```
gens fit class.gens fit
                                                        bg: float
                                                        devf
                                                        elim: NoneType
                                                        gamma
                                                        k
                                                        leastsq: bool
                                                        optbgpeakratio
                                                        out
                                                        quiet: bool
                                                        showplot: bool
                                                        tf
                                                        x df
                                                        x tf
                                                        y_df
                                                        y_tf
                                          afteroptimize(out, s_sq, variables, figname)
                                          check generated samples(x, data)
                                          check spectra()
                                          checkdata()
                                          convlore(f, gamma, x)
                                          convloreorg(f, gamma, x)
                                          correction()
                                          decorrection()
                                          fun lore(x, gamma)
                                          generate data(idevf, itf, check, rebin)
                                          get data(infile)
                                          get hdata(infile)
                                          get_icorrdata(icorrfile)
                                          get idata(infile)
                                          get_sdata(infile)
                                          icorr()
                                          interpolate()
                                          kde hist(kvariables, hvariables)
                                          kde hist sub(tf, devf, kde, variables)
                                          limit(x, y, mergin)
                                          limit2(x, y, elim)
                                          multii(idevf, itf)
                                          optimize(variables, figname)
                                          preprocess(doicorr)
                                          preprocessh(doicorr)
                                          preprocessnoi(doicorr)
                                          preprocesss(doicorr)
                                          rebin generated samples(x, data, num, shift)
                                          reconstruct(elim, check, idevf, itf)
                                          res(coeffs, x, d, t)
                                          res icorr(coeffs, x, t)
                                          save generated data(x, data, savefile)
                                          save result()
                                          testconv()
     gens class fort mpi.gens
                                           qens fit class hist noidata.runhistnoidata
                                                       alpha
                                                       devf
                                                       elim
                                                       elimw
                                                       leastsq: bool
                                                       numcycle: int
                                                       outall: ndarray, list
                                                       outfile
                                                       tf
     selected spectra : ndarray
                                                       X
                                                       yd
                                           check out(cyidx, out)
                                           correction(x, yd, yt)
                                           cycle(variables)
                                           decorrection(x, yd, yt)
                                           generate_data(idata)
     xvec_real : ndarray
                                           get xmlyd(variables)
                                           loadfile()
                                           modify_out(cyidx, out)
                                           optimize(x, yd, yt, variables)
                                           output()
calc_ssvkernel_f90(WinFuncNo)
                                           plot distribution(binwidth1, binwidth2)
                                           plot distribution single(binwidth1, show)
                                           preprocess()
run ssvkernel(num, isnovariablebw)
                                           reconstruct(x, yd, out)
save output(output file)
                                           res(coeffs, x, d, t)
save outputs(output file)
                                           savefile()
                            4
                        qens fit class kde.runkdenoidata
                                 M: int
                                 WinFunc: str
                                 alpha
                                 de
                                 devf
                                 dt
                                 elim
                                 elimw
                                 leastsq: bool
                                 numcycle: int
                                 outall: list
                                 outfile
                                 rank
                                 selected energy
                                 selected spectra
                                 tf
                                 tin real: ndarray
                                 winparam: int
                                 y: tuple
                                 y_hist
                                 yd
            Gauss(x, w)
```

M: int

datadir

dataset

de

tin

add shift() add shift de()

get_xvec()

select spectra()

baloon_estimator()

kde baloon(x, y)

run ssvkernel notused()

kde(x, y, M, winparam, num, WinFunc, isnovariablebw)

gens balloon resample class. Sqens balloon resamples

Nb: int bg: float comm elim: list

etl

cycle()

hist()

get_xmlyd()

preprocess()

plotter()

WinFunc: str

figname: str

odata: bool qsel: bool

quiet: bool save file

selected energy

showplot: bool

tin_real : ndarray

winparam: int

xvec: ndarray

y: tuple

calc_sskernel f90()

gammas : ndarray ishist: bool isnovariablebw: bool kyios kyis : list kyos kys : list leastsq : bool num: int odata : bool orgfiles : list orgmodifier: str outall: list, ndarray pklfile prefixes: list quiet : bool rank rsfiles: list rsmodifier: str runNos: list size spectrab variables: list X y:str CI of intensities() CI of intensities io() CalcBandW(orgfile, inb) DefineFiles() DoQf(inb) DoQfandKDE(inb) DoQfio(inb) Gauss(x, w)balloon(ky, sy) check idata() eachrunno(fidx, inb) eachrunno io(fidx, inb) getrsspectra(rsfile, inb) io(kyo, kyi) load pkl(pklfile) run() run eachkde() run eachkde_io() run io() qens balloon resample classm2. Sqens balloon resamples M: int

rsfiles: list rsmodifier: str runNos: list size variables: list winparam: int DefineFiles() $qens_balloon_resample_classm2_class.Sqbr$ outfile

Nb: int

comm elim : list

qidx

rank

quiet : bool

WinFunc: str

ishist: bool

ispltchk: bool leastsq: bool num: int orgfiles : list orgmodifier: str prefixes: list

gammas : ndarray

isnovariablebw: bool