

LA_tools Python Package

module	function	description	input	output
tagreport (pandas/numpy depends)		methods for pulling, validating & exporting tag reports		
	cap_query	query to pull tag report from PE	PR_rev (string)	report (dataframe)
	check_PRdates	main function to run cap/trans date checks	PR_rev (str) report (df)	cap and/or trans errors (dataframes)
look for != (H or S) instead of == E	→ cap_check	checks for missing start/stop dates, bad dates, est tags, zero tags, tag gaps	PR_rev report	error flags (dataframe)
	→ trans_check			
	get_report	main function to get tag report	PR_rev (str)	report (dataframe)
	export_report	writes report to .csv in directory	PR_rev directory (strings)	prints: -file written -overwrite prompt -error
db_query (cx_Oracle depends)		query DB CHANGE BASED ON PRODUCTION PLATFORM...	...BLOB STORAGE, DB FROM SERVER...	
	oracle_pe	queries PE w/ credentials	query (str)	db results (list)
	oracle_lpss	TO DO?		
	oracle_alps	TO DO?		
offer_summary (pandas/numpy)		calculate statistics based on meter forecasts at PR level		
	offer_summary	TO SPLIT INTO FUNCTIONS		
ADD CP DATES ALL MARKETS	offer APPLY FUNCS	calc annual use, meter peaks, CP, % diff bt tag & CP/actual	ch 3 data (df)	offer sum (dataframe)

module	function	description	input	output
check > act_tag_diff OR > cap_tag_diff	peaktag_check	subset where tag, diff > min_tag, abs(min_diff)	ch3 data, offer sum (df) min tag, min diff (float)	problem meters (df)
	iter_plot	plots multiple graphs	ch3 data (df)	plots
forecast		tools to allow more direct access to EWX forecasting tools		
	parse_engie	parses predeal json to csv	filename (str)	ingestion payload (csv) chl, sca, caps, hb
	parse_ewx	parses response json to csv	filename (str)	response payload (csv) cap tags
validation	periodic_zero	check if zeros are periodic or not	idr (df) margin, threshold (float)	idr w/ pz col attach'd (df)
	int_gap_check	check for gaps in idr	idr	idr w/ time, val diff's & gap after index flag
	variance_val	check for spikes & dips	idr (df) n_sd (float) window (int)	roll mean & sd, center use, roll cen mean & sd, variation, spike & dip flags
	dst_check	checks DST	idr	dst flag
	fix_interval	fixes nonhourly intervals	idr	checks time diff (15 min, need all)
estimation	dst_fix	adds or subs hour	idr	idr
	interp	interpolates vals (based on overall error flag)	idr	idr
adjust to end of yr	gen_year	generates next year dates	idr, num days (int)	idr

module	function	description	input	output
	timeshift	shifts data by weekday, hour, mon	idr, num_yrs (int)	timeshift (df)
split into val and ts funcs (take parameters for val?)	forecast_main	uses json/csv to validate & timeshift	idr (df), read_dir, write_dir (str)	writes timeshift, val csv return ts
NEPOOL IDR Drop		automation workaround for dropping NEPOOL IDR data		
	emailsrape	scrapes GSERNA box for utility portal info	past days (int)	usernames, passwords, accounts, NGRID or not (json)
TIMEOUT ERROR IN CHECKING BOXES & DOWNLOADING STAGE	webscrape	uses login to download IDR for <= 5 meters at a time	email scrape json	raw meter data (csv)
	raw_split	splits raw data into individual raw IDR files	raw meter data (csv)	ind raw IDR (csv)
	data_drop	filters raw to clean ch 1 data	ind raw IDR (csv)	IDR (csv) writes to local, possible to write to DATARESPONSE

TO DO: convert code to work w/ classes,
expand database query functionality?,
expand offer summary across markets,
small improvements (listed above)
create setup.py, readme & dependencies