- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts''' != "" {
= while `''' != "" {
local ++n
gettoken item`n' opts : opts, bind

```

```

        }
        - if `n' < 2 {
        = if 0 < 2 {
        - exit
    }
}
> ----- end opts_exclusive -----
    - local fvwrapon `word' `width'
    = local fvwrapon
    - if `:length local fvwrapon' {
    = if 0 {
        c_local fvwrapon fvwrapon(`fvwrapon')
    }
}
> ----- end _get_diopts.Wrapon -----
    - if "`fvwrap'" != "" {
    = if "" != "" {
        FVWrap, `fvwrap'
    }
    - c_local `c_diopts' `levelopt' `vsquish' `allbaselevels' `b
> aselevels' `cnsreport' `fullcnsreport' `emptycells' `omitted' `lstretch' `co
> eflegend' `selegend' `cformat' `sformat' `pformat' `coding' `versus' `compar
> e' `fvwrap' `fvwrapon' `fvlabel' `nocl' `nopvalues' `markdown' `citype' `abb
> rev'
    = c_local DUPS
    - if `K' == 2 {
    = if 1 == 2 {
        c_local `c_opts' ``options'
    }
    - sreturn local coding `coding'
    = sreturn local coding
    - sreturn local versus `versus'
    = sreturn local versus
    - sreturn local compare `compare'
    = sreturn local compare
    - sreturn local level `level'
    = sreturn local level 95
}
> ----- end _get_diopts.GetDiopts -----
    - }
    - else {
        GetDiopts DUPS options, `options'
        c_local `name2' ``options'
    }
    - local DUPS : list DUPS - DEFAULT
    - gettoken DUPS : DUPS
    - if ``DUPS' != "" {

```

```

= if `""' != "" {
    if (strpos(`""DUPS'""', "(")) {
        gettoken DUPS : DUPS, parse("(")
        local DUPS `""DUPS'(')'"
    }
    di as err "duplicate {bf:`DUPS'} option not allowed"
    exit 198
}
- sreturn clear
- foreach sname of local snames {
- sreturn local `sname' = `""S_`sname'""'
= sreturn local level = `95'
- }
- sreturn local `sname' = `""S_`sname'""'
= sreturn local k = `0'
- }

> _____ end _get_diopts _____
> _____ begin _diparm _____
- version 11
- if _caller() < 8.2 {
    _diparm_8 `0'
    exit
}
- syntax anything(name=eqs) [, Level(cilevel) NOTAB DOF(passth
> ru) BMATrix(passthru) VMATrix(passthru) DFMATrix(passthru) COEFlegend SELEGE
> ND * ]
- tempname T
- if `:length local notab' {
= if 5 {
- `T' = ._b_diparm.new, level(`level') `dof' `bmatrix' `vmatr
> ix' `dfmatrix' `coeflegend' `selegend'
= .__000008 = ._b_diparm.new, level(95)
-class _b_diparm {
-     string  LEVEL    = ""
-     double  IS_MI    = 0
-     string  BMAT      = ""
-     string  COLNA     = ""
-     string  VMAT      = ""
-     string  DFMAT     = ""
-     string  COLFUL    = ""
-     double  DF        = 0
-     double  DF_HOLD   = 0
-     double  EXTRA     = 1
-     string  METHOD     = ""

```

```

-      double  has_b   = 0
-      double  has_v   = 0
-      double  has_df  = 0
-      double  has_Cns = 0
-}

```

---

```

> ----- begin _b_diparm.new -----
- version 11
- .IS_MI = ``e(mi)'' == "mi" & ``e(df_mi)'' == "matrix"
= .IS_MI = ``'' == "mi" & ``'' == "matrix"
- if `.IS_MI' {
= if 0 {
  local ALTOPT DFMATrix(string)
}
- syntax [, Level(cilevel) DOF(real -1) BMATrix(string) VMAT
> rix(string) EXTRA(real 1) MCOMPARE(passthru) `ALTOPT' ]
  = syntax [, Level(cilevel) DOF(real -1) BMATrix(string) VMAT
> rix(string) EXTRA(real 1) MCOMPARE(passthru) ]
- _get_mcompare, `mcompare'
= _get_mcompare,

```

---

```

> ----- begin _get_mcompare -----
- version 11
- local cmd ``e(cmd)''
= local cmd `nbreg'
- syntax [name(name=caller)] [, *]
- local is_pwc = inlist("pwcompare", "`cmd'", "`caller'")
> | inlist("pwmean", "`cmd'", "`caller'")
  = local is_pwc = inlist("pwcompare", "nbreg", "") | inlist
> ("pwmean", "nbreg", "")
- local is_marg = inlist("margins", "`cmd'", "e(cmd2)",
> "`caller'")
  = local is_marg = inlist("margins", "nbreg", "", "")
- local OPTIONS NOADJust BONferroni DUNCAn DUNNett SCHeffe
> SIDak SNK TUKey ADJustall
- if `is_pwc' {
= if 0 {
  syntax [name] [, `OPTIONS' MCOMPare(string) *]
}
- else {
- syntax [name] [, MCOMPare(string) *]
- local suboption mcompare()
- }
- sreturn clear
- sreturn local options ``options''
= sreturn local options ``''

```

```

- if `is_pwc' {
= if 0 {
    local METHOD `noadjust' `bonferroni' `duncan' `dunnett'
> `scheffe' `sidak' `snk' `tukey'
    local METHOD : list uniq METHOD
    opts_exclusive "`METHOD'"
    local ALL `adjustall'
    opts_exclusive "`noadjust' `ALL'"
    }
- local 0 `", `mcompare'"
= local 0 `", ""
- capture syntax [, `OPTIONS']
= capture syntax [, NOADJUST BONferroni DUNCAn DUNNett SCH
> effe SIDak SNK TUKey ADJustall]
- if c(rc) {
    di as err "invalid mcompare() option;"
    di as err "method ``mcompare'' not allowed"
    error 198
    }
- local METHOD `METHOD' `noadjust' `bonferroni' `duncan' `
> dunnett' `scheffe' `sidak' `snk' `tukey'
= local METHOD
- local METHOD : list uniq METHOD
- local ALL `ALL' `adjustall'
= local ALL
- local ALL : list uniq ALL
- opts_exclusive "`METHOD'" `suboption'
= opts_exclusive "" mcompare()

> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ```opts'"" != "" {
= while `""' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- if "`METHOD'" == "" {

```

```

= if "" == "" {
- if "`ALL'" != "" {
= if "" != "" {
  di as err "multiple comparison method required with the
> adjustall option"
  exit 198
}
- local METHOD noadjust
- local noadjust noadjust
- }
- opts_exclusive "`ALL' `noadjust'" `suboption'
= opts_exclusive " noadjust" mcompare()


---


> 

---

 begin opts_exclusive 

---


- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts noadjust
- local n 0
- while ``opts'" != "" {
= while `"noadjust'" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts'" != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 1 < 2 {
- exit


---


> 

---

 end opts_exclusive 

---


- local EXCLUDE duncan dunnett snk tukey
- if `:list METHOD in EXCLUDE' {
= if 0 {
  if `is_pwc' {
    local opt : copy local METHOD
  }
  else {
    local opt "mcompare(`METHOD')"
  }
  local cmdlist anova manova regress mvreg pwmean
  if "`cmd'" == "pwcompare" {

```

```

        local cmd ``e(cmd_est)'''
    }
    if `is_marg' {
        local cmd margins
    }
    if !`:list cmd in cmdlist' | !`is_pwc' {
        di as err `"method `opt' is not allowed with results fro
> m `cmd'''
        exit 322
    }
    if ``e(prefix)''' == "svy" {
        di as err `"method `opt' is not allowed with results usi
> ng the svy prefix"
        exit 322
    }
    local vce ``e(vce)'''
    if !inlist("`vce'", "", "ols") {
        di as err `"method `opt' is not allowed with results usi
> ng vce(`vce')"
        exit 322
    }
    }
    - local EXCLUDE duncan dunnett snk tukey scheffe
    - if `:list METHOD in EXCLUDE' {
    = if 0 {
        if "`ALL'" != "" {
            di as err `"option adjustall is not allowed with method
> `METHOD'''
            exit 198
        }
    }
    - sreturn local method ``METHOD'''
    = sreturn local method `noadjust"
    - sreturn local adjustall ``ALL'''
    = sreturn local adjustall `""'

> _____ end _get_mcompare _____
    - if !inlist("`s(method)", "noadjust", "bonferroni", "sidak
> ") {
    = if !inlist("noadjust", "noadjust", "bonferroni", "sidak")
> {
        di as err "option `mcompare' not allowed"
        exit 198
    }
    - .METHOD = "`s(method)'"
    = .METHOD = "noadjust"

```

```

- .LEVEL = "`level'"
= .LEVEL = "95"
- if "`bmatrix'" == "" {
= if "" == "" {
- if `:length local vmatrix' {
= if 0 {
    di as err "option vmatrix() requires option bmatrix()"
    exit 198
  }
- }
- if "`vmatrix'" == "" {
= if "" == "" {
- if `:length local dfmatrix' {
= if 0 {
    di as err "option dfmatrix() requires option vmatrix()"
    exit 198
  }
- }
- if `'.IS_MI' & "`bmatrix'`vmatrix'`dfmatrix'`e(properties)'
> "==" {
= if 0 & "b V"==" {
    local bmatrix e(b_mi)
    local vmatrix e(V_mi)
    local dfmatrix e(df_mi)
  }
- if `:length local bmatrix' {
= if 0 {
    .has_b = 1
    .BMAT = "`bmatrix'"
    .COLNA = "`:colful `bmatrix'"
  }
- else {
- .COLNA = "`:colful e(b)'"
= .COLNA = "y_timing_max:cb_authors_ext y_timing_max:estimat
> ed y_timing_max:wg_ndx_mult y_timing_max:stky_pr_rotemberg y_timing_max:wlth
> y_timing_max:stky_pr_calvo y_timing_max:ln_neq y_timing_max:open y_timing_m
> ax:_cons /lnalpha"
- .has_Cns = "`e(Cns)'" == "matrix"
= .has_Cns = "" == "matrix"
- }
- if `:length local vmatrix' {
= if 0 {
    .has_v = 1
    .VMAT = "`vmatrix'"
  }
- if `:length local dfmatrix' {

```



```

= if 0 {
    .has_df = 1
    .DFMAT = "`dfmatrix'"
}
- if `dof' <= 0 {
= if -1 <= 0 {
- local df_r 0
- Chk4PosInt df_r
}

> ----- begin _b_diparm.Chk4PosInt -----
- version 11
- args ename
- if ``"e(`ename')"' != "" {
= if `""' != "" {
    capture confirm integer number `e(`ename')'
    if !c(rc) {
        if `e(`ename')' > 0 {
            c_local `ename' `e(`ename')'
        }
    }
}

> ----- end _b_diparm.Chk4PosInt -----
- if `df_r' {
= if 0 {
    .DF = `df_r'
}
- }
- else {
    .DF = `dof'
}
- if `extra' > 1 {
= if 1 > 1 {
    .EXTRA = `extra'
}

> ----- end _b_diparm.new -----
- .`T'.compute `eqs', `options'
= .__000008.compute lnalpha, exp

> ----- begin _b_diparm.compute -----
- version 11
- version 11
- syntax anything(name=EQS id="equation name(s)") [, NOProb
> Prob EXP TANH ILOGIT INVLOGIT Function(str) Derivative(str) CI(string) ]
- opts_exclusive "`prob' `noprob'"

```

```

= opts_exclusive " "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts
- local n 0
- while ``opts' != "" {
= while `"' != "" {
    local ++n
    gettoken item`n' opts : opts, bind
    }
- if `n' < 2 {
= if 0 < 2 {
- exit

> ----- end opts_exclusive -----
- local prob `prob' `noprob'
= local prob
- if ``ilogit'`invlogit' != "" {
= if "" != "" {
    local ilogit ilogit
    }
- local builtin `exp' `tanh' `ilogit'
= local builtin exp
- local fd 0
- if `:length local function' {
= if 0 {
    local ++fd
    }
- if `:length local derivative' {
= if 0 {
    local ++fd
    }
- if `fd' == 1 {
= if 0 == 1 {
    di as err "both function() and derivative() must be specif
> ied"

    exit 198
    }
- if `fd' {
= if 0 {
    local fopt "function()"
    }
- opts_exclusive "`builtin' `fopt'"

```

```

= opts_exclusive "exp "
> ----- begin opts_exclusive -----
- version 8.2
- args opts optname errcode optslab
- local opts `opts'
= local opts exp
- local n 0
- while ``opts' != "" {
= while "exp" != "" {
- local ++n
- gettoken item`n' opts : opts, bind
= gettoken item1 opts : opts, bind
- }
- while ``opts' != "" {
= while `"' != "" {
  local ++n
  gettoken item`n' opts : opts, bind
}
- if `n' < 2 {
= if 1 < 2 {
- exit
> ----- end opts_exclusive -----
- if `.has_b' {
= if 0 {
  local bmat ``.BMAT'
}
- else {
- local bmat `e(b)'
- }
- foreach EQ of local EQS {
- if substr(`EQ',1,1) != "/" {
= if substr(`lnalpha',1,1) != "/" {
- local eq `"/EQ'
= local eq `"/lnalpha'
- }
- else local eq : copy local EQ
- local el : colnumb `bmat' `eq'
= local el : colnumb e(b) "/lnalpha"
- if inlist(`el', "", ".") {
= if inlist(`10', "", ".") {
  di as err "parameter named {bf:`EQ'} not found"
  exit 322
}
- local eqs `eqs' `eq'

```

```

= local eqs /lnalpha
- }
- local neq : list sizeof eqs
- if `.has_Cns' {
= if 0 {
    tempname Cns C
    matrix `Cns' = e(Cns)
    }
- if `neq' != 1 {
= if 1 != 1 {
    local el 0
    }
- if `.IS_MI' {
= if 0 {
    if `.has_df' {
        local df = el(`.DFMAT',1,`el')
    }
    else {
        local df = el(e(df_mi),1,`el')
    }
    if missing(`df') {
        local df 0
    }
}
- else {
- local df = `.DF'
= local df = 0
- }
- tempname z alpha
- scalar `alpha' = (100 - `.LEVEL')/100
= scalar __00000A = (100 - 95)/100
- mata: st_numscalar("`z'", _mc_crit("`METHOD'", `df' == 0
> ? . : `df', st_numscalar("`alpha'"), ., `.EXTRA'))
= mata: st_numscalar("__000009", _mc_crit("noadjust", 0 == 0
> ? . : 0, st_numscalar("__00000A"), ., 1))
- local isCns 0
- if `neq' == 1 {
= if 1 == 1 {
- tempname b se
- if `.has_b' {
= if 0 {
    scalar `b' = el(`.BMAT',1,`el')
    if `.has_v' {
        scalar `se' = sqrt(el(`.VMAT',`el',`el'))
    }
    else {

```

```

    scalar `se' = 0
  }
}
- else {
- scalar `b' = _b[`eqs']
= scalar __00000B = _b[/lnalpha]
- if "`e(V)'" == "matrix" {
= if "matrix" == "matrix" {
- scalar `se' = _se[`eqs']
= scalar __00000C = _se[/lnalpha]
- }
- else {
  scalar `se' = 0
  }
- if `'.has_Cns' & `se' == 0 {
= if 0 & __00000C == 0 {
  matrix `C' = `Cns'[1...,`el']
  matrix `C' = `C'*`C'
  local isCns = `C'[1,1] != 0
  }
- }
- if `se' == 0 {
= if __00000C == 0 {
  scalar `se' = .
  }
- if "`function'" != "" {
= if "" != "" {
  .Function `b' `se' `z' "`function'" "`derivative'"
  }
- else if "`builtin'" != "" {
= else if "exp" != "" {
- .`builtin' `b' `se' `z'
= .exp __00000B __00000C __000009

```

---

> begin \_b\_diparm.exp

```

- version 11
- args b se z
- return scalar est = exp(`b')
= return scalar est = exp(__00000B)
- return scalar se = return(est)*`se'
= return scalar se = return(est)*__00000C
- return scalar lb = exp(`b'-`z'*`se')
= return scalar lb = exp(__00000B-__000009*__00000C)
- return scalar ub = exp(`b'+`z'*`se')
= return scalar ub = exp(__00000B+__000009*__00000C)
- return scalar i = 0

```

```

> ----- end _b_diparm.exp -----
- }
- else {
  if ("`prob'" != "noprob") local prob prob
  .identity `b' `se' `z' `el' "`ci'"
}
- }
- else {
  .Multiple "`eqs'" `z' "`function'" "`derivative'" "`ci'"
  local isCns = r(cns)
}
- .AddTest `df', `prob'
= .AddTest 0,

> ----- begin _b_diparm.AddTest -----
- version 11
- syntax anything(name=df) [, Prob]
- return add
- if "`prob'" == "" {
= if "" == "" {
- return scalar p = .
- return scalar z = .
- exit

> ----- end _b_diparm.AddTest -----
- return add
- if `df' == 0 {
= if 0 == 0 {
- return scalar df = .
- }
- else return scalar df = `df'
= else return scalar df = 0
- return scalar crit = `z'
= return scalar crit = __000009
- return scalar cns = `isCns'
= return scalar cns = 0

> ----- end _b_diparm.compute -----
- exit

> ----- end _diparm -----

> -----
y_timing_max | Robust
               | Coefficient std. err.      z      P>|z|      [95% conf. inte

```

```

> rval]

```

> _____							
cb_authors_ext	.3359876	.0889968	3.78	0.000	.1615572	.51	
> 04181							
estimated	.3925615	.091225	4.30	0.000	.2137637	.57	
> 13592							
wg_ndx_mult	.614663	.2465807	2.49	0.013	.1313738	1.0	
> 97952							
stky_pr_rotemberg	-1.252631	.1472768	-8.51	0.000	-1.541288	-.96	
> 39736							
wlth	-.3508718	.109559	-3.20	0.001	-.5656035	-.13	
> 61401							
stky_pr_calvo	-.7503445	.1466333	-5.12	0.000	-1.03774	-.46	
> 29486							
ln_neq	.2013941	.0655946	3.07	0.002	.0728311	.32	
> 99572							
open	-.4139468	.1435197	-2.88	0.004	-.6952402	-.13	
> 26534							
_cons	.4444038	.1912598	2.32	0.020	.0695416	.8	
> 19266							
> _____							
/lnalpha	-2.723656	.8525592			-4.394642	-1.0	
> 52671							
> _____							
alpha	.0656343	.0559572			.0123433	.34	
> 90043							
> _____							

```

- if "`markdown'" != "" {
= if "" != "" {
    set linesize `lsize'
}
- if "`eqselect'" == "" & "`eformnote'" != "noeformnote" {
= if "" == "" & "" != "noeformnote" {
- if ``"eformdi"'"!="" & ("`e(cmd)'"!="gnbreg" | "`e(prefix)'"==
> "svy") {
= if ``"'!="" & ("nbreg"!="gnbreg" | ""=="svy") {
    local k_eform `r(k_eform)'
    if ("`k_eform'"=="") {
        local k_eform = 1
    }
    local k_eq `e(k_eq)'
    if ("`k_eq'"=="") {

```

```

        local k_eq = 1
    }
    if (`k_eform' < `k_eq' & `k_eform') {
        if ("`e(cmd2)'" == "stintreg" | "`e(cmd2)'" == "streg") & ("`e
> (cmd)'" == "weibull" | "`e(cmd)'" == "gompertz") {
        }
        else {
            _eform_multeq_note efnote : `k_eform' ``tr_item'''
            di as txt "`efnote'"
        }
    }
    - if ("`eformdi'" != "" & "`e(noconstant)'" == "0" & "`eform_cons
> _ti'" != "" & "`e(consonly)'" != "1") {
        = if ("'" != "" & "0" == "0" & "'" != "" & "0" != "1") {
            if ("`e(cmd)'" == "asclogit" | "`e(cmd)'" == "cmclogit" | "`e(cmd)
> '" == "cmmixlogit" | "`e(cmd)'" == "cmxtmixlogit" ) {
                local asvar = "`e(ranvars)'"`e(fixvars)'"`e(indvars)'" != ""
                if ("`e(casevars)'" != "" & `asvar' {
                    local note1 "Exponentiated coefficients represent odds ratios
> for "
                    local note2 "alternative-specific variables (first equation) a
> nd "
                    local note3 "{help j_cmrisk##|_new:relative-risk ratios} "
                    local note4 "for case-specific variables."
                    di as txt "{p 0 6 2}Note: `note1'`note2'`note3'`note4'`p_end'"
                }
            }
            if "`eform_cons_ti'" == "Inc. rate" {
                local eform_cons_note "incidence rate"
            }
            else if "`eform_cons_ti'" == "Rel. risk" {
                local eform_cons_note "relative risk"
            }
            else if "`eform_cons_ti'" == "Health" {
                local eform_cons_note "health (probability of no disease)"
            }
            else {
                local eform_cons_note = strlower("`eform_cons_ti'")
            }
            local eform_cons_note "baseline `eform_cons_note'"
            if ("`e(cmd2)'" == "") {
                local cmd `e(cmd)'
            }
            else {
                local cmd `e(cmd2)'
            }

```



```

    }
    local f2 = udsustr("`cmd'",1,2)
    local is_re = ("`f2'"=="xt" | "`f2'"=="me")
    if `is_re' {
        local extranote " (conditional on zero random effects)"
    }
    di as txt "{p 0 6 2}Note: {bf:_cons} estimates " "`eform_cons_
> note'`extranote'" _c
    if ("`e(cmd)'"=="mlogit" | "`e(cmd)'"=="asclogit" | "`e(cmd)'"
> ==="cmclogit" | "`e(cmd)'"=="cmmixlogit" | "`e(cmd)'"=="cmxtmixlogit" ) {
        di as txt " for each outcome" _c
    }
    di as txt ".{p_end}"
    }
- }
- return add
- if !`keepmc' {
= if !0 {
- return local mcmethod
- return local mctitle
- }

_____ end _coef_table _____
>
- if "`footnote'" == "" {
= if "nofootnote" == "" {
    ml_footnote
}

_____ end mopt.Display _____
>
- exit

_____ end mopt _____
>
- exit

_____ end ml _____
>
- if "`e(chi2_ct)'"!="LR" { exit }
= if ""!="LR" { exit }

_____ end nbreg.Display _____
>
- }
- error `e(rc)'
= error 0

_____ end nbreg.Estimate _____
>
- version 10: ereturn local cmdline ``nbreg `0''
= version 10: ereturn local cmdline ``nbreg y_timing_max cb_authors_ext es

```

```
> timed wg_ndx_mult stky_pr_rotemberg wlth stky_pr_calvo ln_neq open if y_  
> timing_max<60 , vce(r)'''  
  
----- end nbreg -----  
>  
- log close  
  name: <unnamed>  
  log: /Users/connorbrennan/Library/CloudStorage/OneDrive-TheUniversityo  
> fChicago/mmb/output/stepwise_regressions/y_timing_max_all_det.smcl  
  log type: smcl  
closed on: 8 May 2025, 20:04:57  
  
-----
```