Package 'R4MFCL'

June 26, 2013

Title R funcions for MFCL

Version 0.1

Author Simon Hoyle
Description miscellaneous funtions for dealing with input to and output from MFCL
Maintainer <simonh@spc.int></simonh@spc.int>
License GPL-2
Date 2011-05-25
R topics documented:
add.catch.frq
add.cpue.frq
add.flag
alb.clean.lfdata
alb.initial_clean_2008
alb.initial_clean_2009
alb.initial_clean_2011
alb.pagocatch_2009
carry.effort.frq
change.fishflag
change.flag
change.negflag
change.size.frq
change_data
check.eff.devs
check_flag_value
clean.lfdata
compare.ce.frq
compare.frq
compare.size.frq
compare.size.frq2
compare_par_flags
condor.go
condor.go2
Convert.frq.ver6

2

	11
e	16
1	17
	17
	18
	18
	19
= 6	19
1	20
- e	20
	21
1	21
e e	22
	22
get.outcomes	
get.outcomes.alb	23
get.outcomes.test	24
get.weight.output	24
initial_clean_2008	25
	25
	25
	26
	26
11	 27
e i	 27
	 28
	28
	29
1	29
	29
	2) 30
	30
	30 31
<u>.</u>	31
	31 32
L Company of the Comp	32
L L	33
1	33
	34
r	34 25
$oldsymbol{\mathcal{E}}$	35 25
..	35
rr J	36
rr	36
LL	37
rr	37
	38
plot.recruitment.combined	38
plot_cpue_cv_frq	39
read.catchrep	39
read.ests	40
read.fit	40
	41
icau.iiq	+1

read.impact	
read.ini	
read.par	
read.rep	
read.tag	
read.tags.JP	
read.var	
read_nmd.frq	44
read_nmd.par	45
reconstruct.frq.ce	45
reconstruct.frq.ce2008	46
reconstruct.frq.size	46
region_single_frq	47
region_single_ini	
region_single_tag	
regroup_fishery_grps.doitall	
rename.fisheries.doitall	
rename.fisheries.frq	
rename.fisheries.tag	
rename.fishery.grps.doitall	
retro.frq	
retro.tag	
rm_fisheries.doitall	
rm_fisheries.frq	
rm_fisheries.tag	
rm_fishflag	
rm_flag.doitall	
run.profile	
seas.flag	
seas.frq	
seas.tag	
setup.cpue	
setup.effcreep	
setup.growth	
setup.growth.offsets	
setup.idphcatch	
setup.lupiicateii	
•	
setup.LFwt	
setup.M	
setup.pscatch	
setup.startyr	
setup.steepness	
setup.timesplit	
sort.frq	
start_year.frq	63
steepness.doit	
summarise.size.frq.bet	
tag_grps_rm	
timesplit.doitall	
timesplit.frq	
timesplit.tag	
varfromstr	66

4 add.cpue.frq

```
      write.frq
      67

      write.ini
      67

      write.par
      68

      write.tag
      68

      write_nmd.frq
      69

      write_nmd.par
      69

Index
```

add.catch.frq.Rd

Usage

```
add.catch.frq(frq, filepath, fshries)
```

Arguments

```
frq
filepath
fshries
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

add.cpue.frq add.cpue.frq.Rd

Usage

```
add.cpue.frq(CPUE.file = "P:/yft/2009/Data Preparation/CPUE/indices/yft_JPstd_R1
```

Arguments

```
CPUE.file
data
fishery
add.cv
```

```
##--- Should be DIRECTLY executable !! ----
##-- => Define data, use random,
##--or do help(data=index) for the standard data sets.
```

add.flag 5

```
add.flag add.flag.Rd
```

Usage

```
add.flag(doitall, flagtype, flagnum, newval, phase)
```

Arguments

```
doitall
flagtype
flagnum
newval
phase
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
alb.clean.lfdata alb.clean.lfdata.Rd
```

Usage

```
alb.clean.lfdata(infrq)
```

Arguments

infrq

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

alb.initial_clean_2009

```
alb.initial_clean_2008

alb.initial_clean_2008.Rd
```

Usage

```
alb.initial_clean_2008(frq.obj)
```

Arguments

```
frq.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
alb.initial_clean_2009

alb.initial_clean_2009.Rd
```

Usage

```
alb.initial_clean_2009(frq.obj)
```

Arguments

```
frq.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

alb.initial_clean_2011 7

Usage

```
alb.initial_clean_2011(frq.obj)
```

Arguments

```
frq.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
alb.pagocatch_2009 alb.pagocatch_2009.Rd
```

Usage

```
alb.pagocatch_2009(frq.obj)
```

Arguments

```
frq.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

8 change.fishflag

```
carry.effort.frq carry.effort.frq.Rd
```

Usage

```
carry.effort.frq(data = out.data, fishery = 1, last = 2008)
```

Arguments

```
data
fishery
last
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
change.fishflag change.fishflag.Rd
```

Usage

```
change.fishflag(a, fisheries, flagnum, newvals)
```

Arguments

```
a
fisheries
flagnum
newvals
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

change.flag 9

```
change.flag change.flag.Rd
```

Usage

```
change.flag(doitall, flagtype, flagnum, newval)
```

Arguments

```
doitall
flagtype
flagnum
newval
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

change.negflag

change.negflag.Rd

Usage

```
change.negflag(doitall, flagtype, flagnum, newval)
```

Arguments

```
doitall
flagtype
flagnum
newval
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

10 change_data

```
change.size.frq change.size.frq.Rd
```

Usage

```
change.size.frq(ver = 6, data = data, FISH = 1, LF.FILE = "P:/yft/2009/Data Prep
```

Arguments

```
ver
data
FISH
LF.FILE
WT.FILE
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

change_data

change_data.Rd

Usage

```
change_data(obj, searchtext, xlines, newline)
```

Arguments

```
obj
searchtext
xlines
newline
```

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

check.eff.devs 11

```
check.eff.devs. check.eff.devs.Rd
```

Usage

```
check.eff.devs(parfile, repfile, frqfile, parlim = 5.9)
```

Arguments

```
parfile
repfile
frqfile
parlim
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

Usage

```
check_flag_value(parname, flagtype, flagnums, fishery = NA, flaglist = T)
```

Arguments

```
parname
flagtype
flagnums
fishery
flaglist
```

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

12 compare.ce.frq

```
clean.lfdata
```

clean.lfdata.Rd

Usage

```
clean.lfdata(infrq)
```

Arguments

infrq

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
compare.ce.frq
```

compare.ce.frq.Rd

Usage

```
compare.ce.frq(file1, file2, fm, plotname, fdesc = "")
```

Arguments

```
file1
file2
fm
plotname
fdesc
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

compare.frq 13

```
compare.frq. compare.frq.Rd
```

Usage

```
compare.frq(file1, file2, fm = "all", plotname, fdesc = "")
```

Arguments

```
file1
file2
fm
plotname
fdesc
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
compare.size.frq compare.size.frq.Rd
```

Usage

```
compare.size.frq(frq1, frq2, fishery = 5)
```

Arguments

```
frq1
frq2
fishery
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

14 compare_par_flags

```
\verb|compare.size.frq2| compare.size.frq2.Rd|
```

Usage

```
compare.size.frq2(frq1, frq2, fishery = 5, wt = T)
```

Arguments

```
frq1
frq2
fishery
wt
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
compare_par_flags compare_par_flags.Rd
```

Usage

```
compare_par_flags(par1, par2, flaglist = T)
```

Arguments

```
par1
par2
flaglist
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

condor.go 15

condor.go condor.go.Rd

Usage

```
condor.go(run.dir, frq.obj, tag.obj, doitall.obj, ini.obj, sub.obj = suball, spe
```

Arguments

```
run.dir
frq.obj
tag.obj
doitall.obj
ini.obj
sub.obj
species
condor_f
par.obj
run_now
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

condor.go2

condor.go2.Rd

Usage

```
condor.go2(run.dir, frq.obj, tag.obj, doitall.obj, ini.obj, sub.obj = suball, sp
```

Arguments

```
run.dir
frq.obj
tag.obj
doitall.obj
ini.obj
sub.obj
species
condor_f
par.obj
run_now
```

16 create.missing.ce

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
Convert.frq.ver6 Convert.frq.ver6.Rd
```

Usage

```
Convert.frq.ver6(a)
```

Arguments

а

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
create.missing.ce create.missing.ce.Rd
```

Usage

```
create.missing.ce(data = data, year = 2008)
```

Arguments

```
data
year
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

create.ver6.frq

```
create.ver6.frq create.ver6.frq.Rd
```

Usage

```
create.ver6.frq(frq = base.frq)
```

Arguments

frq

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
crit.fishery.summary crit.fishery.summary.Rd
```

Usage

```
crit.fishery.summary(crit)
```

Arguments

crit

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

18 datfromstr

crit.summary

crit.summary.Rd

Usage

```
crit.summary(crit, years)
```

Arguments

```
crit
years
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

datfromstr

datfromstr.Rd

Usage

```
datfromstr(datstring)
```

Arguments

```
datstring
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

do.critical.calcs

```
do.critical.calcs do.critical.calcs.Rd
```

Usage

```
do.critical.calcs(repfile = "P:/yft/2007/BaseYFT/yftfinal2007.rep", ests = "P:/y
```

Arguments

```
repfile ests
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
doit.rm_flag
```

doit.rm_flag.Rd

Usage

```
doit.rm_flag(a, flagtype, flag, value)
```

Arguments

```
flagtype flag value
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

20 fix_growth

effortcreep

effortcreep.Rd

Usage

```
effortcreep(frq.obj, fisheries, creep)
```

Arguments

```
frq.obj
fisheries
creep
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

fix_growth

fix_growth.Rd

Usage

```
fix_growth(a)
```

Arguments

а

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

frq.change.nint 21

```
frq.change.nint frq.change.nint.Rd
```

Usage

```
frq.change.nint(in.frq, add.lfint, add.wfint)
```

Arguments

```
in.frq
add.lfint
add.wfint
```

Examples

```
##--- Should be DIRECTLY executable !! ---
##-- => Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
frq.remove.size.or.weight.data

frq.remove.size.or.weight.data.Rd
```

Usage

```
frq.remove.size.or.weight.data(data = test.data, exclude = exclude, size = T, we
```

Arguments

```
data
exclude
size
weight
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

22 get.length.output

```
get.critical.age get.critical.age.Rd
```

Usage

```
get.critical.age(data = Base.rep)
```

Arguments

data

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
get.length.output get.length.output.Rd
```

Usage

```
get.length.output(REGION = 1, DIR = "P:/yft/2009/Data Preparation/size data/")
```

Arguments

REGION

DIR

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

get.outcomes 23

```
get.outcomes get.outcomes.Rd
```

Usage

```
get.outcomes(file.rep, file.par, nofish = T)
```

Arguments

```
file.rep
file.par
nofish
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
get.outcomes.alb get.outcomes.alb.Rd
```

Usage

```
get.outcomes.alb(file.rep, file.par, catch.rep, nofish = T, SBcurr_1yr = F)
```

Arguments

```
file.rep
file.par
catch.rep
nofish
SBcurr_lyr
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

24 get.weight.output

```
get.outcomes.test get.outcomes.test.Rd
```

Usage

```
get.outcomes.test(file.rep, file.par, catch.rep, nofish = T)
```

Arguments

```
file.rep
file.par
catch.rep
nofish
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
get.weight.output get.weight.output.Rd
```

Usage

```
get.weight.output(REGION = 1, DIR = "P:/yft/2009/Data Preparation/size data/")
```

Arguments

REGION DIR

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

initial_clean_2008 25

```
initial_clean_2008 initial_clean_2008.Rd
```

Usage

```
initial_clean_2008(frq.obj)
```

Arguments

```
frq.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

labels_store

labels_store.Rd

Usage

```
labels_store(labelfile = basecase.labels)
```

Arguments

labelfile

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

load.LFdata

load.LFdata.Rd

Usage

```
load.LFdata(species = "ALB", gear = "L")
```

Arguments

```
species
gear
```

26 map_all_pacific

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
make.projection.betyft.frq

make.projection.betyft.frq.Rd
```

Usage

```
make.projection.betyft.frq(frq.in = base.frq, fish = 1:24, years = 10)
```

Arguments

```
frq.in
fish
years
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
map_all_pacific map_all_pacific.Rd
```

Usage

```
map\_all\_pacific(plot\_title = "", lims = c(100, 300, -45, 45), eezfile = "L:/alb/"
```

Arguments

```
plot_title
lims
eezfile
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

merge.frq 27

merge.frq

merge.frq.Rd

Usage

```
merge.frq(frq.obj, oldf, newf, mergelf = FALSE)
```

Arguments

```
frq.obj
oldf
newf
mergelf
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

merge.tag

merge.tag.Rd

Usage

```
merge.tag(tag.obj, oldf, newf)
```

Arguments

```
tag.obj
oldf
newf
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

28 NZtrollglm

```
merge_tag_objs. merge_tag_objs.Rd
```

Usage

```
merge_tag_objs(obj1, obj2, relgrps)
```

Arguments

obj1 obj2

relgrps

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

NZtrollglm

NZ troll glm. Rd

Usage

```
NZtrollglm(frq.obj, fishery, coef.file)
```

Arguments

```
frq.obj
fishery
coef.file
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

oldsetup 29

oldsetup

oldsetup.Rd

Usage

```
oldsetup(oldbasedir)
```

Arguments

oldbasedir

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
pack.fisheries.frq pack.fisheries.frq.Rd
```

Usage

```
pack.fisheries.frq(frq.obj)
```

Arguments

frq.obj

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

pagocatch

pagocatch.Rd

Usage

```
pagocatch(frq.obj)
```

Arguments

```
frq.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

30 plot.biomass

```
plot.base.comparison

plot.base.comparison.Rd
```

Usage

```
plot.base.comparison(baseres, labs)
```

Arguments

```
baseres
labs
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.biomass

plot.biomass.Rd

Usage

```
plot.biomass(plotdir = "H:/rmfcl/test/figs/", plotrep = test, varfile = NULL, ty
```

Arguments

```
plotdir
plotrep
varfile
type
plotname
plottype
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.biomass.combined 31

```
plot.biomass.combined plot.biomass.combined.Rd
```

Usage

```
plot.biomass.combined(plotdir = "H:/rmfcl/test/figs/", plotrep = test, varfile =
```

Arguments

```
plotdir
plotrep
varfile
type
plotname
plottype
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.F.time plot.F.time.Rd
```

Usage

```
plot.F.time(plotdir = "H:/rmfcl/test/figs/", plotrep = "C:/assessments/alb/2008/
```

Arguments

```
plotdir
plotrep
inifile
plotname
plottype
COL
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

32 plot.Kobe

```
plot.fishery.impact.r

plot.fishery.impact.r.Rd
```

Usage

```
plot.fishery.impact.r(plotdir = "H:/rmfcl/test/figs/", type = "Total", plotrep =
```

Arguments

```
plotdir
type
plotrep
impnames
plotname
plottype
COL
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.Kobe

plot.Kobe.Rd

Usage

```
plot.Kobe(plotdir = "S:/OFP Publications/Tuna Fishery Assessment Report/2007/Fig
```

Arguments

```
plotdir
plotrep
type
plotname
plottype
COL
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.Kobe.template.bw

33

```
plot.Kobe.template.bw plot.Kobe.template.bw.Rd
```

Usage

```
plot.Kobe.template.bw(Type)
```

Arguments

Type

Examples

```
##--- Should be DIRECTLY executable !! ---
##-- => Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
{\it plot.} {\it Kobe.template.col.} \\ {\it plot.Kobe.template.col.Rd}
```

Usage

```
plot.Kobe.template.col(Type)
```

Arguments

Type

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

34 plot.nofishing

```
plot.mfcl.betyft09 plot.mfcl.betyft09.Rd
```

Usage

```
plot.mfcl.betyft09(lims = c(100, 260, -45, 45))
```

Arguments

lims

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.nofishing

plot.nofishing.Rd

Usage

```
plot.nofishing(plotdir = "H:/rmfcl/test/figs/", plotrep = testq0, type = "SSB",
```

Arguments

```
plotdir
plotrep
type
plotname
plottype
COL
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.nofishing.combined plot.nofishing.combined.Rd
```

Usage

```
plot.nofishing.combined(plotdir = "H:/rmfcl/test/figs/", plotrep = testq0, type
```

Arguments

```
plotdir
plotrep
type
plotname
plottype
COL
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.pacific.alb plot.pacific.alb.Rd
```

Usage

```
plot.pacific.alb(plot_title = "", eez_dir = "I:/assessments/Pop dy modeling/MFCI
```

Arguments

```
plot_title
eez_dir
plot_eez
```

```
##--- Should be DIRECTLY executable !! ----
##-- => Define data, use random,
##--or do help(data=index) for the standard data sets.
```

36 plot.pacific.species

```
plot.pacific.skj plot.pacific.skj.Rd
```

Usage

```
plot.pacific.skj(plot_title = "")
```

Arguments

```
plot_title
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

Usage

```
plot.pacific.species(plot_title = "", uselims = NA, add.WCPFC = F, add.EPO = F,
```

Arguments

```
plot_title
uselims
add.WCPFC
add.EPO
sp
add.EEZ
eez_file
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot.pacific.WCPFC 37

```
\verb"plot.pacific.WCPFC" \textit{plot.pacific.WCPFC.Rd}
```

Usage

```
plot.pacific.WCPFC(plot_title = "", lims = c(100, 260, -45, 45))
```

Arguments

```
plot_title
lims
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.pacific.yft plot.pacific.yft.Rd
```

Usage

```
plot.pacific.yft(plot_title = "", lims = c(100, 260, -45, 45), add.WCPFC = F)
```

Arguments

```
plot_title
lims
add.WCPFC
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.recruitment plot.recruitment.Rd
```

Usage

```
plot.recruitment(plotdir = "H:/rmfcl/test/figs/", plotrep = test, varfile = NULI
```

Arguments

```
plotdir
plotrep
varfile
plotname
plottype
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
plot.recruitment.combined \\ plot.recruitment.combined.Rd
```

Usage

```
plot.recruitment.combined(plotdir = "H:/rmfcl/test/figs/", plotrep = test, varfi
```

Arguments

```
plotdir
plotrep
varfile
plotname
plottype
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

plot_cpue_cv_frq 39

```
plot_cpue_cv_frq plot_cpue_cv_frq.Rd
```

Usage

```
plot_cpue_cv_frq(frq, parf, fisheries)
```

Arguments

```
frq
parf
fisheries
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.catchrep

read.catchrep.Rd

Usage

```
read.catchrep(catchrep.file)
```

Arguments

```
catchrep.file
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

40 read.fit

read.ests

read.ests.Rd

Usage

```
read.ests(rep.obj, ests = "C:/assessments/alb/2008/6_area/28.splitgr3/ests.rep",
```

Arguments

```
rep.obj
ests
x
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.fit

read.fit.Rd

Usage

```
read.fit(fit.file)
```

Arguments

```
fit.file
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.frq 41

```
read.frq. read.frq.Rd
```

Usage

```
read.frq(frq.file, frq.title = "", ntop = 0, fishdefs = NA)
```

Arguments

```
frq.file
frq.title
ntop
fishdefs
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.impact

read.impact.Rd

Usage

```
read.impact(impdir = "H:/rmfcl/test/", impnames = c("ll", "psassoc", "psunassoc"
```

Arguments

```
impdir
impnames
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.rep

read.ini

read.ini.Rd

Usage

```
read.ini(ini.file)
```

Arguments

ini.file

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.par

read.par.Rd

Usage

```
read.par(par.file)
```

Arguments

```
par.file
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.rep

read.rep.Rd

Usage

```
read.rep(rep.file)
```

Arguments

```
rep.file
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.tag 43

read.tag

read.tag.Rd

Usage

```
read.tag(tagfile)
```

Arguments

tagfile

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read.tags.JP

read.tags.JP.Rd

Usage

```
read.tags.JP(tagfile, reg_areas, fisheries)
```

Arguments

```
tagfile
reg_areas
fisheries
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read_nmd.frq

read.var

read.var.Rd

Usage

```
read.var(var.file)
```

Arguments

```
var.file
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read_nmd.frq

read_nmd.frq.Rd

Usage

```
read_nmd.frq(frq.file, frq.title = "", ntop = 0, fishdefs = NA)
```

Arguments

```
frq.file
frq.title
ntop
fishdefs
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

read_nmd.par 45

```
read_nmd.par
read_nmd.par.Rd
```

Usage

```
read_nmd.par(par.file)
```

Arguments

```
par.file
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
reconstruct.frq.ce reconstruct.frq.ce.Rd
```

Usage

```
reconstruct.frq.ce(CPUE.file = "X:/yft/2009/Data Preparation/CPUE/indices/yft_JF
```

Arguments

```
CPUE.file data fishery
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

46 reconstruct.frq.size

```
reconstruct.frq.ce2008

reconstruct.frq.ce2008.Rd
```

Usage

```
reconstruct.frq.ce2008(data = out.data, fishery = 1)
```

Arguments

data fishery

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
reconstruct.frq.size reconstruct.frq.size.Rd
```

Usage

```
reconstruct.frq.size(data = data, FISH = 1, LF.FILE = "P:/yft/2009/Data Preparat
```

Arguments

```
data
FISH
LF.FILE
WT.FILE
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

region_single_frq 47

```
region_single_frq region_single_frq.Rd
```

Usage

```
region_single_frq(frq, region)
```

Arguments

```
frq region
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

Usage

```
region_single_ini(ini)
```

Arguments

ini

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
region_single_tag region_single_tag.Rd
```

Usage

```
region_single_tag(tag, region, keepfish)
```

Arguments

```
tag
region
keepfish
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
regroup_fishery_grps.doitall
regroup_fishery_grps.doitall.Rd
```

Usage

```
regroup_fishery_grps.doitall(doitall, f, flag, newgrps)
```

Arguments

```
doitall
f
flag
newgrps
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

rename.fisheries.doitall 49

```
{\it rename.fisheries.doitall.} Rd
```

Usage

```
rename.fisheries.doitall(doitall, oldfs, newfs)
```

Arguments

```
doitall
oldfs
newfs
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
{\it rename.fisheries.frq.} Rd {\it rename.fisheries.frq.} Rd
```

Usage

```
rename.fisheries.frq(frq.obj, oldfish, newfish)
```

Arguments

```
frq.obj
oldfish
newfish
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
rename.fisheries.tag

rename.fisheries.tag.Rd
```

Usage

```
rename.fisheries.tag(tag.obj, oldfish, newfish)
```

Arguments

```
tag.obj
oldfish
newfish
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
rename.fishery.grps.doitall

rename.fishery.grps.doitall.Rd
```

Usage

```
rename.fishery.grps.doitall(doitall, oldfs, newfs, flag, keep = T, newgrps = c(0
```

Arguments

```
doitall
oldfs
newfs
flag
keep
newgrps
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

retro.frq 51

```
retro.frq
```

retro.frq.Rd

Usage

```
retro.frq(frq.obj, retro.tag.obj = NA)
```

Arguments

```
frq.obj
retro.tag.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

retro.tag

retro.tag.Rd

Usage

```
retro.tag(tag.obj, yr)
```

Arguments

```
tag.obj
yr
```

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

52 rm_fisheries.frq

```
rm_fisheries.doitall

rm_fisheries.doitall.Rd
```

Usage

```
rm_fisheries.doitall(a, rmfisheries)
```

Arguments

```
a rmfisheries
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
rm_fisheries.frq rm_fisheries.frq.Rd
```

Usage

```
rm_fisheries.frq(frq.obj, fishery)
```

Arguments

```
frq.obj
fishery
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

rm_fisheries.tag 53

Usage

```
rm_fisheries.tag(tag.obj, fisheries)
```

Arguments

```
tag.obj
fisheries
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

rm_fishflag

rm_fishflag.Rd

Usage

```
rm_fishflag(doitall, flag)
```

Arguments

```
doitall flag
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

run.profile

Usage

```
rm_flag.doitall(a, flagtype, flag, value)
```

Arguments

```
a
flagtype
flag
value
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

run.profile

run.profile.Rd

Usage

```
run.profile(rundir, rungrp, startpar = NA, ptype = "Fmult", target, nsteps = 300
```

Arguments

```
rundir
rungrp
startpar
ptype
target
nsteps
penalty
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

seas.flag 55

```
seas.flag seas.flag.Rd
```

Usage

```
seas.flag(a, fishery, flagnum, seasf.list)
```

Arguments

```
fishery
flagnum
seasf.list
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

seas.frq

seas.frq.Rd

Usage

```
seas.frq(frq.obj, seas.fish)
```

Arguments

```
frq.obj
seas.fish
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

56 setup.cpue

```
seas.tag
```

seas.tag.Rd

Usage

```
seas.tag(tag.obj, fishlist)
```

Arguments

```
tag.obj
fishlist
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.cpue

setup.cpue.Rd

Usage

```
setup.cpue(rungrp, sourcedir, cpue, spp)
```

Arguments

```
rungrp
sourcedir
cpue
spp
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.effcreep 57

```
setup.effcreep.Rd
```

Usage

```
setup.effcreep(rungrp, creeprate)
```

Arguments

```
rungrp
creeprate
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.growth

setup.growth.Rd

Usage

```
setup.growth(rungrp, VBopt)
```

Arguments

```
rungrp
VBopt
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

58 setup.idphcatch

```
setup.growth.offsets

setup.growth.offsets.Rd
```

Usage

```
setup.growth.offsets(rungrp, ageclasses, penwt, phase, tog)
```

Arguments

```
rungrp
ageclasses
penwt
phase
tog
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

```
setup.idphcatch setup.idphcatch.Rd
```

Usage

```
setup.idphcatch(rungrp, sourcedir, idph, spp)
```

Arguments

```
rungrp
sourcedir
idph
spp
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.lensel 59

```
setup.lensel
```

setup.lensel.Rd

Usage

```
setup.lensel(rungrp, fisheries, tog)
```

Arguments

```
rungrp
fisheries
tog
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.LFwt

setup.LFwt.Rd

Usage

```
setup.LFwt(rungrp, newLFwt)
```

Arguments

```
rungrp
newLFwt
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.pscatch

setup.M

setup.M.Rd

Usage

```
setup.M(rungrp, newM)
```

Arguments

```
rungrp
newM
```

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.pscatch

setup.pscatch.Rd

Usage

```
setup.pscatch(rungrp, sourcedir, PScatch, spp)
```

Arguments

```
rungrp
sourcedir
PScatch
spp
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

61 setup.startyr

```
setup.startyr
                      setup.startyr.Rd
```

Usage

```
setup.startyr(rungrp, newstartyr)
```

Arguments

```
rungrp
newstartyr
```

Examples

```
##---- Should be DIRECTLY executable !! ----
\#\#-- ==>  Define data, use random,
##--or do help(data=index) for the standard data sets.
```

setup.steepness setup.steepness.Rd

Usage

```
setup.steepness(rungrp, newsteep)
```

Arguments

```
rungrp
newsteep
```

```
##---- Should be DIRECTLY executable !! ----
\#\#-- ==>  Define data, use random,
##--or do help(data=index) for the standard data sets.
```

62 sort.frq

```
setup.timesplit setup.timesplit.Rd
```

Usage

```
setup.timesplit(rungrp, splitx, storefish)
```

Arguments

```
rungrp
splitx
storefish
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

sort.frq

sort.frq.Rd

Usage

```
sort.frq(frq.obj)
```

Arguments

```
frq.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

start_year.frq 63

```
start_year.frq start_year.frq.Rd
```

Usage

```
start_year.frq(frq.obj, start_yr, halfyr = F)
```

Arguments

```
frq.obj
start_yr
halfyr
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

steepness.doit

steepness.doit.Rd

Usage

```
steepness.doit(doitall, new.steepness, add_header = T, gap = 2)
```

Arguments

```
doitall
new.steepness
add_header
gap
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

tag_grps_rm

```
summarise.size.frq.bet
summarise.size.frq.bet.Rd
```

Usage

```
summarise.size.frq.bet(frq1, fishery = 5)
```

Arguments

```
frq1
fishery
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

tag_grps_rm

 $tag_grps_rm.Rd$

Usage

```
tag_grps_rm(tag.obj, keep)
```

Arguments

```
tag.obj
keep
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

timesplit.doitall 65

```
timesplit.doitall timesplit.doitall.Rd
```

Usage

```
timesplit.doitall(doitall, fishsplit, qsplit = T)
```

Arguments

```
doitall
fishsplit
qsplit
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

timesplit.frq

timesplit.frq.Rd

Usage

```
timesplit.frq(frq.obj, divyrs, div.fish)
```

Arguments

```
frq.obj
divyrs
div.fish
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

66 varfromstr

```
timesplit.tag
```

timesplit.tag.Rd

Usage

```
timesplit.tag(tag.obj, fishsplit)
```

Arguments

```
tag.obj
fishsplit
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

varfromstr

varfromstr.Rd

Usage

```
varfromstr(datstring, cols = c(2:3))
```

Arguments

```
datstring
cols
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

write.frq 67

write.frq

write.frq.Rd

Usage

```
write.frq(new.frq, frq.obj)
```

Arguments

```
new.frq
frq.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

write.ini

write.ini.Rd

Usage

```
write.ini(ini.file, ini.obj)
```

Arguments

```
ini.file
ini.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

68 write.tag

write.par

write.par.Rd

Usage

```
write.par(par.file, par.obj)
```

Arguments

```
par.file
par.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

write.tag

write.tag.Rd

Usage

```
write.tag(tagfile, tag.obj)
```

Arguments

```
tagfile
tag.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

write_nmd.frq 69

```
write_nmd.frq.Rd
```

Usage

```
write_nmd.frq(new.frq, frq.obj)
```

Arguments

```
new.frq
frq.obj
```

Examples

```
##--- Should be DIRECTLY executable !! ---
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

write_nmd.par

write_nmd.par.Rd

Usage

```
write_nmd.par(par.file, par.obj)
```

Arguments

```
par.file
par.obj
```

```
##--- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.
```

Index

*Topic \textasciitildekwd1	initial_clean_2008,25
add.catch.frq,4	labels_store, 25
add.cpue.frq,4	load.LFdata, 25
add.flag,5	<pre>make.projection.betyft.frq,</pre>
alb.clean.lfdata,5	26
alb.initial_clean_2008,6	map_all_pacific, 26
alb.initial_clean_2009,6	merge.frq,27
alb.initial_clean_2011,7	merge.tag, 27
carry.effort.frq,8	merge_tag_objs,28
change.fishflag, 8	NZtrollglm, 28
change.flag, 9	oldsetup, 29
change.negflag,9	pack.fisheries.frq,29
change.size.frq,10	pagocatch, 29
change_data, 10	${ t plot.base.comparison, 30}$
check.eff.devs, 11	plot.biomass, 30
check_flag_value, 11	plot.biomass.combined, 31
clean.lfdata,12	plot.F.time, 31
compare.ce.frq, 12	plot.fishery.impact.r,32
compare.frq, 13	plot.Kobe, 32
compare.size.frq,13	plot.Kobe.template.bw, 33
compare.size.frq2,14	plot.Kobe.template.col, 33
compare_par_flags, 14	plot.mfcl.betyft09,34
condor.go,15	plot.nofishing, 34
condor.go2,15	plot.nofishing.combined, 35
Convert.frq.ver6, 16	plot.pacific.alb,35
create.missing.ce,16	plot.pacific.skj,36
create.ver6.frq,17	plot.pacific.species, 36
crit.fishery.summary,17	plot.pacific.WCPFC, 37
crit.summary, 18	plot.pacific.yft, 37
datfromstr, 18	plot.recruitment, 38
do.critical.calcs, 19	plot.recruitment.combined, 38
doit.rm_flag, 19	plot_cpue_cv_frq,39
effortcreep, 20	read.catchrep, 39
fix_growth, 20	read.ests, 40
frq.change.nint,21	read.fit,40
frq.remove.size.or.weight.data,	read.frq,41
21	read.impact, 41
get.critical.age, 22	read.ini,42
get.length.output, 22	read.par, 42
get.outcomes, 23	read.rep, 42
get.outcomes.alb, 23	read.tag, 43
get.outcomes.test, 24	read.tags.JP,43
get.weight.output,24	read.var,44

read_nmd.frq,44	write_nmd.par,69
read_nmd.par,45	*Topic \textasciitildekwd2
reconstruct.frq.ce,45	add.catch.frq,4
reconstruct.frq.ce2008,46	add.cpue.frq,4
reconstruct.frq.size,46	add.flag,5
region_single_frq,47	alb.clean.lfdata,5
region_single_ini,47	alb.initial_clean_2008,6
region_single_tag, 48	alb.initial_clean_2009,6
regroup_fishery_grps.doitall,	alb.initial_clean_2011,7
48	carry.effort.frq,8
rename.fisheries.doitall,49	change.fishflag, 8
rename.fisheries.frq,49	change.flag,9
rename.fisheries.tag, 50	change.negflag,9
rename.fishery.grps.doitall,	change.size.frq, 10
50	change_data, 10
retro.frq,51	check.eff.devs, 11
retro.tag, 51	check_flag_value, 11
rm_fisheries.doitall,52	clean.lfdata, 12
rm_fisheries.frq,52	compare.ce.frq, 12
rm_fisheries.tag, 53	compare.frq, 13
rm_fishflag, 53	compare.size.frq, 13
rm_flag.doitall,54	compare.size.frq2, 14
run.profile, 54	compare_par_flags, 14
seas.flag,55	condor.go, 15
seas.frq,55	condor.go2, 15
seas.tag, 56	Convert.frq.ver6, 16
setup.cpue, 56	create.missing.ce, 16
setup.effcreep, 57	create.wer6.frq, 17
setup.growth, 57	crit.fishery.summary, 17
setup.growth.offsets, 58	crit.summary, 18
setup.idphcatch, 58	datfromstr, 18
setup.lensel, 59	do.critical.calcs, 19
setup.LFwt, 59	doit.rm_flag, 19
setup.M, 60	effortcreep, 20
setup.pscatch, 60	fix_growth, 20
setup.startyr, 61	frq.change.nint, 21
setup.steepness, 61	frq.remove.size.or.weight.data,
setup.steephess, 01 setup.timesplit, 62	21
sort.frq, 62	get.critical.age, 22
start_year.frq, 63	get.length.output, 22
steepness.doit, 63	get.length.output, 22
summarise.size.frq.bet, 64	get.outcomes, 23
tag_grps_rm, 64	get.outcomes.arb, 23
	_
timesplit.doitall,65 timesplit.frq,65	get.weight.output, 24 initial_clean_2008, 25
timesplit.tag,66	labels_store, 25
varfromstr, 66	load.LFdata, 25
write.frq,67	make.projection.betyft.frq,
write.ini,67	26
write.par,68	map_all_pacific, 26
write.tag, 68	merge.frq,27
write_nmd.frq,69	merge.tag,27

merge_tag_objs, 28	rm_fisheries.doitall, 52
NZtrollglm, 28	$rm_fisheries.frq, 52$
oldsetup, 29	rm_fisheries.tag,53
pack.fisheries.frq, 29	rm_fishflag,53
pagocatch, 29	rm_flag.doitall,54
plot.base.comparison, 30	run.profile, 54
plot.biomass, 30	seas.flag, 55
plot.biomass.combined, 31	seas.frq,55
plot.F.time, 31	seas.tag, 56
plot.fishery.impact.r, 32	setup.cpue, 56
plot.Kobe, 32	setup.effcreep, 57
plot.Kobe, 32 plot.Kobe.template.bw, 33	setup.growth, 57
plot.Kobe.template.bw, 33	setup.growth.offsets, 58
	setup.idphcatch, 58
plot.mfcl.betyft09,34	setup.lensel, 59
plot.nofishing, 34	setup.LFwt, 59
plot.nofishing.combined, 35	setup.M, 60
plot.pacific.alb, 35	setup.pscatch, 60
plot.pacific.skj,36	setup.startyr, 61
plot.pacific.species, 36	
plot.pacific.WCPFC, 37	setup.steepness, 61
plot.pacific.yft,37	setup.timesplit, 62
plot.recruitment, 38	sort.frq, 62
plot.recruitment.combined, 38	start_year.frq,63
plot_cpue_cv_frq,39	steepness.doit,63
read.catchrep,39	summarise.size.frq.bet,64
read.ests,40	tag_grps_rm,64
read.fit, 40	timesplit.doitall,65
read.frq,41	timesplit.frq,65
read.impact,41	timesplit.tag,66
read.ini,42	varfromstr,66
read.par,42	write.frq,67
read.rep, 42	write.ini,67
read.tag, 43	write.par,68
read.tags.JP,43	write.tag,68
read.var,44	write_nmd.frq,69
read_nmd.frq,44	write_nmd.par,69
read_nmd.par,45	
reconstruct.frq.ce, 45	add.catch.frq,4
reconstruct.frq.ce2008,46	add.cpue.frq,4
reconstruct.frq.size, 46	add.flag,5
	alb.clean.lfdata,5
region_single_frq,47	alb.initial_clean_2008, 6
region_single_ini,47	alb.initial_clean_2009, 6
region_single_tag,48	alb.initial_clean_2011,7
regroup_fishery_grps.doitall,	alb.pagocatch_2009,7
48	_
rename.fisheries.doitall,49	carry.effort.frq,8
rename.fisheries.frq,49	change.fishflag,8
rename.fisheries.tag, 50	change.flag,9
rename.fishery.grps.doitall,	change.negflag, 9
50	change.size.frq,10
retro.frq,51	change_data, 10
retro.tag,51	check.eff.devs, 11

check_flag_value, 11	plot.fishery.impact.r,32
clean.lfdata, 12	plot.Kobe, 32
compare.ce.frq,12	plot.Kobe.template.bw,33
compare.frq, 13	plot.Kobe.template.col, 33
compare.size.frq,13	plot.mfcl.betyft09,34
compare.size.frq2,14	plot.nofishing,34
compare_par_flags,14	plot.nofishing.combined, 35
condor.go, 15	plot.pacific.alb, 35
condor.go2, 15	plot.pacific.skj,36
Convert.frq.ver6,16	plot.pacific.species, 36
create.missing.ce, 16	plot.pacific.WCPFC, 37
create.ver6.frq,17	plot.pacific.yft, 37
crit.fishery.summary,17	plot.recruitment,38
crit.summary, 18	plot.recruitment.combined, 38
	plot_cpue_cv_frq, 39
datfromstr, 18	
do.critical.calcs, 19	read.catchrep, 39
doit.rm_flag,19	read.ests, 40
	read.fit,40
effortcreep, 20	read.frg,41
	read.impact,41
fix_growth, 20	read.ini,42
frq.change.nint,21	read.par, 42
<pre>frq.remove.size.or.weight.data,</pre>	read.rep, 42
21	read.tag, 43
	read.tags.JP,43
get.critical.age, 22	read.var,44
get.length.output,22	read_nmd.frq,44
get.outcomes, 23	read_nmd.par, 45
get.outcomes.alb, 23	
get.outcomes.test, 24	reconstruct.frq.ce, 45
get.weight.output,24	reconstruct.frq.ce2008,46
	reconstruct.frq.size, 46
initial_clean_2008,25	region_single_frq,47
	region_single_ini,47
labels_store, 25	region_single_tag,48
load.LFdata, 25	regroup_fishery_grps.doitall,48
	rename.fisheries.doitall,49
make.projection.betyft.frq,26	rename.fisheries.frq,49
map_all_pacific,26	rename.fisheries.tag,50
merge.frq,27	rename.fishery.grps.doitall,50
merge.tag,27	retro.frq,51
merge_tag_objs,28	retro.tag,51
3 — 3— 3 /	rm_fisheries.doitall, 52
NZtrollglm, 28	$rm_fisheries.frq, 52$
	rm_fisheries.tag,53
oldsetup, 29	rm_fishflag,53
	rm_flag.doitall,54
pack.fisheries.frq,29	run.profile,54
pagocatch, 29	
plot.base.comparison,30	seas.flag,55
plot.biomass, 30	seas.frq,55
plot.biomass.combined, 31	seas.tag,56
plot.F.time, 31	setup.cpue,56

```
setup.effcreep, 57
setup.growth, 57
setup.growth.offsets, 58
setup.idphcatch, 58
setup.lensel, 59
setup.LFwt, 59
setup.M, 60
setup.pscatch, 60
setup.startyr, 61
setup.steepness, 61
setup.timesplit, 62
sort.frq, 62
start\_year.frq, 63
steepness.doit, 63
summarise.size.frq.bet,64
tag_grps_rm, 64
timesplit.doitall,65
timesplit.frq,65
timesplit.tag, 66
varfromstr, 66
write.frq,67
write.ini,67
write.par, 68
write.tag, 68
write_nmd.frq,69
write_nmd.par, 69
```