**Directory Listing (./steenburgh-group10/mewessler/)**

./era5/

era5\_orog.nc Sample orography from the era5

./era5/iso/

Archived WR subset of ERA5 pressure level data (1979-2019)

From the NCAR RDA

./era5/sfc/

Archived WR subset of ERA5 surface level data (1979-2019)

From the NCAR RDA

./era5/profiles/

Extracted profiles for the ERA5 for gridpoints used in the study

./gfs/

Archived WR subset of pressure level and surface GFS data (Jan 2015 – May 2020)

From the NCAR RDA

**Directory Listing (./steenburgh-group10/mewessler/snow-liquid-ratio-final/)**

./baxter/

All things baxter climo related…

./era5/

Scripts for subsetting ERA5 and extracting profiles at points

./gfs/

Aggregate GFS NetCDF file

./notebooks/

All notebook files listed below

./observations

All observations, raw, QC’ed, and parsed. See Snow\_Data\_Processing.docx for more information on subdirectories and how to parse raw data

./output/

Saved SVR and other models, supporting files like Z-scoring and variable listings

./allsta\_t\_r\_spd\_swe\_train1224/

Saved latest iteration and most successful SVR model – import with np.load()

Files are appropriately named with the model saved to file, including scaler and key files that support the SVR model file.

./scripts/

Executable python scripts described in below file list

./SLR\_CODE\_DOC.docx (this document)

**Code**

Anaconda/Python Environment: All scripts are written for Python 3.8.3. Environments were managed using a custom Anaconda3 environment, NOT a CHPC module (<https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html#activating-an-environment>). You can configure these on your own and add whatever packages and versions of Python that are necessary. A full package listing can be found at the end of this document.

As these scripts have been consolidated from their original directory structure to clean them up and for ease of use in the future. Paths within each script should be checked and updated to make sure they point to the correct working directories, temp directories, and archives. Files are organized into /scripts (.py) and /notebooks (.ipynb).

|  |  |
| --- | --- |
| **Script** | **Function** |
| **SLRfuncs\_UUtah2019\_WPC.ipynb** | **Contains a complete writeup with working examples that was provided to WPC to implement the legacy SLR algorithm from the weather.utah.edu page into their systems. Will be very helpful if a future student is to rewrite the GFS SLR code (or apply to literally any other model).** |
| ingest\_config.py | Contains header and timezone info for generic\_data\_ingest.py. Must be appropriately named and set up for each raw data file. Some preprocessing of each data file in excel is necessary to make this run smoothly. See existing data files for examples. |
| generic\_data\_ingest.py | Automated script to ingest all existing data files in a given directory. Uses glob to identify all \*.csv files in the observations directory then runs through each with a metadata entry in the ingest config. Exports a Pandas binary (\*.pd) that can be easily read in later scripts. |
| aggregate\_observations.ipynb | Hasn’t been scripted as not much of a need exists, can just do in notebook on the fly. Collects all existing Pandas binary files (\*.pd) into an aggregate Pandas dataframe. Could be exported to NetCDF or a SQL style database easily. This would help for sharing the database upon publication. |
| ingest\_cln.ipynb | **Example notebook (not needed in bottom-up processing)** used to develop the generic data ingest script. Designed specifically and only for the Alta Collins data (currently – can be expanded). Has components that were not fully implemented into generic\_data\_ingest.py which include comparing against data from the Synoptic Data Timeseries/Precipitation API (in the form of saved CSVs). These comparisons are useful and should be implemented fully as a sanity check on observations if possible. |
| Ingest\_cssl.ipynb | An attempt to use Pandas to parse through the varied/complex spreadsheets provided from CSSL. Good example of how to approach this problem (would apply to other CDOT and UDOT spreadsheets) but needs refining. Note there is no correction for the offset SWE/Snow observation timing yet. |
| /era5/scripts/subset\_era5.py | Space saving measure. Start with ERA5 grids obtained from the NCAR RDA in full (using globus file transfer is fastest, Copernicus API is terribly slow for pulling large amounts of data across time dimension). Outputs grids trimmed to western region -130:-100;30:50 |
| /era5/scripts/extract\_profile.py | Extracts ERA5 profiles variable by variable, year by year. Computationally more efficient to do this and then aggregate these into compiled profiles after |
| /era5/scripts/aggregate\_profile.py | Aggregates the year/variable data extracted above into more useful NetCDF files by variable/year |
| /era5/scripts/extract\_agg\_profile.sh [lat] [lon] [startyear] [endyear] | Parent script to call extract and aggregate functions to create the ERA5 profiles |
| pair\_era5profiles\_isobaric.ipynb | Combines the ERA5 profile for a given site with the observation data. Produces histograms of SLR over time as well as other useful metrics (monthly SLR, standard deviation change over time, etc). Outputs a pandas binary file that is used in later scripts to train and test the SLR model. |
| pair\_era5profiles.ipynb | Same as above, with a key modification to allow the transfer of isobaric coordinate to numbered level above ground (can use model surface, or actual elevation; max, min, or mean geopotential height of the surface). Noted that this method produced degraded performance in the model, chose to continue using isobaric method. |
| pair\_era5profiles.py | Same as above (for numbered levels AGL) but scripted for automation. Not used currently but provides a framework for streamlining the process in the future. |
| gfs\_profiles.ipynb | Compiles all singular GFS grib data into a single NetCDF file for ease of use. Takes a significant amount of time to run, needs to have wind speed and direction added in for completeness (this was done on the fly in another script and the current NetCDF compilation has these variables). Suggest only appending new data to this NetCDF rather than rebuilding from scratch to save time but a full re-run will be fine as well. |
| train\_model\_isobaric.ipynb | The ‘meat and potatoes.’ Ingests the compiled binary data files, runs through linear, elastic, MLP, and SVR models with stats, outputs the model as a .npy binary to be read in the GFS test scripts, as well as the Z-scaling information and variable list. Set up to be fairly adjustable and to allow quick tests. Sometimes (rarely) the SVR training hangs, just restart the kernel and re-run the script. Reduce the number of grid-search items or CPUs used in multithreading to reduce chances of crashing. Lots of plots to visualize performance of the training/testing. This is the latest iteration of the code and is designed to work with data compiled in isobaric coordinates. |
| train\_model\_isobaric\_GFS.ipynb | Same as above but uses the GFS (2015 forward) to train the model for a MOS style approach and not a perfect prog approach. Performance is decent but a longer dataset is needed. Inclusion of the 19-20 and 20-21 winter could make this a viable and very appealing solution. Does NOT have all of the features as the ERA training script but would be easy to transfer these in. |
| linear\_correlations.ipynb | Produces linear correlation coefficient plots in the vertical dimension for each site. Used to determine differences between sites, variables, identify location of maximum correlation, and behavior of sub-surface levels against SLR. Easily modifiable for other purposes. |
| event\_pairs.ipynb | Produces scatterplots of GFS vs ERA5 for event pairs (only precipitating events) at one/all sites. Used to compare behavior of variables in GFS against the ERA5 to determine if there are significant issues present that warrant exclusion of a variable or model level |
| era\_verify.ipynb | Produces scatterplots and 2d histograms (density plots/heatmaps) of forecast vs observed SLR for some/all sites (as configured). The SLR models are saved out from the training script and read into this script. Need to update directories of these throughout. |

**How to implement the SVR:**

**See era\_verify.ipynb for guidance!**

Clear-cut example of how to import the gridded GFS at an init time, then use the svr\_etc() function as a guideline for how to process the input variables into an SLR using the model.predict() function. Will have to broadcast this function over whatever set of x,y gridpoints and t times that are needed. Use python multiprocessing to streamline this.

**Observations**

|  |  |
| --- | --- |
| raw/CAIC | Misc station set some as few as 1-3 years, other 10+. Lots of manual entry in excel that needs to be parsed. Data taken from various CDOT sources |
| raw/Mammoth | Main lodge, snow stake, AM/PM for 12-hour data if needed. Data taken by ski patrol |
| raw/Snoqualmie | Julie Lundquist’s dataset, high quality, lots of metadata incl. snow level and rain-on-snow. Mostly 24h but some 1h data from recent years. ~30y record ending 2015 |
| raw/CSSL | Messy dataset needs further parsing than what has already been done. Discrepancy between time of snow obs and time of SWE obs that needs to be accounted for somehow in calculating SLR |
| raw/CUES | Energy balance sites. Mostly automated data https://vesr.nrs.ucsb.edu/about/facilities/snarl/cues-snow-science-laboratory |
| raw/NUTAH | Most data from UT Avalanche Center, acquired through NWS SLC. Other data from UDOT avalanche program. Some mountain COOP stations available, including Brighton and Alta. Look into SnoCourse for more data |
| raw/BNSF | BNSF Railroad avalanche safety program in Flatheads area of MT. Data courtesy the Flathead Avalanche Center. Nice manual dataset with consistent methods and results across 3 sites. Mix of observation times and intervals needs to be addressed |
| raw/NYSM | NY State Mesonet snow/SWE data. Unknown usefulness/quality, yet to be analyzed |
| raw/other | Misc collection of other sites that don’t fit the above locations, one-offs, or unknown locations |

**Points of Contact**

|  |  |
| --- | --- |
| NWS SLC | Allister Martinelli |
| UDOT Cottonwoods | Mark Sauer  Damian Jackson  Steven Clark |
| Utah Avalanche Center | Drew Hardesty |
| BNSF Avalanche Program | Adam Clark  Blase Reardon |
| CAIC | Mike Cooperstein (Coop)  Ethan Greene |
| Central Sierra Snow Lab | Tony Crocker [crockeraf@gmail.com](mailto:crockeraf@gmail.com) |
| Mammoth/CUES | Ned Bair [nbair@eri.ucsb.edu](mailto:nbair@eri.ucsb.edu) |
| Snoqualmie Pass | Julie Lundquist |
| Sawtooth Avalanche Center  USFS Ketchum Idaho | Scott Savage [scott@sawtoothavalanche.com](mailto:scott@sawtoothavalanche.com) |
| Montana State University Snow Science | Karl Birkeland  (Multiple attempts, no reply) |

[u1070830@notchpeak6 ~]$ conda list -n xlab

# packages in environment at /uufs/chpc.utah.edu/common/home/u1070830/anaconda3/envs/xlab:

#

# Name Version Build Channel

\_libgcc\_mutex 0.1 main

\_tflow\_select 2.3.0 eigen

absl-py 0.11.0 py38h578d9bd\_0 conda-forge

affine 2.3.0 py\_0 conda-forge

aiohttp 3.7.3 py38h25fe258\_0 conda-forge

appdirs 1.4.3 py\_1 conda-forge

argon2-cffi 20.1.0 py38h7b6447c\_1

arm\_pyart 1.11.2 py38hab2c0dc\_0 conda-forge

astor 0.8.1 pyh9f0ad1d\_0 conda-forge

astunparse 1.6.3 pyhd8ed1ab\_0 conda-forge

async-timeout 3.0.1 py\_1000 conda-forge

async\_generator 1.10 py\_0

attrs 20.2.0 py\_0

backcall 0.2.0 py\_0

blas 1.0 mkl

bleach 3.2.1 py\_0

blinker 1.4 py\_1 conda-forge

bokeh 2.2.1 py38\_0

boost-cpp 1.72.0 h8e57a91\_0 conda-forge

boto3 1.15.13 py\_0

botocore 1.18.13 py\_0

brotlipy 0.7.0 py38h7b6447c\_1000

bzip2 1.0.8 h7b6447c\_0

c-ares 1.16.1 h516909a\_3 conda-forge

ca-certificates 2020.12.5 ha878542\_0 conda-forge

cachetools 4.2.1 pyhd8ed1ab\_0 conda-forge

cairo 1.16.0 hcf35c78\_1003 conda-forge

cartopy 0.18.0 py38h172510d\_0 conda-forge

certifi 2020.12.5 py38h578d9bd\_1 conda-forge

cffi 1.14.3 py38h5bae8af\_0 conda-forge

cfgrib 0.9.8.4 py\_0 conda-forge

cfitsio 3.470 hce51eda\_6 conda-forge

cftime 1.2.1 py38h8790de6\_0 conda-forge

chardet 3.0.4 py38\_1003

click 7.1.2 py\_0

click-plugins 1.1.1 py\_0 conda-forge

cligj 0.5.0 py\_0 conda-forge

cloudpickle 1.6.0 py\_0

cryptography 3.1.1 py38h1ba5d50\_0

curl 7.71.1 he644dc0\_8 conda-forge

cycler 0.10.0 py38\_0

cytoolz 0.11.0 py38h7b6447c\_0

dask 2.30.0 py\_0

dask-core 2.30.0 py\_0

dbus 1.13.16 hb2f20db\_0

decorator 4.4.2 py\_0

defusedxml 0.6.0 py\_0

descartes 1.1.0 py\_4 conda-forge

distributed 2.30.0 py38\_0

docutils 0.15.2 py38\_0

eccodes 2.17.0 h59f7be3\_1 conda-forge

entrypoints 0.3 py38\_0

expat 2.2.9 he6710b0\_2

fiona 1.8.13.post1 py38hc820daa\_0

fontconfig 2.13.1 h86ecdb6\_1001 conda-forge

freetype 2.10.2 h5ab3b9f\_0

freexl 1.0.5 h516909a\_1002 conda-forge

fsspec 0.8.0 py\_0

g2clib 1.6.0 hf3f1b0b\_9 conda-forge

gast 0.3.3 py\_0 conda-forge

gdal 3.0.4 py38h172510d\_6 conda-forge

geopandas 0.8.1 py\_0

geos 3.8.1 he1b5a44\_0 conda-forge

geotiff 1.5.1 h05acad5\_10 conda-forge

gettext 0.19.8.1 hc5be6a0\_1002 conda-forge

giflib 5.2.1 h516909a\_2 conda-forge

glib 2.66.1 h680cd38\_0 conda-forge

google-auth 1.24.0 pyhd3deb0d\_0 conda-forge

google-auth-oauthlib 0.4.1 py\_2 conda-forge

google-pasta 0.2.0 pyh8c360ce\_0 conda-forge

grpcio 1.33.2 py38heead2fc\_2 conda-forge

gst-plugins-base 1.14.5 h0935bb2\_2 conda-forge

gstreamer 1.14.5 h36ae1b5\_2 conda-forge

h5netcdf 0.8.1 py\_0 conda-forge

h5py 2.10.0 nompi\_py38h513d04c\_102 conda-forge

hdf4 4.2.13 h3ca952b\_2

hdf5 1.10.5 nompi\_h3c11f04\_1104 conda-forge

hdfeos2 2.20 h64bfcee\_1000 conda-forge

hdfeos5 5.1.16 h8b6279f\_6 conda-forge

heapdict 1.0.1 py\_0

icu 64.2 he1b5a44\_1 conda-forge

idna 2.10 py\_0

importlib-metadata 1.7.0 py38\_0

importlib\_metadata 1.7.0 0

importlib\_resources 3.0.0 py38h32f6830\_0 conda-forge

intel-openmp 2020.2 254

ipykernel 5.3.4 py38h5ca1d4c\_0

ipython 7.18.1 py38h5ca1d4c\_0

ipython\_genutils 0.2.0 py38\_0

ipywidgets 7.5.1 py\_1

jasper 1.900.1 hd497a04\_4

jedi 0.17.2 py38\_0

jinja2 2.11.2 py\_0

jmespath 0.10.0 py\_0

joblib 0.17.0 py\_0

jpeg 9d h516909a\_0 conda-forge

json-c 0.13.1 hbfbb72e\_1002 conda-forge

json5 0.9.5 py\_0

jsonschema 3.2.0 py38\_1

jupyter 1.0.0 py38\_7

jupyter\_client 6.1.7 py\_0

jupyter\_console 6.2.0 py\_0

jupyter\_core 4.6.3 py38\_0

jupyterlab 2.2.6 py\_0

jupyterlab\_pygments 0.1.2 py\_0

jupyterlab\_server 1.2.0 py\_0

kealib 1.4.13 hec59c27\_0 conda-forge

keras 2.4.3 py\_0 conda-forge

keras-preprocessing 1.1.2 pyhd8ed1ab\_0 conda-forge

kiwisolver 1.2.0 py38hfd86e86\_0

krb5 1.17.1 hfafb76e\_3 conda-forge

lcms2 2.11 h396b838\_0

ld\_impl\_linux-64 2.33.1 h53a641e\_7

libaec 1.0.4 he6710b0\_1

libcurl 7.71.1 hcdd3856\_8 conda-forge

libdap4 3.20.6 h1d1bd15\_1 conda-forge

libedit 3.1.20191231 h14c3975\_1

libev 4.33 h516909a\_1 conda-forge

libffi 3.2.1 he1b5a44\_1007 conda-forge

libgcc-ng 9.1.0 hdf63c60\_0

libgdal 3.0.4 h3dfc09a\_6 conda-forge

libgfortran-ng 7.3.0 hdf63c60\_0

libiconv 1.16 h516909a\_0 conda-forge

libkml 1.3.0 hd79254b\_1012 conda-forge

libnetcdf 4.7.4 nompi\_h9f9fd6a\_101 conda-forge

libnghttp2 1.41.0 h8cfc5f6\_2 conda-forge

libpng 1.6.37 hbc83047\_0

libpq 12.3 h5513abc\_0 conda-forge

libprotobuf 3.13.0.1 h8b12597\_0 conda-forge

libsodium 1.0.18 h7b6447c\_0

libspatialindex 1.9.3 he6710b0\_0

libspatialite 4.3.0a h2482549\_1038 conda-forge

libssh2 1.9.0 h1ba5d50\_1

libstdcxx-ng 9.1.0 hdf63c60\_0

libtiff 4.1.0 h2733197\_1

libuuid 2.32.1 h14c3975\_1000 conda-forge

libwebp-base 1.1.0 h516909a\_3 conda-forge

libxcb 1.14 h7b6447c\_0

libxml2 2.9.10 hee79883\_0 conda-forge

locket 0.2.0 py38\_1

lz4-c 1.9.2 he6710b0\_1

markdown 3.3.3 pyh9f0ad1d\_0 conda-forge

markupsafe 1.1.1 py38h7b6447c\_0

matplotlib 3.3.1 0

matplotlib-base 3.3.1 py38h817c723\_0

metpy 0.12.2 py\_0 conda-forge

mistune 0.8.4 py38h7b6447c\_1000

mkl 2020.2 256

mkl-service 2.3.0 py38he904b0f\_0

mkl\_fft 1.2.0 py38h23d657b\_0

mkl\_random 1.1.1 py38h0573a6f\_0

msgpack-python 1.0.0 py38hfd86e86\_1

multidict 4.7.5 py38h1e0a361\_2 conda-forge

munch 2.5.0 py\_0

nbclient 0.5.0 py\_0

nbconvert 6.0.7 py38\_0

nbformat 5.0.7 py\_0

ncurses 6.2 he6710b0\_1

nest-asyncio 1.4.1 py\_0

netcdf4 1.5.3 nompi\_py38heb6102f\_103 conda-forge

notebook 6.1.4 py38\_0

numpy 1.19.1 py38hbc911f0\_0

numpy-base 1.19.1 py38hfa32c7d\_0

oauthlib 3.0.1 py\_0 conda-forge

olefile 0.46 py\_0

openjpeg 2.3.1 h981e76c\_3 conda-forge

openssl 1.1.1h h516909a\_0 conda-forge

opt\_einsum 3.3.0 py\_0 conda-forge

owslib 0.20.0 py\_0 conda-forge

packaging 20.4 py\_0

pandas 1.1.3 py38he6710b0\_0

pandoc 2.10.1 0

pandocfilters 1.4.2 py38\_1

parso 0.7.0 py\_0

partd 1.1.0 py\_0

pcre 8.44 he6710b0\_0

pexpect 4.8.0 py38\_0

pickleshare 0.7.5 py38\_1000

pillow 7.2.0 py38hb39fc2d\_0

pint 0.16.1 py\_0 conda-forge

pip 20.2.3 py38\_0

pixman 0.38.0 h516909a\_1003 conda-forge

pooch 1.2.0 py\_0 conda-forge

poppler 0.67.0 h14e79db\_8 conda-forge

poppler-data 0.4.9 1 conda-forge

postgresql 12.3 h8573dbc\_0 conda-forge

proj 7.0.0 h966b41f\_5 conda-forge

prometheus\_client 0.8.0 py\_0

prompt-toolkit 3.0.7 py\_0

prompt\_toolkit 3.0.7 0

protobuf 3.13.0.1 py38hadf7658\_1 conda-forge

psutil 5.7.2 py38h7b6447c\_0

ptyprocess 0.6.0 py38\_0

pyasn1 0.4.8 py\_0 conda-forge

pyasn1-modules 0.2.7 py\_0 conda-forge

pycparser 2.20 py\_2

pyepsg 0.4.0 py\_0 conda-forge

pygments 2.7.1 py\_0

pygrib 2.0.5 py38hfcef17a\_0 conda-forge

pyjwt 2.0.1 pyhd8ed1ab\_0 conda-forge

pynio 1.5.5 py38h031d99c\_12 conda-forge

pyopenssl 19.1.0 py\_1

pyparsing 2.4.7 py\_0

pyproj 2.6.1.post1 py38h7521cb9\_0 conda-forge

pyqt 5.9.2 py38h05f1152\_4

pyrsistent 0.17.3 py38h7b6447c\_0

pyshp 2.1.2 pyh9f0ad1d\_0 conda-forge

pysocks 1.7.1 py38\_0

python 3.8.3 cpython\_he5300dc\_0 conda-forge

python-dateutil 2.8.1 py\_0

python\_abi 3.8 1\_cp38 conda-forge

pytz 2020.1 py\_0

pyyaml 5.3.1 py38h7b6447c\_1

pyzmq 19.0.2 py38he6710b0\_1

qt 5.9.7 h0c104cb\_3 conda-forge

qtconsole 4.7.7 py\_0

qtpy 1.9.0 py\_0

rasterio 1.1.5 py38h033e0f6\_1 conda-forge

readline 8.0 h7b6447c\_0

regionmask 0.6.1 py\_1 conda-forge

requests 2.24.0 py\_0

requests-oauthlib 1.3.0 pyh9f0ad1d\_0 conda-forge

rsa 4.7 pyhd3deb0d\_0 conda-forge

rtree 0.9.4 py38\_1

s3fs 0.3.0 py\_0 conda-forge

s3transfer 0.3.3 py38\_0

scikit-learn 0.23.2 py38h0573a6f\_0

scipy 1.5.2 py38h0b6359f\_0

seaborn 0.11.0 py\_0

send2trash 1.5.0 py38\_0

setuptools 50.3.0 py38hb0f4dca\_1

shapely 1.7.1 py38hc7361b7\_0 conda-forge

sip 4.19.13 py38he6710b0\_0

six 1.15.0 py\_0

snuggs 1.4.7 py\_0 conda-forge

sortedcontainers 2.2.2 py\_0

sqlite 3.33.0 h62c20be\_0

tbb 2020.2 hc9558a2\_0 conda-forge

tblib 1.7.0 py\_0

tensorboard 2.4.1 pyhd8ed1ab\_0 conda-forge

tensorboard-plugin-wit 1.8.0 pyh44b312d\_0 conda-forge

tensorflow 2.3.0 eigen\_py38h71ff20e\_0

tensorflow-base 2.3.0 eigen\_py38hb57a387\_0

tensorflow-estimator 2.4.0 pyh9656e83\_0 conda-forge

termcolor 1.1.0 py\_2 conda-forge

terminado 0.8.3 py38\_0

testpath 0.4.4 py\_0

threadpoolctl 2.1.0 pyh5ca1d4c\_0

tiledb 1.7.7 h8efa9f0\_3 conda-forge

tk 8.6.10 hbc83047\_0

toolz 0.11.1 py\_0

tornado 6.0.4 py38h7b6447c\_1

traitlets 5.0.4 py38\_0

trmm\_rsl 1.49 3 conda-forge

typing-extensions 3.7.4.3 0 conda-forge

typing\_extensions 3.7.4.3 py\_0

tzcode 2020a h516909a\_0 conda-forge

urllib3 1.25.10 py\_0

wcwidth 0.2.5 py\_0

webencodings 0.5.1 py38\_1

werkzeug 1.0.1 pyh9f0ad1d\_0 conda-forge

wheel 0.35.1 py\_0

widgetsnbextension 3.5.1 py38\_0

wrapt 1.12.1 py38h1e0a361\_1 conda-forge

wrf-python 1.3.2 py38h7eb8c7e\_1 conda-forge

xarray 0.16.1 py\_0

xerces-c 3.2.2 h8412b87\_1004 conda-forge

xlrd 1.2.0 py\_0

xorg-kbproto 1.0.7 h14c3975\_1002 conda-forge

xorg-libice 1.0.10 h516909a\_0 conda-forge

xorg-libsm 1.2.3 h84519dc\_1000 conda-forge

xorg-libx11 1.6.12 h516909a\_0 conda-forge

xorg-libxext 1.3.4 h516909a\_0 conda-forge

xorg-libxrender 0.9.10 h516909a\_1002 conda-forge

xorg-renderproto 0.11.1 h14c3975\_1002 conda-forge

xorg-xextproto 7.3.0 h14c3975\_1002 conda-forge

xorg-xproto 7.0.31 h14c3975\_1007 conda-forge

xz 5.2.5 h7b6447c\_0

yaml 0.2.5 h7b6447c\_0

yarl 1.6.3 py38h25fe258\_0 conda-forge

zeromq 4.3.2 he6710b0\_3

zict 2.0.0 py\_0

zipp 3.3.0 py\_0

zlib 1.2.11 h7b6447c\_3

zstd 1.4.5 h9ceee32\_0