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 M: int WinFunc: str alpha datadir devf dataset elim elimw de figname: str leastsq: bool odata: bool qsel: bool numcycle: int quiet: bool outall: list save file outfile selected energy tf selected spectra : ndarray X yd showplot: bool check_out(cyidx, _out) tin correction(x, yd, yt) tin_real : ndarray cycle(variables) winparam: int decorrection(x, yd, yt) xvec: ndarray generate_data(idata) xvec_real : ndarray get xmlyd(variables) y: tuple loadfile() modify_out(cyidx, out) add shift() add shift de() optimize(x, yd, yt, variables) calc_sskernel f90() output() calc_ssvkernel_f90(WinFuncNo) plot distribution(binwidth1, binwidth2) get_xvec() plot distribution single(binwidth1, show) plotter() preprocess() run ssvkernel(num, isnovariablebw) reconstruct(x, yd, out) save output(output file) res(coeffs, x, d, t) save outputs(output file) savefile() select spectra() 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) baloon_estimator() cycle() get_xmlyd()

gens fit class.gens fit

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 classm.qens balloon resamples M: int Nb: int WinFunc: str comm elim : list gammas: ndarray ishist: bool isnovariablebw: bool ispltchk: bool leastsq: bool num: int orgmodifier: str prefixes: list qidx quiet : bool rank rsmodifier: str runNos : list size spectrab variables: list winparam: int y: str CalcBandW(orgfile, inb) getrsspectra(rsfile, inb) qens balloon resample org classm class.qens org classm

outall: ndarray, list

combine_qs()
get out()

run for mqs()

outfile: str

qidx qsize

hist()

kde baloon(x, y)

run ssvkernel notused()

preprocess()

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

gens balloon resample class. Sqens balloon resamples

gammas : ndarray

isnovariablebw: bool

ishist: bool

Nb: int bg: float comm elim: list

etl

kyios
kyis: list
kyos
kys: list
leastsq: bool
num: int
odata: bool
orgfiles: list
orgmodifier: str
outall: ndarray, list

pklfile

prefixes : list quiet : bool