frbussupport.Rnw: Create Support Files

Gary Young

July 2, 2016

0.1 stdver_varinfo

This code creates the varinfo.csv support file, parsing the data from the fixed format text file "frbus_package/mods/stdver_varinfo" into a data.frame.

```
raw = readLines("frbus_package/mods/stdver_varinfo")
```

Here we define the fixed length fields. The only reference to the original file seems to be in the stochsim program, to determine which are stochastic equations. There seems to be information in this file that is not used anywhere. I've made up names for some fields to use until I find better information.

```
flds = c("seq","vname","vdesc","vtype","vrule","sector",
         "var7","stoch","var8","var9","decomp")
start = c(1,5,16,111,115,117,130,132,135,137,139)
length = c(3,8,94,4,1,13,1,2,1,1,2)
parse = data.frame(flds,start,length)
rownames(parse) = parse$flds
(parse = subset(parse,select = -c(flds)))

##        start length
## seq        1      3
## vname      5      8
## vdesc     16     94
## vtype    111      4
## vrule    115      1
## sector   117     13
## var7     130      1
## stoch    132      2
## var8     135      1
## var9     137      1
## decomp   139      2
```

Using the field definitions that I created above, I transform the file's information into a data.frame.

```
varinfo = data.frame(lapply(flds,
            function(x) (trimws(substr(raw,parse[x,"start"],
                                sum(c(parse[x,"start"],parse[x,"length"],-1)))))))
colnames(varinfo) = flds
varinfo = varinfo[varinfo$vname!="ZZZBLANK",]
rownames(varinfo) = as.character(varinfo[,"vname"])
varinfo$seq = as.numeric(as.character(varinfo$seq))
varinfo$var7 = as.numeric(as.character(varinfo$var7))
varinfo$var8 = as.numeric(as.character(varinfo$var8))
varinfo$var9 = as.numeric(as.character(varinfo$var9))
```

```
varinfo$decomp = as.numeric(as.character(varinfo$decomp))
str(varinfo)
```

```
## 'data.frame': 508 obs. of  11 variables:
##  $ seq   : num  1 2 3 4 5 6 7 8 9 10 ...
##  $ vname : Factor w/ 509 levels "CENG","D01Q4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ vdesc : Factor w/ 507 levels "10-year expected PCE price inflation (Survey of Professional
##  $ vtype : Factor w/ 16 levels "","B","B.1","B.2",..: 6 16 16 16 12 12 12 12 12 12 ...
##  $ vrule : Factor w/ 2 levels "","A": 2 1 1 1 1 1 1 1 1 1 ...
##  $ sector: Factor w/ 387 levels "","sector_a.1",..: 69 1 1 1 1 1 1 1 1 1 ...
##  $ var7  : num  4 1 1 1 1 1 1 1 1 1 ...
##  $ stoch : Factor w/ 11 levels "","FN","GV","IN",..: 8 7 7 7 7 7 7 7 7 7 ...
##  $ var8  : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ var9  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ decomp: num  16 27 27 27 14 27 14 14 27 27 ...
```

I haven't figured out what most of the varinfo fields are used for but we can examine the values.

```
varinfo$vname
```

```
##   [1] CENG     D01Q4    D2002    D2003    D69      D79A     D8095
##   [8] D81      D83      D86      D87      DCON     DDOCKM   DDOCKX
##  [15] DELRFF   DEUC     DFMPRR   DFPDBT   DFPEX    DFPSRP   DGLPRD
##  [22] DMPALT   DMPEX    DMPGEN   DMPINTAY DMPRR    DMPSTB   DMPTAY
##  [29] DMPTLR   DMPTLUR  DMPTMAX  DMPTPI   DMPTR    DMPTRSH  DPADJ
##  [36] DPGAP    DRSTAR   EC       ECD      ECH      ECNIA    ECNIAN
##  [43] ECO      EGF      EGFI     EGFIN    EGFIT    EGFL     EGFLN
##  [50] EGFLT    EGFN     EGFO     EGFON    EGFOT    EGPDIN   EGS
##  [57] EGSI     EGSIN    EGSIT    EGSL     EGSLN    EGSLT    EGSN
##  [64] EGSO     EGSON    EGSOT    EH       EHN      EI       EIN
##  [71] EM       EMN      EMO      EMON     EMP      EMPN     EMPT
##  [78] EPD      EPDN     EPI      EPIN     EPS      EPSN     EX
##  [85] EXN      FCBN     FCBRN    FGDP     FGDPT    FNICN    FNILN
##  [92] FNIN     FNIRN    FPC      FPCM     FPI10    FPI10T   FPIC
##  [99] FPITRG   FPX      FPXM     FPXR     FPXRR    FPXRRT   FRL10
## [106] FRS10    FRSTAR   FTCIN    FXGAP    FYNICN   FYNILN   FYNIN
## [113] GFDBTN   GFDRT    GFINTN   GFS      GFSN     GFSRPN   GFSRT
## [120] GFSUB    GFSUBN   GFT      GFTN     GFTRD    GFTRT    GSDBTN
## [127] GSDRT    GSINTN   GSSRPN   GSSRT    GSSUB    GSSUBN   GST
## [134] GSTN     GSTRD    GSTRT    HGEMP    HGGDP    HGGDPT   HGPCDR
## [141] HGPDR    HGPIR    HGPKIR   HGPPSR   HGVPD    HGVPI    HGVPS
## [148] HGX      HGYNID   HKS      HKSR     HLEPT    HLPRDT   HMFPT
## [155] HQLFPR   HQLWW    HUQPCT   HUXB     HXBT     JCCACN   JCCAN
## [162] JKCD     JRCD     JRH      JRPD     JRPI     JRPS     JYGFEN
## [169] JYGFGN   JYGSEN   JYGSGN   JYNCN    KCD      KH       KI
```

```
## [176] KPD      KPI      KPS      KS       LEF      LEFT     LEH
## [183] LEO      LEP      LEPPOT   LES      LEST     LEUC     LF
## [190] LFPR     LHP      LPRDT    LQUALT   LUR      LURBLS   LURNAT
## [197] LURTRSH  LWW      MEI      MEP      MFPT     N16      PCDR
## [204] PCENG    PCENGR   PCER     PCFR     PCFRT    PCHR     PCNIA
## [211] PCOR     PCPI     PCPIX    PCSTAR   PCXFE    PGDP     PGFIR
## [218] PGFL     PGFOR    PGSIR    PGSL     PGSOR    PHOUSE   PHR
## [225] PIC4     PICNGR   PICNIA   PICX4    PICXFE   PIECI    PIGDP
## [232] PIPL     PIPXNC   PITARG   PITRSH   PKIR     PKPDR    PL
## [239] PLMIN    PLMINR   PMO      PMP      POIL     POILR    POILRT
## [246] PPDR     PPIR     PPSR     PTR      PWSTAR   PXB      PXG
## [253] PXNC     PXP      PXR      QEC      QECD     QECO     QEH
## [260] QEPD     QEPI     QEPS     QKIR     QLEOR    QLEP     QLF
## [267] QLFPR    QLHP     QLWW     QPCNIA   QPL      QPMO     QPXG
## [274] QPXNC    QPXP     QYNIDN   RBBB     RBBBE    RBBBP    RCAR
## [281] RCCD     RCCH     RCGAIN   REQ      REQP     RFF      RFFALT
## [288] RFFE     RFFFIX   RFFGEN   RFFINTAY RFFMIN   RFFRULE  RFFTAY
## [295] RFFTLR   RFNICT   RFRS10   RFYNIC   RFYNIL   RG10     RG10E
## [302] RG10P    RG30     RG30E    RG30P    RG5      RG5E     RG5P
## [309] RGFINT   RGW      RME      RPD      RRFFE    RRFIX    RRMET
## [316] RRTR     RSPNIA   RSTAR    RTB      RTBE     RTINV    RTPD
## [323] RTPI     RTPS     RTR      T47      TAPDAD   TAPDD    TAPDDP
## [330] TAPDS    TAPDT    TAPSAD   TAPSDA   TAPSSL   TFCIN    TFDIV
## [337] TFIBN    TFPN     TFSIN    TRFCI    TRFCIM   TRFIB    TRFP
## [344] TRFPM    TRFPT    TRFPTX   TRFSI    TRSCI    TRSCIT   TRSIB
## [351] TRSIBT   TRSP     TRSPP    TRSPT    TRSPTX   TRSSI    TRSSIT
## [358] TRYH     TSCIN    TSIBN    TSPN     TSSIN    UCES     UCFS
## [365] UEMOT    UEMP     UFCBR    UFNIR    UFPCM    UFPXM    UFTCIN
## [372] UGFDBT   UGSDBT   UGSINT   UGSSUB   UJCCA    UJCCAC   UJYGFE
## [379] UJYGFG   UJYGSE   UJYGSG   ULEF     ULES     UPCPI    UPCPIX
## [386] UPGFL    UPGSL    UPKPD    UPMP     UPXB     UQPCT    UVEOA
## [393] UVPD     UVPI     UVPS     UXBT     UXENG    UYD      UYHI
## [400] UYHLN    UYHPTN   UYHSN    UYHTN    UYL      UYNI     UYNICP
## [407] UYP      UYSEN    VEO      VEOA     VPD      VPI      VPS
## [414] WDNFCN   WPO      WPON     WPS      WPSN     XB       XBN
## [421] XBO      XBT      XENG     XFS      XFSN     XG       XGAP
## [428] XGAP2    XGDE     XGDEN    XGDI     XGDIN    XGDO     XGDP
## [435] XGDPN    XGDPT    XGDPTN   XGN      XGO      XGPOT    XP
## [442] XPN      YCSN     YDN      YGFSN    YGSSN    YH       YHGAP
## [449] YHIBN    YHIN     YHL      YHLN     YHP      YHPCD    YHPGAP
## [456] YHPNTN   YHPSHR   YHPTN    YHSHR    YHSN     YHT      YHTGAP
## [463] YHTN     YHTSHR   YKIN     YKPDN    YKPSN    YMSDN    YNICPN
## [470] YNIDN    YNIIN    YNILN    YNIN     YNISEN   YPN      ZDIVGR
## [477] ZECD     ZECO     ZEH      ZGAP05   ZGAP10   ZGAP30   ZGAPC2
## [484] ZLHP     ZPI10    ZPI10F   ZPI5     ZPIB5    ZPIC30   ZPIC58
```

```
## [491] ZPICXFE  ZPIECI   ZRFF10   ZRFF30   ZRFF5    ZVPD     ZVPI
## [498] ZVPS     ZXBD     ZXBI     ZXBS     ZYH      ZYHP     ZYHPST
## [505] ZYHST    ZYHT     ZYHTST   ZYNID
## 509 Levels: CENG D01Q4 D2002 D2003 D69 D79A D8095 D81 D83 D86 D87 ... ZZZBLANK

table(varinfo$vtype)

##
##       B B.1 B.2 B.3 B.4 B.6 B.7   I I.3 X.1 X.2 X.3 X.4 X.5 X.7
##   0  54  12   1   1  73   1   4 239   1   3  72  34   6   1   6

table(varinfo$vrule)

##
##       A
## 126 382

str(varinfo$sector)

##  Factor w/ 387 levels "","sector_a.1",..: 69 1 1 1 1 1 1 1 1 1 ...

summary(varinfo$var7)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     1.0     1.0     4.0     2.7     4.0     5.0       4

table(varinfo$var7)

##
##   1   2   3   4   5
## 201  25   3 274   1

table(varinfo$stoch)

##
##      FN  GV  IN  IS  LB  NO  OT  PR  RW  ST
##   0  12  15   4  10   7 437  10   9   2   2

table(varinfo$var8)

##
##   0   1   2   7
## 413  90   1   1

table(varinfo$var9)

##
##   0   1   2   3   4
## 370  24  80  21  10
```

```
table(varinfo$decomp)
```

```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## 30  2 33  5  5  4 12  4  7  7 11  3 24 49 34  8  8 30  2 11 16 53 44 26 17
## 26 27 28 29
## 32 18 12  1
```

```
table(varinfo$vtype,varinfo$stoch)
```

```
##
##            FN  GV  IN  IS  LB  NO  OT  PR  RW  ST
##         0   0   0   0   0   0   0   0   0   0   0
##   B     0   0   0   0   0   6  46   0   0   0   2
##   B.1   0   3   5   0   0   0   2   0   2   0   0
##   B.2   0   0   0   0   0   0   1   0   0   0   0
##   B.3   0   0   0   0   0   0   1   0   0   0   0
##   B.4   0   9  10   4  10   0  21  10   7   2   0
##   B.6   0   0   0   0   0   0   1   0   0   0   0
##   B.7   0   0   0   0   0   1   3   0   0   0   0
##   I     0   0   0   0   0   0 239   0   0   0   0
##   I.3   0   0   0   0   0   0   1   0   0   0   0
##   X.1   0   0   0   0   0   0   3   0   0   0   0
##   X.2   0   0   0   0   0   0  72   0   0   0   0
##   X.3   0   0   0   0   0   0  34   0   0   0   0
##   X.4   0   0   0   0   0   0   6   0   0   0   0
##   X.5   0   0   0   0   0   0   1   0   0   0   0
##   X.7   0   0   0   0   0   0   6   0   0   0   0
```

```
table(varinfo$var7,varinfo$var9)
```

```
##
##       0   1   2   3   4
##   1 201   0   0   0   0
##   2   4   0  21   0   0
##   3   2   0   0   1   0
##   4 161  24  59  20  10
##   5   1   0   0   0   0
```

```
table(varinfo$var7,varinfo$stoch)
```

```
##
##          FN  GV  IN  IS  LB  NO  OT  PR  RW  ST
##   1   0  10   9   0   0   5 173   0   2   1   1
##   2   0   2   0   0   0   0  23   0   0   0   0
##   3   0   0   0   0   0   0   3   0   0   0   0
##   4   0   0   6   4  10   2 233  10   7   1   1
##   5   0   0   0   0   0   0   1   0   0   0   0
```

```
table(varinfo$var8,varinfo$stoch)

##
##          FN  GV  IN  IS  LB  NO  OT  PR  RW  ST
##   0   0  12   9   0   0   5 383   0   2   1   1
##   1   0   0   6   3  10   2  50  10   7   1   1
##   2   0   0   0   0   0   0   1   0   0   0   0
##   7   0   0   0   1   0   0   0   0   0   0   0

table(varinfo$var9,varinfo$stoch)

##
##          FN  GV  IN  IS  LB  NO  OT  PR  RW  ST
##   0   0  12  15   4  10   7 299  10   9   2   2
##   1   0   0   0   0   0   0  24   0   0   0   0
##   2   0   0   0   0   0   0  80   0   0   0   0
##   3   0   0   0   0   0   0  21   0   0   0   0
##   4   0   0   0   0   0   0  10   0   0   0   0
```

Here we create the support file from the data.frame.

```
write.csv(varinfo,"support/varinfo.csv")
```

0.2 stdver_coeffs.txt

```
raw = readLines("frbus_package/mods/stdver_coeffs.txt")
str(raw)

##  chr [1:173] "" ...
```

```
coeffs = lapply(raw[which(raw!="" & raw!="theend")], function(x) strsplit(x, "\t"))
coeffs = data.frame(
           name = unlist(lapply(coeffs, function(x) x[[1]][1])),
           len = as.numeric(lapply(coeffs, function(x) x[[1]][2])),
           vec = unlist(lapply(coeffs, function(x) x[[1]][3])),
             stringsAsFactors = FALSE)
coeffs$vec = lapply(coeffs$vec, function(x) as.numeric(strsplit(x, ",")[[1]]))
row.names(coeffs) = coeffs$name
coeffs["y_emo","vec"]

## [[1]]
## [1]  0.01701497 -0.19847532  1.35232826  1.67397668  0.35662832  0.38031136
```

```
coeffs["y_emo","vec"][[1]][3]
```

```
## [1] 1.352328
```

```
coeffs["y_emo","len"]
```

```
## [1] 6
```

```
length(coeffs["y_emo","vec"][[1]])
```

```
## [1] 6
```

```
# rownames(coeffs)
# coeffs$vec
# coeffs = subset(coeffs, select=c(vec))
# str(coeffs)
# coeffs["y_emo"]
# coeffs$vec["y_emo"]
```

0.3 stdver_eqs.txt

```
raw = readLines("frbus_package/mods/stdver_eqs.txt")
raw2 = paste(raw[which(raw!="" & raw!="theend")], collapse="<endline>")
raw3 = gsub(" *_<endline> *", " ", raw2)
raw4 = gsub("[[:space:]]+", " ", raw3)
raw5 = strsplit(raw4, "<endline>")
raw6 = raw5[[1]]
gsub(":.*$", "", raw6)
```

```
##   [1] "ceng"     "delrff"   "dmptlur"  "dmptmax"  "dmptpi"   "dmptr"
##   [7] "dpadj"    "dpgap"    "ec"       "ecd"      "ech"      "ecnia"
##  [13] "ecnian"   "eco"      "egf"      "egfi"     "egfin"    "egfit"
##  [19] "egfl"     "egfln"    "egflt"    "egfn"     "egfo"     "egfon"
##  [25] "egfot"    "egpdin"   "egs"      "egsi"     "egsin"    "egsit"
##  [31] "egsl"     "egsln"    "egslt"    "egsn"     "egso"     "egson"
##  [37] "egsot"    "eh"       "ehn"      "ei"       "ein"      "em"
##  [43] "emn"      "emo"      "emon"     "emp"      "empn"     "empt"
##  [49] "epd"      "epdn"     "epi"      "epin"     "eps"      "epsn"
##  [55] "ex"       "exn"      "fcbn"     "fcbrn"    "fgdp"     "fgdpt"
##  [61] "fnicn"    "fniln"    "fnin"     "fnirn"    "fpc"      "fpcm"
##  [67] "fpi10"    "fpi10t"   "fpic"     "fpx"      "fpxm"     "fpxr"
##  [73] "fpxrr"    "frl10"    "frs10"    "frstar"   "ftcin"    "fxgap"
##  [79] "fynicn"   "fyniln"   "fynin"    "gfdbtn"   "gfintn"   "gfs"
```

```
##  [85] "gfsn"     "gfsrpn"   "gfsub"    "gfsubn"   "gft"      "gftn"
##  [91] "gftrd"    "gsdbtn"   "gsintn"   "gssrpn"   "gssub"    "gssubn"
##  [97] "gst"      "gstn"     "gstrd"    "hgemp"    "hggdp"    "hggdpt"
## [103] "hgpdr"    "hgpir"    "hgpkir"   "hgppsr"   "hgvpd"    "hgvpi"
## [109] "hgvps"    "hgx"      "hgynid"   "hks"      "hlept"    "hlprdt"
## [115] "hmfpt"    "hqlfpr"   "hqlww"    "huqpct"   "huxb"     "hxbt"
## [121] "jccacn"   "jccan"    "jkcd"     "jygfen"   "jygfgn"   "jygsen"
## [127] "jygsgn"   "jyncn"    "kcd"      "kh"       "ki"       "kpd"
## [133] "kpi"      "kps"      "ks"       "lef"      "left"     "leh"
## [139] "leo"      "lep"      "leppot"   "les"      "lest"     "lf"
## [145] "lfpr"     "lhp"      "lprdt"    "lur"      "lurbls"   "lurnat"
## [151] "lww"      "mei"      "mep"      "mfpt"     "pcdr"     "pceng"
## [157] "pcengr"   "pcer"     "pcfr"     "pchr"     "pcnia"    "pcor"
## [163] "pcpi"     "pcpix"    "pcxfe"    "pgdp"     "pgfir"    "pgfl"
## [169] "pgfor"    "pgsir"    "pgsl"     "pgsor"    "phouse"   "phr"
## [175] "pic4"     "picngr"   "picnia"   "picx4"    "picxfe"   "pieci"
## [181] "pigdp"    "pipl"     "pipxnc"   "pkpdr"    "pl"       "plmin"
## [187] "pmo"      "pmp"      "poil"     "poilr"    "ppdr"     "ppir"
## [193] "ppsr"     "ptr"      "pwstar"   "pxb"      "pxg"      "pxnc"
## [199] "pxp"      "pxr"      "qec"      "qecd"     "qeco"     "qeh"
## [205] "qepd"     "qepi"     "qeps"     "qkir"     "qlep"     "qlf"
## [211] "qlfpr"    "qlhp"     "qlww"     "qpcnia"   "qpl"      "qpmo"
## [217] "qpxg"     "qpxnc"    "qpxp"     "qynidn"   "rbbb"     "rbbbe"
## [223] "rbbbp"    "rcar"     "rccd"     "rcch"     "rcgain"   "req"
## [229] "reqp"     "rff"      "rffalt"   "rffe"     "rffgen"   "rffintay"
## [235] "rffrule"  "rfftay"   "rfftlr"   "rfynic"   "rfynil"   "rg10"
## [241] "rg10e"    "rg10p"    "rg30"     "rg30e"    "rg30p"    "rg5"
## [247] "rg5e"     "rg5p"     "rgfint"   "rgw"      "rme"      "rpd"
## [253] "rrffe"    "rrmet"    "rrtr"     "rspnia"   "rstar"    "rtb"
## [259] "rtbe"     "rtinv"    "rtpd"     "rtpi"     "rtps"     "rtr"
## [265] "tapdd"    "tapsda"   "tfcin"    "tfibn"    "tfpn"     "tfsin"
## [271] "trfci"    "trfp"     "trfpt"    "trsci"    "trsib"    "trsp"
## [277] "trspt"    "trssi"    "tryh"     "tscin"    "tsibn"    "tspn"
## [283] "tssin"    "uces"     "ucfs"     "uqpct"    "uxbt"     "veo"
## [289] "veoa"     "vpd"      "vpi"      "vps"      "wdnfcn"   "wpo"
## [295] "wpon"     "wps"      "wpsn"     "xb"       "xbn"      "xbo"
## [301] "xbt"      "xeng"     "xfs"      "xfsn"     "xg"       "xgap"
## [307] "xgap2"    "xgde"     "xgden"    "xgdi"     "xgdin"    "xgdo"
## [313] "xgdp"     "xgdpn"    "xgdpt"    "xgdptn"   "xgn"      "xgo"
## [319] "xgpot"    "xp"       "xpn"      "ycsn"     "ydn"      "ygfsn"
## [325] "ygssn"    "yh"       "yhgap"    "yhibn"    "yhin"     "yhl"
## [331] "yhln"     "yhp"      "yhpcd"    "yhpgap"   "yhpntn"   "yhpshr"
## [337] "yhptn"    "yhshr"    "yhsn"     "yht"      "yhtgap"   "yhtn"
## [343] "yhtshr"   "ykin"     "ykpdn"    "ykpsn"    "ynicpn"   "ynidn"
## [349] "yniin"    "yniln"    "ynin"     "ynisen"   "ypn"      "zdivgr"
```

```
## [355] "zecd"     "zeco"     "zeh"      "zgap05"   "zgap10"   "zgap30"
## [361] "zgapc2"   "zlhp"     "zpi10"    "zpi10f"   "zpi5"     "zpib5"
## [367] "zpic30"   "zpic58"   "zpicxfe"  "zpieci"   "zrff10"   "zrff30"
## [373] "zrff5"    "zvpd"     "zvpi"     "zvps"     "zxbd"     "zxbi"
## [379] "zxbs"     "zyh"      "zyhp"     "zyhpst"   "zyhst"    "zyht"
## [385] "zyhtst"   "zynid"
```

raw6

```
##   [1] "ceng: d( log(ceng), 0, 1 ) - ceng_aerr = y_ceng(1) * (log(ceng(-1))-log(xg(-
##   [2] "delrff: delrff - delrff_aerr = rff - rff(-1) "
##   [3] "dmptlur: dmptlur - dmptlur_aerr = 1/(1+exp(y_dmptlur(1)*(lur-lurtrsh))) "
##   [4] "dmptmax: dmptmax - dmptmax_aerr = (@recode((dmptlur)>(dmptpi),dmptlur,dmptpi)
##   [5] "dmptpi: dmptpi - dmptpi_aerr = 1/(1+exp(y_dmptpi(1)*(zpic58-pitrsh))) "
##   [6] "dmptr: dmptr - dmptr_aerr = (@recode((dmptmax)>(dmptr(-1)),dmptmax,dmptr(-1))
##   [7] "dpadj: dpadj - dpadj_aerr - dpadj(-1) = y_dpadj(1) * dpgap(-1) "
##   [8] "dpgap: dpgap - dpgap_aerr = pipxnc/400 - ( .5 * (ehn/(xpn - ecnian)+ ehn(-1)/
##   [9] "ec: log(ec) - ec_aerr = log(ec(-1)) + .5 * (pcor*pcnia*eco/(ec*pcnia) + pcor
##  [10] "ecd: d( log(ecd), 0, 1) - ecd_aerr = y_ecd(1) * log(qecd(-1)/ecd(-1)) + y_ecd
##  [11] "ech: d( (ech)/kh(-1), 0, 1 ) - ech_aerr = y_ech(1) + y_ech(2) * ech(-1)/kh(-1
##  [12] "ecnia: log(ecnia) - ecnia_aerr = log(ecnia(-1)) + .5 * .01 * (pcor*pcnia*eco/
##  [13] "ecnian: ecnian - ecnian_aerr = .01*pcnia*ecnia "
##  [14] "eco: d( log(eco), 0, 1) - eco_aerr = (y_eco(1) * log(qeco(-1)/eco(-1)) + y_ec
##  [15] "egf: log(egf) - egf_aerr = log(egf(-1)) + .5 * (egfon/egfn + egfon(-1)/egfn(-
##  [16] "egfi: d( log(egfi), 0, 1 ) - egfi_aerr = y_egfi(1) + y_egfi(2) * log(egfi(-1)
##  [17] "egfin: egfin - egfin_aerr = .01 * pxp * pgfir * egfi "
##  [18] "egfit: d( log(egfit), 0, 1 ) - egfit_aerr = y_egfit(1) + y_egfit(2) * log(.01
##  [19] "egfl: d( log(egfl), 0, 1 ) - egfl_aerr = y_egfl(1) + y_egfl(2) * log(egfl(-1)
##  [20] "egfln: egfln - egfln_aerr = .01 * pgfl * egfl "
##  [21] "egflt: d( log(egflt), 0, 1 ) - egflt_aerr = y_egflt(1) + y_egflt(2) * log(.01
##  [22] "egfn: egfn - egfn_aerr = egfln + egfin + egfon "
##  [23] "egfo: d( log(egfo), 0, 1 ) - egfo_aerr = y_egfo(1) + y_egfo(2) * log(egfo(-1)
##  [24] "egfon: egfon - egfon_aerr = .01 * pxp * pgfor * egfo "
##  [25] "egfot: d( log(egfot), 0, 1 ) - egfot_aerr = y_egfot(1) + y_egfot(2) * log(.01
##  [26] "egpdin: egpdin - egpdin_aerr = epdn + epsn + epin + ehn + ein "
##  [27] "egs: log(egs) - egs_aerr = log(egs(-1)) + .5 * (egson/egsn + egson(-1)/egsn(-
##  [28] "egsi: d( log(egsi), 0, 1 ) - egsi_aerr = y_egsi(1) + y_egsi(2) * log(egsi(-1)
##  [29] "egsin: egsin - egsin_aerr = .01 * pxp * pgsir * egsi "
##  [30] "egsit: d( log(egsit), 0, 1 ) - egsit_aerr = y_egsit(1) + y_egsit(2) * log(.01
##  [31] "egsl: d( log(egsl), 0, 1 ) - egsl_aerr = y_egsl(1) + y_egsl(2) * log(egsl(-1)
##  [32] "egsln: egsln - egsln_aerr = .01 * pgsl * egsl "
##  [33] "egslt: d( log(egslt), 0, 1 ) - egslt_aerr = y_egslt(1) + y_egslt(2) * log(.01
##  [34] "egsn: egsn - egsn_aerr = egsln + egsin + egson "
##  [35] "egso: d( log(egso), 0, 1 ) - egso_aerr = y_egso(1) + y_egso(2) * log(egso(-1)
##  [36] "egson: egson - egson_aerr = .01 * pxp * pgsor * egso "
##  [37] "egsot: d( log(egsot), 0, 1 ) - egsot_aerr = y_egsot(1) + y_egsot(2) * log(.01
```

```
##  [38] "eh: d( log(eh), 0, 1 ) - eh_aerr = y_eh(1) * log(qeh(-1)/eh(-1)) + y_eh(2) * d( log(eh(
##  [39] "ehn: ehn - ehn_aerr = .01 * phr * pxp * eh "
##  [40] "ei: ei - ei_aerr = 4*d( ki, 0, 1 ) "
##  [41] "ein: ein - ein_aerr = .01*pxp*pkir*ei "
##  [42] "em: log(em) - em_aerr = log(em(-1)) + .5 * (emon/emn + emon(-1)/emn(-1)) * d(log(emo),
##  [43] "emn: emn - emn_aerr = emon + empn "
##  [44] "emo: d( log(emo), 0, 1 ) - emo_aerr = y_emo(1) + y_emo(2) * log(emo(-1)*(pmo(-1)/100)/(
##  [45] "emon: emon - emon_aerr = .01 * pmo * emo "
##  [46] "emp: emp - emp_aerr = uemp*(ceng-xeng) "
##  [47] "empn: empn - empn_aerr = .01*pmp*emp "
##  [48] "empt: d( log(empt), 0, 1) - empt_aerr = y_empt(1) * log(emp(-1)/empt(-1)) + y_empt(2) *
##  [49] "epd: d( log(epd), 0, 1 ) - epd_aerr = ( y_epd(1)*(log(qepd(-2)/epd(-2))) + ( y_epd(2) *
##  [50] "epdn: epdn - epdn_aerr = 0.01*ppdr*pxp*epd "
##  [51] "epi: d( log(epi), 0, 1 ) - epi_aerr = ( y_epi(1)*(log(qepi(-2)/epi(-2))) + ( y_epi(2) *
##  [52] "epin: epin - epin_aerr = 0.01*ppir*pxp*epi "
##  [53] "eps: d( log(eps), 0, 1 ) - eps_aerr = (y_eps(1) * log(qeps(-2)/eps(-2)) + ( y_eps(2) *
##  [54] "epsn: epsn - epsn_aerr = .01 * ppsr * pxp * eps "
##  [55] "ex: d( log(ex), 0, 1 ) - ex_aerr = y_ex(1) + y_ex(2) * log(ex(-1)*(pxr(-1)*pxp(-1)*fpx(
##  [56] "exn: exn - exn_aerr = .01*pxp*pxr*ex "
##  [57] "fcbn: fcbn - fcbn_aerr = exn - emn + fynin + fcbrn "
##  [58] "fcbrn: fcbrn - fcbrn_aerr = ufcbr*pxg*xgpot/100 "
##  [59] "fgdp: fgdp - fgdp_aerr = fgdpt*exp(fxgap/100) "
##  [60] "fgdpt: d( log(fgdpt), 0, 1 ) - fgdpt_aerr = y_fgdpt(1) + y_fgdpt(2) * log(fgdpt(-1)/xgd
##  [61] "fnicn: d(fnicn, 0, 1)/xgdptn - fnicn_aerr = .54 * d( log(fpc), 0, 1)*fnicn(-1)/xgdptn -
##  [62] "fniln: fniln - fniln_aerr = fnicn - fnin "
##  [63] "fnin: d( fnin, 0, 1 ) - fnin_aerr = .25*fcbn + .54 * (d( log(fpc), 0, 1) * fnicn(-1)) -
##  [64] "fnirn: fnirn - fnirn_aerr = ufnir * xgdpn "
##  [65] "fpc: fpc - fpc_aerr = fpc(-1)*exp(fpic/400) "
##  [66] "fpcm: fpcm - fpcm_aerr = ufpcm*fpc "
##  [67] "fpi10: fpi10-fpi10_aerr = y_fpi10(1) * ( ( fpi10(-1) + fpi10(-2) + fpi10(-3) + fpi10(-4
##  [68] "fpi10t: fpi10t-fpi10t_aerr = y_fpi10t(1) * fpi10t(-1) + y_fpi10t(2) * fpi10 "
##  [69] "fpic: fpic-fpic_aerr = y_fpic(1) + y_fpic(2) * fpi10 + y_fpic(3) * fpic(-1) "
##  [70] "fpx: fpx - fpx_aerr = fpxr*fpc/pcpi "
##  [71] "fpxm: fpxm - fpxm_aerr = ufpxm*fpx*fpcm/fpc "
##  [72] "fpxr: log(fpxr) - fpxr_aerr - log(fpxrr) = y_fpxr(1)*(rg10e-zpi10f-frl10+fpi10t) + y_fp
##  [73] "fpxrr: d( log(fpxrr), 0, 1 ) - fpxrr_aerr = y_fpxrr(1) * log(fpxrrt(-1)/fpxrr(-1)) + y_
##  [74] "frl10: frl10 - frl10(-1) - frl10_aerr = y_frl10(1) + y_frl10(2) * (frl10(-1) - frs10(-1
##  [75] "frs10: frs10 - frs10_aerr = dfmprr * (y_frs10(1) + y_frs10(2) * frstar(-1) + y_frs10(3)
##  [76] "frstar: frstar - frstar_aerr = y_frstar(1) * frstar(-1) + y_frstar(2) * (frs10 - ( fpi1
##  [77] "ftcin: ftcin - ftcin_aerr = uftcin * ynicpn "
##  [78] "fxgap: fxgap - fxgap_aerr = + y_fxgap(1) * fxgap(-1) + y_fxgap(2) * fxgap(-2) + y_fxgap
##  [79] "fynicn: fynicn - fynicn_aerr = .01*rfynic*fnicn(-1) "
##  [80] "fyniln: fyniln - fyniln_aerr = .01*rfynil*fniln(-1) "
##  [81] "fynin: fynin - fynin_aerr = fynicn - fyniln "
##  [82] "gfdbtn: gfdbtn - gfdbtn_aerr = ugfdbt*(gfdbtn(-1) - .25*gfsrpn + .25*egfin - .25*jygfgn
```

```
##  [83] "gfintn: gfintn - gfintn_aerr = rgfint*gfdbtn(-1) "
##  [84] "gfs: d( log(gfs), 0, 1 ) - gfs_aerr = y_gfs(1) + y_gfs(2) * log(gfsn(-1)/xgdp
##  [85] "gfsn: gfsn - gfsn_aerr = .01*pgdp*gfs "
##  [86] "gfsrpn: gfsrpn - gfsrpn_aerr = tfpn + tfcin + tfibn + tfsin + tfdiv - egfln -
##  [87] "gfsub: d( log(gfsub), 0, 1 ) - gfsub_aerr = y_gfsub(1) + y_gfsub(2) * log(gfs
##  [88] "gfsubn: gfsubn - gfsubn_aerr = .01*pgdp*gfsub "
##  [89] "gft: gft - gft_aerr = (gftrd+gftrt)*xgdpt "
##  [90] "gftn: gftn - gftn_aerr = .01*pgdp*gft "
##  [91] "gftrd: gftrd - gftrd_aerr = y_gftrd(1) + y_gftrd(2) * gftrd(-1) + y_gftrd(3)
##  [92] "gsdbtn: gsdbtn - gsdbtn_aerr = ugsdbt*(gsdbtn(-1) - .25*gssrpn + .25 * egsin
##  [93] "gsintn: gsintn - gsintn_aerr = rgfint*gsdbtn(-1) + ugsint*xbn "
##  [94] "gssrpn: gssrpn - gssrpn_aerr = tspn + tscin + tsibn + tssin + gfsn - egsln -
##  [95] "gssub: gssub - gssub_aerr = ugssub*xgdpt "
##  [96] "gssubn: gssubn - gssubn_aerr = .01*pgdp*gssub "
##  [97] "gst: gst - gst_aerr = (gstrd+gstrt)*xgdpt "
##  [98] "gstn: gstn - gstn_aerr = .01*pgdp*gst "
##  [99] "gstrd: gstrd - gstrd_aerr = y_gstrd(1) + y_gstrd(2) * gstrd(-1) + y_gstrd(3)
## [100] "hgemp: hgemp - hgemp_aerr = y_hgemp(1) * hgemp(-1) + y_hgemp(2) * 400*log(emp
## [101] "hggdp: hggdp - hggdp_aerr = 400*d( log(xgdp), 0, 1 ) "
## [102] "hggdpt: hggdpt - hggdpt_aerr = hxbt + huxb "
## [103] "hgpdr: hgpdr - hgpdr_aerr = y_hgpdr(1) * hgpdr(-1) + y_hgpdr(2) * 400*log(ppd
## [104] "hgpir: hgpir - hgpir_aerr = y_hgpir(1) * hgpir(-1) + y_hgpir(2) * 400*log(ppi
## [105] "hgpkir: hgpkir - hgpkir_aerr = y_hgpkir(1) * hgpkir(-1) + y_hgpkir(2) * 400*l
## [106] "hgppsr: hgppsr - hgppsr_aerr = y_hgppsr(1) * hgppsr(-1) + y_hgppsr(2) * 400*l
## [107] "hgvpd: hgvpd - hgvpd_aerr = y_hgvpd(1) * hgvpd(-1) + y_hgvpd(2) * log(vpd/vpd
## [108] "hgvpi: hgvpi - hgvpi_aerr = y_hgvpi(1) * hgvpi(-1) + y_hgvpi(2) * log(vpi/vpi
## [109] "hgvps: hgvps - hgvps_aerr = y_hgvps(1) * hgvps(-1) + y_hgvps(2) * log(vps/vps
## [110] "hgx: hgx - hgx_aerr = (.7*(hlept + hqlww + 400*d( log(lqualt), 0, 1 )) + .265
## [111] "hgynid: hgynid - hgynid_aerr = 400*d( log((ynicpn-tfcin-tscin)*.5/pxg), 0, 1
## [112] "hks: hks - hks_aerr = 400 * (ykpdn * d( log(kpd), 0, 1 ) + ykpsn * d( log(kps
## [113] "hlept: hlept - hlept_aerr = (1-dmpstb) * 400 * (hqlfpr * n16 * (1-.01*lurnat-
## [114] "hlprdt: hlprdt - hlprdt_aerr = hgx - hlept - hqlww "
## [115] "hmfpt: hmfpt - hmfpt_aerr = y_hmfpt(1) + y_hmfpt(2)*hmfpt(-1) "
## [116] "hqlfpr: hqlfpr - hqlfpr_aerr = y_hqlfpr(1) + y_hqlfpr(2)*hqlfpr(-1) "
## [117] "hqlww: hqlww - hqlww_aerr = y_hqlww(1) * hqlww(-1) + (1-y_hqlww(1)) * y_hqlww
## [118] "huqpct: huqpct - huqpct_aerr = y_huqpct(1) + y_huqpct(2)*huqpct(-1) "
## [119] "huxb: huxb - huxb_aerr = (1-dglprd) *(y_huxb(1) + y_huxb(2)*huxb(-1)) "
## [120] "hxbt: hxbt - hxbt_aerr = ( hgx - .5 *(.035*empn/(.01*pceng*ceng) + .035*empn
## [121] "jccacn: jccacn - jccacn_aerr = ujccac*(jccan - jygfgn - jygfen - jygsgn - jyg
## [122] "jccan: jccan - jccan_aerr = jygfgn + jygfen + jygsgn + jygsen + .01*ujcca*pxp
## [123] "jkcd: jkcd - jkcd_aerr = jrcd * kcd(-1) "
## [124] "jygfen: jygfen - jygfen_aerr = ujygfe * (.01 * pgdp * xgdpt) "
## [125] "jygfgn: jygfgn - jygfgn_aerr = ujygfg * (.01 * pgdp * xgdpt) "
## [126] "jygsen: jygsen - jygsen_aerr = ujygse * (.01 * pgdp * xgdpt) "
## [127] "jygsgn: jygsgn - jygsgn_aerr = ujygsg * (.01 * pgdp * xgdpt) "
```

```
## [128] "jyncn: jyncn - jyncn_aerr = jccan - jccacn - jygfgn - jygfen - jygsgn - jygsen "
## [129] "kcd: kcd - kcd_aerr = .25*ecd + (1-jrcd/4)*kcd(-1) "
## [130] "kh: kh - kh_aerr = .25*eh + (1-jrh/4)*kh(-1) "
## [131] "ki: d( log(ki), 0, 1 ) - ki_aerr = y_ki(5) + y_ki(1) * (log(qkir) - log(ki(-1)/xfs(-1))
## [132] "kpd: kpd - kpd_aerr = 0.25 * epd + (1-jrpd/4) * kpd(-1) "
## [133] "kpi: kpi - kpi_aerr = 0.25 * epi + (1-jrpi/4) * kpi(-1) "
## [134] "kps: kps - kps_aerr = 0.25 * eps + (1-jrps/4) * kps(-1) "
## [135] "ks: log(ks) - ks_aerr = log(ks(-1)) + hks/400 "
## [136] "lef: d( log(lef), 0, 1 ) - lef_aerr = d( log(ulef), 0, 1 ) + d( log(egfl), 0, 1 ) - dgl
## [137] "left: left - left_aerr = y_left(1) * left(-1) * (hqlfpr+n16/n16(-1)) + y_left(2) * lef
## [138] "leh: leh - leh_aerr = lep + leo + les + lef "
## [139] "leo: log(leo) - leo_aerr = log(qleor*qlf) + y_leo(1)*log(leo(-1)/(qleor(-1)*qlf(-1))) +
## [140] "lep: lep - lep_aerr = lhp / lww "
## [141] "leppot: leppot - leppot_aerr = qlf*(1-.01*lurnat - qleor) - left - lest "
## [142] "les: d( log(les), 0, 1 ) - les_aerr = d( log(ules), 0, 1 ) + d( log(egsl), 0, 1 ) - dgl
## [143] "lest: lest - lest_aerr = y_lest(1) * lest(-1) * (hqlfpr+n16/n16(-1)) + y_lest(2) * les
## [144] "lf: lf - lf_aerr = lfpr * n16 "
## [145] "lfpr: d( lfpr, 0, 1) - lfpr_aerr = hqlfpr + y_lfpr(1) * (qlfpr(-1) - lfpr(-1)) + y_lfpr
## [146] "lhp: d( log(lhp), 0, 1 ) - lhp_aerr = y_lhp(1) * (log(qlhp(-1)/lhp(-1))-d( log(mfpt), 0
## [147] "lprdt: log(lprdt) - lprdt_aerr = log(xgpot) - log(leppot) - log(qlww) "
## [148] "lur: lur - lur_aerr = 100*(1 - leh/lf) "
## [149] "lurbls: lurbls - lurbls_aerr = lur "
## [150] "lurnat: lurnat - lurnat_aerr = lurnat(-1) "
## [151] "lww: d( log(lww), 0, 1 ) - lww_aerr = hqlww/400 + y_lww(1) * log(qlww(-1)/lww(-1)) + y_
## [152] "mei: log(mei) - mei_aerr = y_mei(1) * log(mei(-1)) "
## [153] "mep: log(mep) - mep_aerr = y_mep(1) * log(mep(-1)) "
## [154] "mfpt: log(mfpt) - mfpt_aerr = y_mfpt(1) + log(mfpt(-1)) + hmfpt/400 "
## [155] "pcdr: d(log(pcdr), 0, 1) - pcdr_aerr = y_pcdr(1) + y_pcdr(2)*d(log(pcdr(-1)), 0, 1) "
## [156] "pceng: pceng - pceng_aerr = pcengr*pxb "
## [157] "pcengr: d( log(pcengr), 0, 1 ) - pcengr_aerr = y_pcengr(1) + y_pcengr(2) * d( log(pceng
## [158] "pcer: d( log(pcer), 0, 1 ) - pcer_aerr = y_pcer(1) * log((y_pcer(2) *pceng(-1) + (1-y_p
## [159] "pcfr: d( log(pcfr), 0, 1 ) - pcfr_aerr = y_pcfr(1) * log(pcfr(-1)/pcfrt(-1)) + y_pcfr(2
## [160] "pchr: d(log(pchr), 0, 1) - pchr_aerr = y_pchr(1) + y_pchr(2)*d(log(pchr(-1)), 0, 1) "
## [161] "pcnia: d( log(pcnia), 0, 1 ) - pcnia_aerr = picnia / 400 "
## [162] "pcor: log(pcor) - log(pcor(-1)) - pcor_aerr = (- .5 * .01 * (pcdr*pcnia*ecd/ecnian + pc
## [163] "pcpi: pcpi - pcpi_aerr = upcpi * exp(.025*log(pcer)) * pcnia "
## [164] "pcpix: pcpix - pcpix_aerr = upcpix * pcxfe "
## [165] "pcxfe: d(log(pcxfe), 0, 1) - pcxfe_aerr = picxfe/400 "
## [166] "pgdp: pgdp - pgdp_aerr = 100*xgdpn/xgdp "
## [167] "pgfir: log(pgfir) - pgfir_aerr - log(pgfir(-1)) = y_pgfir(1) + pipxnc/400 + dpadj - d(l
## [168] "pgfl: d( log(pgfl), 0, 1 ) - pgfl_aerr = d( log(upgfl), 0, 1 ) + d( log(pl), 0, 1 ) - d
## [169] "pgfor: log(pgfor) - pgfor_aerr - log(pgfor(-1)) = y_pgfor(1) + pipxnc/400 + dpadj - d(l
## [170] "pgsir: log(pgsir) - pgsir_aerr - log(pgsir(-1)) = y_pgsir(1) + pipxnc/400 + dpadj - d(l
## [171] "pgsl: d( log(pgsl), 0, 1 ) - pgsl_aerr = d( log(upgsl), 0, 1 ) + d( log(pl), 0, 1 ) - d
## [172] "pgsor: log(pgsor) - pgsor_aerr - log(pgsor(-1)) = y_pgsor(1) + pipxnc/400 + dpadj - d(l
```

```
## [173] "phouse: d( log(phouse), 0, 1) - phouse_aerr = y_phouse(1) + y_phouse(2) * d(
## [174] "phr: log(phr) - phr_aerr - log(phr(-1)) = y_phr(1) + pipxnc/400 + dpadj - d(
## [175] "pic4: pic4 - pic4_aerr = 100*(pcnia/pcnia(-4) - 1) "
## [176] "picngr: picngr - picngr_aerr = (d( log(pceng/pxp(-1)), 0, 1 ) * ( pceng*ceng
## [177] "picnia: picnia - picnia_aerr = picxfe + ( ( ucfs + ucfs(-1)) /2) * 400 * d(lo
## [178] "picx4: picx4 - picx4_aerr = 100*(pcxfe/pcxfe(-4) - 1) "
## [179] "picxfe: picxfe - picxfe_aerr = (y_picxfe(1)*picxfe(-1) + y_picxfe(3)*zpicxfe
## [180] "pieci: pieci - pieci_aerr = (.25*y_pieci(1)*((1-y_pieci(4))*(pieci(-1)+pieci
## [181] "pigdp: pigdp - pigdp_aerr = 400*d( log(pgdp), 0, 1 ) "
## [182] "pipl: pipl - pipl_aerr = pieci "
## [183] "pipxnc: pipxnc - pipxnc_aerr = picnia - 1.99 * 400 * huqpct + y_pipxnc(1) *
## [184] "pkpdr: pkpdr - pkpdr_aerr = upkpd * ppdr "
## [185] "pl: log(pl) - pl_aerr = log(pl(-1)) + pipl/400 "
## [186] "plmin: plmin - plmin_aerr = plminr*.01*pl "
## [187] "pmo: d( log(pmo), 0, 1 ) - pmo_aerr = y_pmo(1) + y_pmo(2) * (log(qpmo) + .64*
## [188] "pmp: pmp - pmp_aerr = upmp*poil "
## [189] "poil: poil - poil_aerr = poilr*pxb "
## [190] "poilr: d( log(poilr), 0, 1 ) - poilr_aerr = y_poilr(1) * log(poilr(-1)/poilrt
## [191] "ppdr: log(ppdr) - ppdr_aerr - log(ppdr(-1)) = y_ppdr(1) + pipxnc/400 + dpadj
## [192] "ppir: log(ppir) - ppir_aerr - log(ppir(-1)) = pipxnc/400 + dpadj - d(log(pxp)
## [193] "ppsr: log(ppsr) - ppsr_aerr - log(ppsr(-1)) = y_ppsr(1) + pipxnc/400 + dpadj
## [194] "ptr: ptr - ptr_aerr = y_ptr(1)*ptr(-1) + y_ptr(2)*picxfe(-1)+ y_ptr(3)*pitarg
## [195] "pwstar: pwstar - pwstar_aerr = y_pwstar(1) + y_pwstar(2)*pwstar(-1) "
## [196] "pxb: pxb - pxb_aerr = upxb*pgdp "
## [197] "pxg: pxg - pxg_aerr = 100*xgn/xg "
## [198] "pxnc: d( log(pxnc), 0, 1 ) - pxnc_aerr = pipxnc/400 "
## [199] "pxp: d( log(pxp), 0, 1 ) - pxp_aerr = .5*( ecnian/xpn + ecnian(-1)/xpn(-1)) +
## [200] "pxr: log(pxr) - pxr_aerr - log(pxr(-1)) = y_pxr(1) + pipxnc/400 + dpadj - d(
## [201] "qec: qec - qec_aerr = y_qec(1) * zyh + y_qec(2) * (dcon*(zyh-zyht)) + y_qec(3
## [202] "qecd: qecd - qecd_aerr = qec * (jrcd/4 + hggdpt/400 + y_qecd(1)*hgpcdr/400) +
## [203] "qeco: log(qeco) - qeco_aerr = log(qec) - log(pcor) + y_qeco(1) "
## [204] "qeh: qeh - qeh_aerr = qec * (jrh/4 + hggdpt/400) * exp(y_qeh(1) - log(phr*pxp
## [205] "qepd: log(qepd) - qepd_aerr = y_qepd(1) + y_qepd(2) * log(xbo) + y_qepd(3) *
## [206] "qepi: log(qepi) - qepi_aerr = y_qepi(1) + y_qepi(2) * log(xbo) + y_qepi(3) *
## [207] "qeps: log(qeps) - qeps_aerr = y_qeps(1) + y_qeps(2) * log(xbo) + y_qeps(3) *
## [208] "qkir: log(qkir) - qkir_aerr = (1-dglprd)*y_qkir(1) + log(qkir(-1)) "
## [209] "qlep: qlep - qlep_aerr = lhp / qlww "
## [210] "qlf: qlf - qlf_aerr = qlfpr * n16 "
## [211] "qlfpr: qlfpr - qlfpr_aerr = qlfpr(-1) + hqlfpr "
## [212] "qlhp: qlhp - qlhp_aerr = xgo/lprdt "
## [213] "qlww: log(qlww) - qlww_aerr = log(qlww(-1)) + hqlww(-1)/400 "
## [214] "qpcnia: log(qpcnia) - qpcnia_aerr = log(qpxp) + log(uqpct) "
## [215] "qpl: log(qpl) - qpl_aerr = log(pl) + y_qpl(1) * log(pxg/qpxg) "
## [216] "qpmo: log(qpmo) - qpmo_aerr = log(qpmo(-1)) + y_qpmo(1) "
## [217] "qpxg: log(qpxg) - qpxg_aerr = log(pwstar) + y_qpxg(1) + y_qpxg(2)*log(pl/lpro
```

```
## [218] "qpxnc: log(qpxnc) - qpxnc_aerr = log(pxnc) + y_qpxnc(1) * log(qpxp/pxp) + y_qpxnc(2) *
## [219] "qpxp: qpxp - qpxp_aerr = 100*(xpn + (.01*qpxg*xg-xgn))/xp "
## [220] "qynidn: log(qynidn) - qynidn_aerr = y_qynidn(1) + y_qynidn(2)*d79a + y_qynidn(3)*log((@
## [221] "rbbb: rbbb - rbbb_aerr = ( ( (0.01*rbbbe + 1)^.5 - 1 ) * 200 ) "
## [222] "rbbbe: rbbbe - rbbbe_aerr = rbbbp + rg10e "
## [223] "rbbbp: rbbbp - rbbbp_aerr = y_rbbbp(1) + y_rbbbp(2) * zgap10 + y_rbbbp(3) * (rbbbp(-1)
## [224] "rcar: rcar - rcar_aerr = y_rcar(1) + y_rcar(2) * d79a + y_rcar(3) * ((1-d79a)*t47) + y_
## [225] "rccd: rccd - rccd_aerr = (@recode((100*jrcd + rcar - zpi5)>( .01),100*jrcd + rcar - zpi
## [226] "rcch: rcch - rcch_aerr = (@recode((100*jrh + (1-trfpm/100)*(rme+100*trspp) - zpi10)>( .
## [227] "rcgain: rcgain - rcgain_aerr = picx4 + y_rcgain(1) + y_rcgain(2) * xgap2 + y_rcgain(3)
## [228] "req: req - req_aerr = rg30e - zpic30 + reqp "
## [229] "reqp: reqp - reqp_aerr = y_reqp(1) + y_reqp(2) * rbbbp + y_reqp(3) * (reqp(-1) - y_reqp
## [230] "rff: rff - rff_aerr = 36000*( (1+.01*rffe)^(1/365) - 1 ) "
## [231] "rffalt: rffalt - rffalt_aerr = y_rffalt(1) + y_rffalt(2) * rff(-1) + y_rffalt(3) * rff(
## [232] "rffe: rffe - rffe_aerr = (1-dmptrsh) * (@recode((rffrule)>( rffmin),rffrule, rffmin)) +
## [233] "rffgen: rffgen - rffgen_aerr = y_rffgen(1) + ( y_rffgen(2) * rffe(-1) + y_rffgen(3) * r
## [234] "rffintay: rffintay - rffintay_aerr = y_rffintay(3) * rffe(-1) + (1-y_rffintay(3)) * (rs
## [235] "rffrule: rffrule - rffrule_aerr = (@recode((dmpex * 100 * ((1+rfffix/36000)^365-1) + dm
## [236] "rfftay: rfftay - rfftay_aerr = rstar + ( picxfe + picxfe(-1) + picxfe(-2) + picxfe(-3))
## [237] "rfftlr: rfftlr - rfftlr_aerr = rstar + y_rfftlr(1) * pitarg + y_rfftlr(2) * ( ( picxfe
## [238] "rfynic: d( rfynic, 0, 1 ) - rfynic_aerr = y_rfynic(1) + y_rfynic(2) * (rfynic(-1)-rfyni
## [239] "rfynil: d( rfynil, 0, 1 ) - rfynil_aerr = y_rfynil(1) + y_rfynil(2) * rfynil(-1) + y_rf
## [240] "rg10: rg10 - rg10_aerr = (( (.01*rg10e + 1)^.5 - 1) * 200) "
## [241] "rg10e: rg10e - rg10e_aerr = zrff10 + rg10p "
## [242] "rg10p: rg10p - rg10p_aerr = y_rg10p(1) + y_rg10p(2) * zgap10 + y_rg10p(3) * d8095 + y_r
## [243] "rg30: rg30 - rg30_aerr = (( (.01*rg30e + 1)^.5 - 1) * 200) "
## [244] "rg30e: rg30e - rg30e_aerr = zrff30 + rg30p "
## [245] "rg30p: rg30p - rg30p_aerr = y_rg30p(1) + y_rg30p(2) * zgap30 + y_rg30p(3) * d8095 + y_r
## [246] "rg5: rg5 - rg5_aerr = (( (.01*rg5e + 1)^.5 - 1) * 200) "
## [247] "rg5e: rg5e - rg5e_aerr = zrff5 + rg5p "
## [248] "rg5p: rg5p - rg5p_aerr = y_rg5p(1) + y_rg5p(2) * zgap05 + y_rg5p(3) * (rg5p(-1) - y_rg5
## [249] "rgfint: rgfint - rgfint_aerr = (y_rgfint(1) * rgfint(-1) + (1-y_rgfint(1))*rgw(-1))*(gf
## [250] "rgw: rgw - rgw_aerr = y_rgw(1) * rtb + y_rgw(2) * rg5 + y_rgw(3) * rg10 + y_rgw(4) * rg
## [251] "rme: d( rme, 0, 1 ) - rme_aerr = y_rme(1) + y_rme(2) * d( rg10e, 0, 1) + y_rme(3) * d87
## [252] "rpd: rpd - rpd_aerr = 0.5*(7.2 + (1-trfcim)*(rg5e + rbbbe- rg10e) - zpib5) + 0.5*req "
## [253] "rrffe: rrffe - rrffe_aerr = rffe - ( picxfe + picxfe(-1) + picxfe(-2) + picxfe(-3)) / 4
## [254] "rrmet: rrmet - rrmet_aerr = y_rrmet(1) * rrmet(-1) + y_rrmet(2) * (rme-zpi10) "
## [255] "rrtr: rrtr - rrtr_aerr = y_rrtr(1) * rrtr(-1) + y_rrtr(2) * rrffe "
## [256] "rspnia: rspnia - rspnia_aerr = 100 * yhsn / ydn "
## [257] "rstar: rstar - rstar_aerr = rstar(-1) + y_rstar(1) * ((rrffe-rstar(-1))*drstar) "
## [258] "rtb: rtb - rtb_aerr = 36000/90 * (1-(.01*rtbe+1)^(-90/365)) "
## [259] "rtbe: rtbe - rtbe_aerr = y_rtbe(1) + ( y_rtbe(2) * rtbe(-1) + y_rtbe(3) * rtbe(-2)) + (
## [260] "rtinv: rtinv - rtinv_aerr = (.01*rpd - .01*hgpkir) * ( ( pxp*pkir + pxp(-1)*pkir(-1)) /
## [261] "rtpd: rtpd - rtpd_aerr = (.01*rpd + jrpd - .01*hgpdr) * ((1-.01*tapdt-trfcim*(1-tapddp*
## [262] "rtpi: rtpi - rtpi_aerr = (.01*rpd + jrpi - .01*hgpir) * ( ( pxp*ppir + pxp(-1)*ppir(-1)
```

```
## [263] "rtps: rtps - rtps_aerr = (@recode(((.01*rpd + jrps - .01*hgppsr) * ((1-trfcin
## [264] "rtr: rtr - rtr_aerr = rrtr + ptr "
## [265] "tapdd: tapdd - tapdd_aerr = .5 * d2003 + .5 * d2003 * (2.0 / (2.0 + .01 * tap
## [266] "tapsda: tapsda - tapsda_aerr = (1-tapsad)*(1-exp(-0.01*(rpd+zpib5)*tapssl))/
## [267] "tfcin: tfcin - tfcin_aerr = trfci * ynicpn "
## [268] "tfibn: tfibn - tfibn_aerr = trfib * ecnian "
## [269] "tfpn: tfpn - tfpn_aerr = trfp * (ypn - gftn - gstn) "
## [270] "tfsin: tfsin - tfsin_aerr = trfsi * yniln "
## [271] "trfci: trfci - trfci_aerr = y_trfci(1) + y_trfci(2) * trfci(-1) + y_trfci(3)
## [272] "trfp: trfp - trfp_aerr = y_trfp(1) * trfpt + ( y_trfp(2) * (trfp(-1)-trfpt(-1
## [273] "trfpt: trfpt - trfpt_aerr = dfpex * trfptx + dfpdbt * ( trfpt(-1) + y_trfpt(1
## [274] "trsci: trsci - trsci_aerr = y_trsci(1) * trsci(-1) + ( y_trsci(2) * trscit +
## [275] "trsib: trsib - trsib_aerr = y_trsib(1) * trsib(-1) + ( y_trsib(2) * trsibt +
## [276] "trsp: trsp - trsp_aerr = y_trsp(1) * trsp(-1) + ( y_trsp(2) * trspt + y_trsp
## [277] "trspt: trspt - trspt_aerr = dfpex * trsptx + dfpdbt * ( trspt(-1) + y_trspt(1
## [278] "trssi: trssi - trssi_aerr = ( y_trssi(1) * trssi(-1) + y_trssi(2) * trssi(-2)
## [279] "tryh: tryh - tryh_aerr = (tfpn+tspn)/(yhln+yhptn) "
## [280] "tscin: tscin - tscin_aerr = trsci * ynicpn "
## [281] "tsibn: tsibn - tsibn_aerr = trsib * ecnian "
## [282] "tspn: tspn - tspn_aerr = trsp * (ypn - gftn - gstn) "
## [283] "tssin: tssin - tssin_aerr = trssi * yniln "
## [284] "uces: d( log(uces), 0, 1 ) - uces_aerr = y_uces(1) * log(uces(-1)) + y_uces(1
## [285] "ucfs: d( log(ucfs), 0, 1 ) - ucfs_aerr = y_ucfs(1) * log(ucfs(-1)) + y_ucfs(1
## [286] "uqpct: log(uqpct) - uqpct_aerr = y_uqpct(1) + log(uqpct(-1)) + huqpct "
## [287] "uxbt: log(uxbt) - uxbt_aerr = y_uxbt(1) + log(uxbt(-1)) + .0025*huxb "
## [288] "veo: log(veo) - veo_aerr = log(pxb/pceng) "
## [289] "veoa: log(veoa) - veoa_aerr = y_veoa(1) * log(veoa(-1)) + y_veoa(2) * log(veo
## [290] "vpd: vpd - vpd_aerr = uvpd*(pkpdr/ppdr)/rtpd "
## [291] "vpi: vpi - vpi_aerr = uvpi/rtpi "
## [292] "vps: vps - vps_aerr = uvps/rtps "
## [293] "wdnfcn: d( log(wdnfcn), 0, 1) - wdnfcn_aerr = y_wdnfcn(1) * log(wdnfcn(-1)/(y
## [294] "wpo: wpo - wpo_aerr = wpon/(.01*pcnia) "
## [295] "wpon: wpon - wpon_aerr = wpon(-1)*exp( (1-((phouse(-1)*kh(-1)/116)/wpon(-1)))
## [296] "wps: wps - wps_aerr = wpsn/(.01*pcnia) "
## [297] "wpsn: log(wpsn) - wpsn_aerr = log((ynicpn-tfcin-tscin)*.5) - .25 * (req-zdivg
## [298] "xb: xb - xb_aerr = xbn/ (pxb/100) "
## [299] "xbn: xbn - xbn_aerr = pxb/100*xbo + xgdpn -xgdo*pgdp/100 "
## [300] "xbo: log(xbo) - xbo_aerr = log(xbt) + y_xbo(1) * xgap2/100 "
## [301] "xbt: log(xbt) - xbt_aerr = log(xb) + (log(xgpot/xg) - .5 *(.035*empn/(.01*pce
## [302] "xeng: xeng - xeng_aerr = uxeng * xgpot "
## [303] "xfs: log(xfs) - xfs_aerr = log(xfs(-1)) + .5*( (ecnian/xfsn + ecnian(-1)/xfsn
## [304] "xfsn: xfsn - xfsn_aerr = xgdpn - ein "
## [305] "xg: log(xg) - xg_aerr = log(xg(-1)) + (1 - .5*(.035*empn/(.01*pceng*ceng) +
## [306] "xgap: xgap - xgap_aerr = 100*log(xgo/xgpot) "
## [307] "xgap2: xgap2 - xgap2_aerr = 100 * log(xgdo/xgdpt) "
```

```
## [308] "xgde: log(xgde) - xgde_aerr = log(xgde(-1)) + .5*( (xgdpn/xgden + xgdpn(-1)/xgden(-1))
## [309] "xgden: xgden - xgden_aerr = xgdpn + emn - exn "
## [310] "xgdi: xgdi - xgdi_aerr = xgdo*mei "
## [311] "xgdin: xgdin - xgdin_aerr = xgdi *(pgdp/100) "
## [312] "xgdo: xgdo - xgdo_aerr = xgdp/mep "
## [313] "xgdp: xgdp - xgdp_aerr = xgdp(-1) * @sqrt( ( (xfsn(-1)/xgdpn(-1)) * (xfs/xfs(-1)) + (.0
## [314] "xgdpn: xgdpn - xgdpn_aerr = xpn + ein - emn + egfln + egsln "
## [315] "xgdpt: log(xgdpt) - xgdpt_aerr = log(xbt) + log(uxbt) "
## [316] "xgdptn: xgdptn - xgdptn_aerr = .01*pgdp*xgdpt "
## [317] "xgn: xgn - xgn_aerr = xbn + empn "
## [318] "xgo: log(xgo) - xgo_aerr = log(xgpot) + y_xgo(1) * xgap2/100 "
## [319] "xgpot: log(xgpot) - xgpot_aerr = (y_xgpot(1) * (log(leppot) + log(qlww) + log(lqualt))
## [320] "xp: log(xp) - xp_aerr = log(xp(-1)) + .5 * (ecnian/xpn + ecnian(-1)/xpn(-1)) * d(log(ec
## [321] "xpn: xpn - xpn_aerr = .01 * pxp * xp "
## [322] "ycsn: ycsn - ycsn_aerr = ynicpn - tfcin - tscin - ftcin - ynidn + jccacn "
## [323] "ydn: ydn - ydn_aerr = uyd * (ypn - tfpn - tspn) "
## [324] "ygfsn: ygfsn - ygfsn_aerr = gfsrpn + jygfgn + jygfen "
## [325] "ygssn: ygssn - ygssn_aerr = gssrpn + jygsgn + jygsen "
## [326] "yh: yh - yh_aerr = yhl + yht + yhp "
## [327] "yhgap: yhgap - yhgap_aerr = 100*(yhshr/zyhst-1) "
## [328] "yhibn: d( log(yhibn), 0, 1 ) - yhibn_aerr = y_yhibn(1) * ( picxfe/1600 + picxfe(-1)/160
## [329] "yhin: yhin - yhin_aerr = uyhi * (yniin + gfintn + gsintn + yhibn) "
## [330] "yhl: yhl - yhl_aerr = (1-tryh)*yhln/(.01*pcnia) "
## [331] "yhln: yhln - yhln_aerr = uyhln * (yniln - tfsin - tssin) "
## [332] "yhp: yhp - yhp_aerr = ((1-tryh)*yhptn+yhpntn)/(.01*pcnia) "
## [333] "yhpcd: log(yhpcd) - yhpcd_aerr = log(y_yhpcd(1)) + log(kcd(-1)) "
## [334] "yhpgap: yhpgap - yhpgap_aerr = 100*(yhpshr/zyhpst-1) "
## [335] "yhpntn: yhpntn - yhpntn_aerr = .01*pcnia*pcdr*yhpcd - yhibn + ynicpn - tfcin - tscin -
## [336] "yhpshr: yhpshr - yhpshr_aerr = yhp/yh "
## [337] "yhptn: yhptn - yhptn_aerr = uyhptn*(ynisen+yhin+ynidn) "
## [338] "yhshr: yhshr - yhshr_aerr = yh/xgdp "
## [339] "yhsn: yhsn - yhsn_aerr = yhln + yhtn + yhptn - tfpn - tspn - ecnian - yhibn + uyhsn * x
## [340] "yht: yht - yht_aerr = yhtn/(.01*pcnia) "
## [341] "yhtgap: yhtgap - yhtgap_aerr = 100*(yhtshr/zyhtst-1) "
## [342] "yhtn: yhtn - yhtn_aerr = uyhtn*(gftn+gstn) "
## [343] "yhtshr: yhtshr - yhtshr_aerr = yht/yh "
## [344] "ykin: ykin - ykin_aerr = .01*rtinv*pxb* (ki + ki(-1)) /2 "
## [345] "ykpdn: ykpdn - ykpdn_aerr = .01*rtpd*pxb* ( kpd + kpd(-1)) /2 "
## [346] "ykpsn: ykpsn - ykpsn_aerr = .01*rtps*pxb* ( kps + kps(-1)) /2 "
## [347] "ynicpn: ynicpn - ynicpn_aerr = uynicp * (@recode((ynin-yniln-yniin-ynisen-tfibn-tsibn+g
## [348] "ynidn: d( log((ynidn-ymsdn)/pxb), 0, 1 ) - ynidn_aerr = y_ynidn(1) * log(qynidn(-1)/(yn
## [349] "yniin: yniin/(ynin(-1)-yniln(-1)) - yniin_aerr = y_yniin(1) + y_yniin(2) * (yniin(-1)/(
## [350] "yniln: yniln - yniln_aerr = 0.01 * uyl * (pl*lhp + pgfl*egfl + pgsl*egsl) "
## [351] "ynin: ynin - ynin_aerr = uyni*(xgdin+fynin-jccan) "
## [352] "ynisen: ynisen - ynisen_aerr = uysen*xbn "
```

```
## [353] "ypn: ypn - ypn_aerr = uyp * (yhln + yhtn + yhptn) "
## [354] "zdivgr: zdivgr-zdivgr_aerr = y_zdivgr(1) + ( y_zdivgr(2) * picnia + y_zdivgr
## [355] "zecd: zecd-zecd_aerr = ( y_zecd(1) * picnia(-1) + y_zecd(2) * picnia(-2) + y_
## [356] "zeco: zeco-zeco_aerr = ( y_zeco(1) * picnia(-1) + y_zeco(2) * picnia(-2) + y_
## [357] "zeh: zeh-zeh_aerr = ( y_zeh(1) * picnia(-1) + y_zeh(2) * picnia(-2) + y_zeh(3
## [358] "zgap05: zgap05-zgap05_aerr = y_zgap05(1) + ( y_zgap05(2) * picnia + y_zgap05
## [359] "zgap10: zgap10-zgap10_aerr = y_zgap10(1) + ( y_zgap10(2) * picnia + y_zgap10
## [360] "zgap30: zgap30-zgap30_aerr = y_zgap30(1) + ( y_zgap30(2) * picnia + y_zgap30
## [361] "zgapc2: zgapc2-zgapc2_aerr = ( y_zgapc2(1) * picnia(-1) + y_zgapc2(2) * picni
## [362] "zlhp: zlhp-zlhp_aerr = ( y_zlhp(1) * picnia(-1) + y_zlhp(2) * picnia(-2) + y_
## [363] "zpi10: zpi10-zpi10_aerr = ( y_zpi10(1) * picnia(-1) + y_zpi10(2) * picnia(-2)
## [364] "zpi10f: zpi10f-zpi10f_aerr = zpi10 "
## [365] "zpi5: zpi5-zpi5_aerr = ( y_zpi5(1) * picnia(-1) + y_zpi5(2) * picnia(-2) + y_
## [366] "zpib5: zpib5-zpib5_aerr = y_zpib5(1) + ( y_zpib5(2) * picnia(-1) + y_zpib5(3)
## [367] "zpic30: zpic30-zpic30_aerr = y_zpic30(1) + ( y_zpic30(2) * picnia + y_zpic30
## [368] "zpic58: zpic58-zpic58_aerr = ( y_zpic58(1) * picnia + y_zpic58(2) * picnia(-1
## [369] "zpicxfe: zpicxfe-zpicxfe_aerr = ( y_zpicxfe(1) * picxfe(-1) + y_zpicxfe(2) *
## [370] "zpieci: zpieci-zpieci_aerr = ( y_zpieci(1) * picxfe(-1) + y_zpieci(2) * picxf
## [371] "zrff10: zrff10-zrff10_aerr = y_zrff10(1) + ( y_zrff10(2) * picnia + y_zrff10
## [372] "zrff30: zrff30-zrff30_aerr = y_zrff30(1) + ( y_zrff30(2) * picnia + y_zrff30
## [373] "zrff5: zrff5-zrff5_aerr = y_zrff5(1) + ( y_zrff5(2) * picnia + y_zrff5(3) * p
## [374] "zvpd: zvpd-zvpd_aerr = y_zvpd(1) + ( y_zvpd(2) * picnia(-1) + y_zvpd(3) * pic
## [375] "zvpi: zvpi-zvpi_aerr = ( y_zvpi(1) * picnia(-1) + y_zvpi(2) * picnia(-2) + y_
## [376] "zvps: zvps-zvps_aerr = ( y_zvps(1) * picnia(-1) + y_zvps(2) * picnia(-2) + y_
## [377] "zxbd: zxbd-zxbd_aerr = y_zxbd(1) + ( y_zxbd(2) * picnia(-1) + y_zxbd(3) * pic
## [378] "zxbi: zxbi-zxbi_aerr = ( y_zxbi(1) * picnia(-1) + y_zxbi(2) * picnia(-2) + y_
## [379] "zxbs: zxbs-zxbs_aerr = ( y_zxbs(1) * picnia(-1) + y_zxbs(2) * picnia(-2) + y_
## [380] "zyh: log(zyh) - zyh_aerr = ( y_zyh(1) * picnia + y_zyh(2) * picnia(-1) + y_zy
## [381] "zyhp: log(zyhp) - zyhp_aerr = ( y_zyhp(1) * picnia + y_zyhp(2) * picnia(-1) +
## [382] "zyhpst: zyhpst-zyhpst_aerr = zyhpst(-1) + y_zyhpst(1)*(yhpshr-zyhpst(-1)) "
## [383] "zyhst: zyhst-zyhst_aerr = zyhst(-1) + y_zyhst(1)*(yhshr-zyhst(-1)) "
## [384] "zyht: log(zyht) - zyht_aerr = ( y_zyht(1) * picnia + y_zyht(2) * picnia(-1) +
## [385] "zyhtst: zyhtst - zyhtst_aerr = zyhtst(-1) + y_zyhtst(1)*(yhtshr-zyhtst(-1)) "
## [386] "zynid: zynid - zynid_aerr = y_zynid(1) + ( y_zynid(2) * picnia(-1) + y_zynid
```

0.4 Chunks

0.5 Index