Practical Session on Seasonal Rainfall Forecasting with Climate Model Ensembles

PREPARE Drought and Flood Early Warning for Pacific Islands
Training Workshop
Nadi, Fiji, 15-20 July 2024

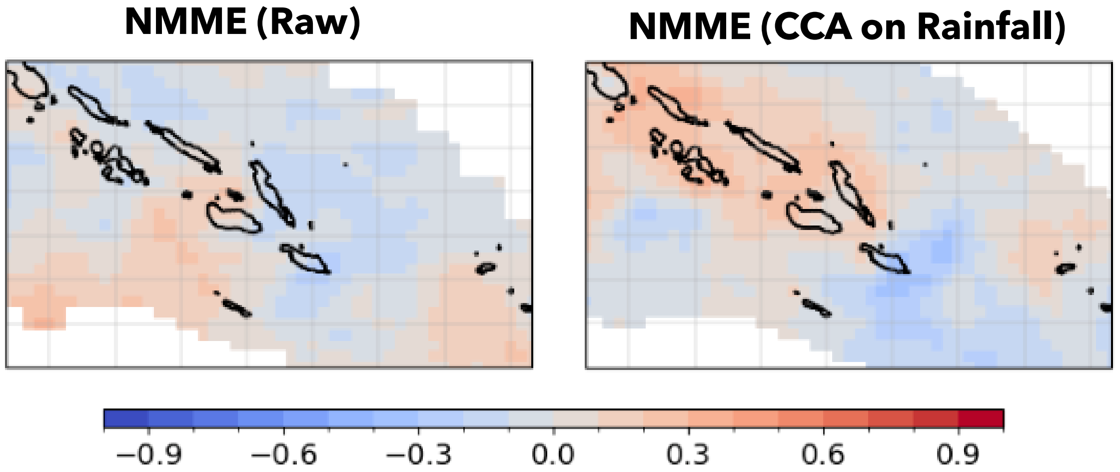
Dr. Katie Kowal

NOAA/CPC/International Desks

17 July 2024

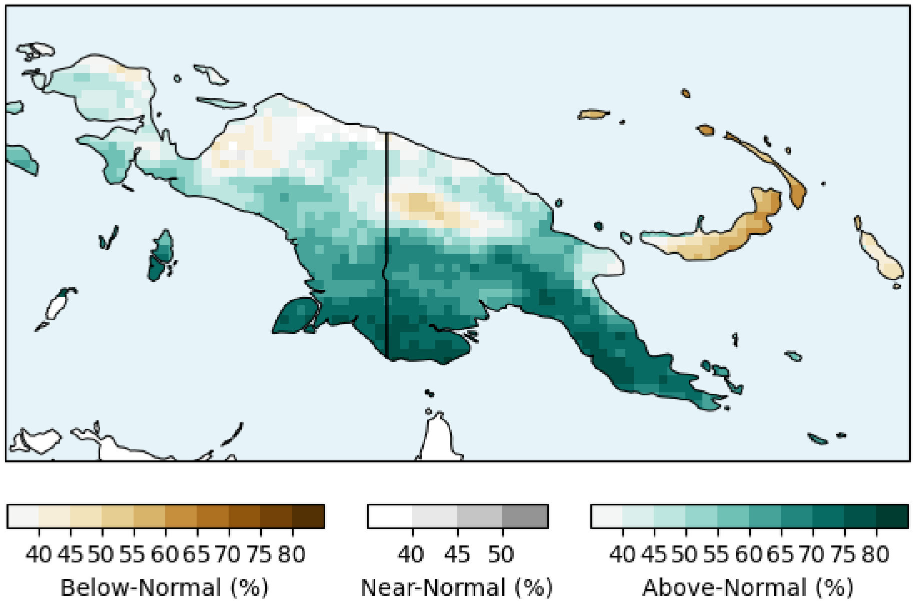
Goals of Today's Practical Session

Use	**Use CCA to bias correct an NMME forecast**
Compare	**Compare** the **CCA approaches** to see how **driving forecasts with SSTs vs rainfall** affects forecast performance
Consider	**Consider seasonal variability** of the correction techniques
Consider	**Consider spatial variability** of the above correction techniques
Predict	**Predict seasonal rainfall** by generating a probabilistic tercile forecast using a bias correction technique of choice



Solomon Islands Pearson Correlation, Initialized Sep; Oct-Dec Prediction

Papua New Guinea Initialized Jun; Aug-Oct Forecast (CCA on Rainfall)

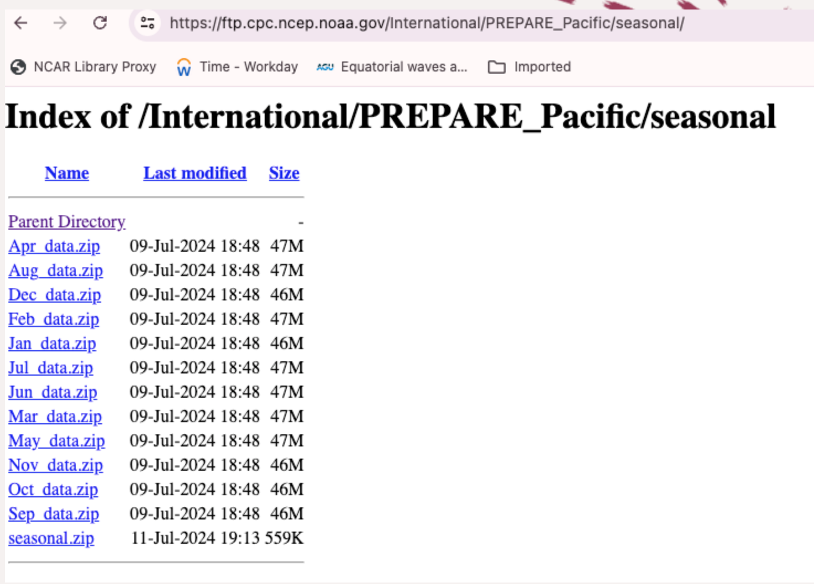


Example Plots

Preparing for the Practical Session

- **DIVIDE INTO GROUPS** – **By region**, we need **at least 4 people per group** with a computer **that can run code in the intdesk_train environment**
- **ASSIGN ROLES –**
  - **CCA – SST global oceans**
  - **CCA – Precip large region**
  - **CCA – SST pacific ocean**
  - **CCA – Precip small target zone for your island**
- **PICK 1st SEASON OF INTEREST** – choose one initialization month, e.g. if you pick August, you will test 3 forecasts over three lead times : Sep-Nov, Oct-Dec, and Nov-Jan
- **PICK 2nd SEASON OF INTEREST** to run if there is time

Starting the Practical Session



- Download the demo file (**Windows users**):
  1. **Download file** from:

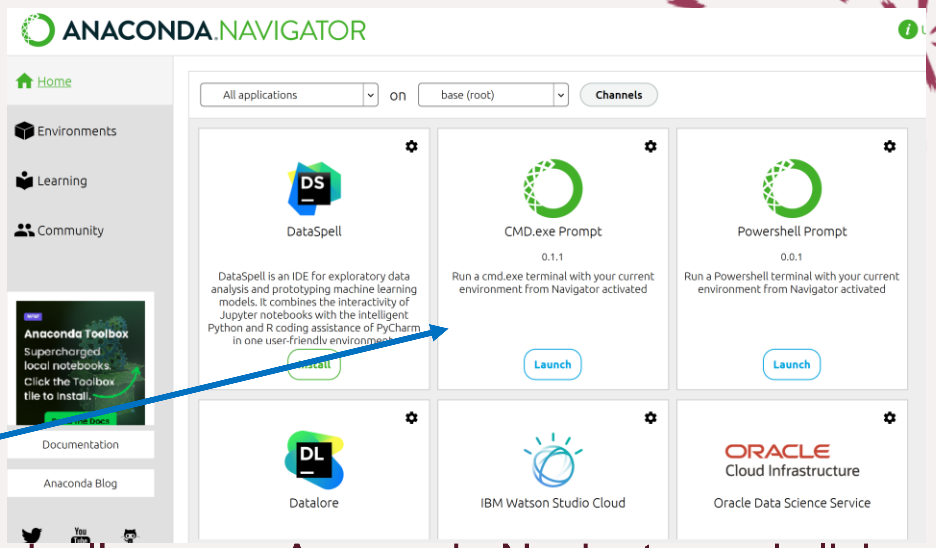
     https://ftp.cpc.ncep.noaa.gov/International/PREPARE_Pacific/seasonal/seasonal.zip
  2. **Save compressed file** into your working project directory (e.g. C:\Users\“your user name“\Desktop\pacisl_workshop“)
  3. **Right click and unzip seasonal.zip**
- Download the demo file (**Linux & Mac users**):
  1. **Change your directory** to your home folder using your Ubuntu terminal , **replace ‘home_folder_name’** with name of your directory (e.g. C:/Users/USERNAME)

     $ cd home_folder_name
  2. **Download file** using ‘wget’ command in terminal:

     $ wget --no-check-certificate https://ftp.cpc.ncep.noaa.gov/International/PREPARE_Pacific/seasonal/seasonal.zip
  3. **Unzip file** using

     $ unzip seasonal.zip

Practical Session Setup



Launch a Terminal Window – many ways, could search for 'Terminal' or open Anaconda Navigator and click on the 'CMD.exe Prompt' icon to launch

Change directory to seasonal using your path you put your seasonal folder in:

$ cd seasonal

OR

$ cd Desktop/pacisl_workshop/seasonal

Launch Jupyter Notebook

1. **Activate** the **intdesk_train** environment

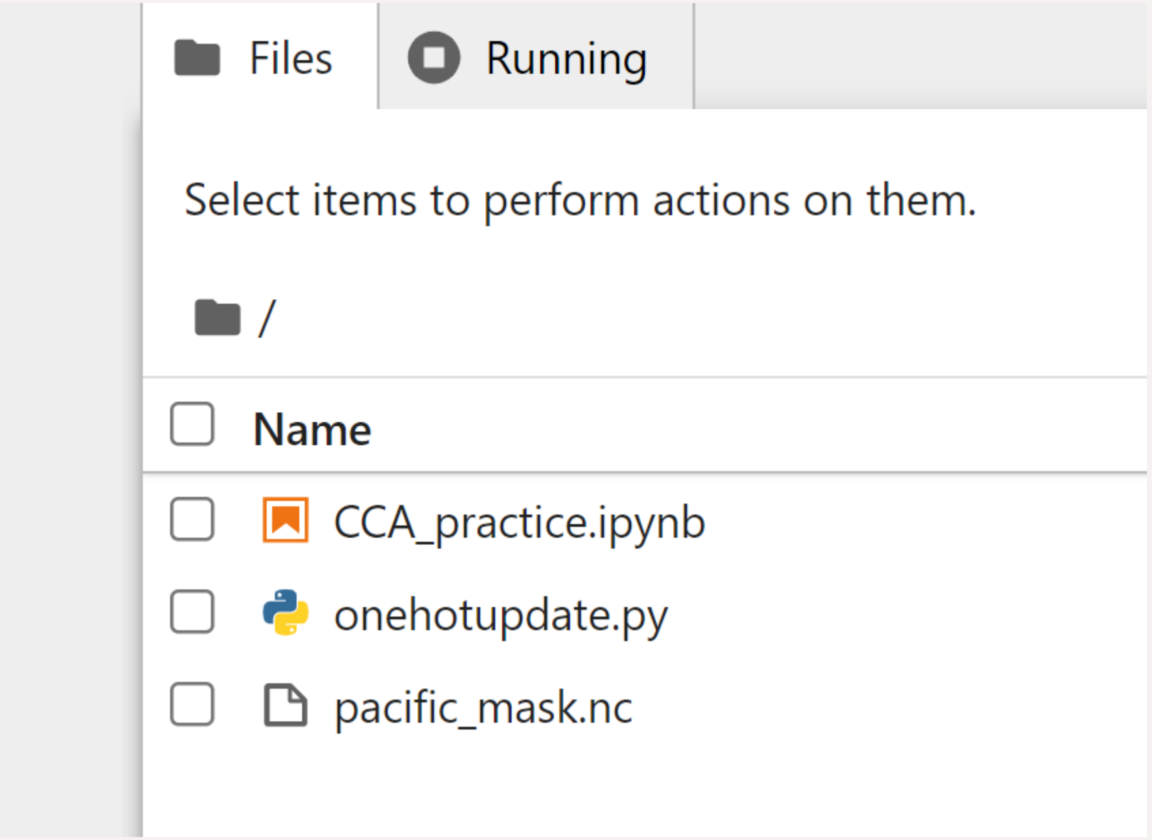
   $ conda activate intdesk_train

2. **Launch Jupyter Notebook**, using the command below:

   $ jupyter notebook

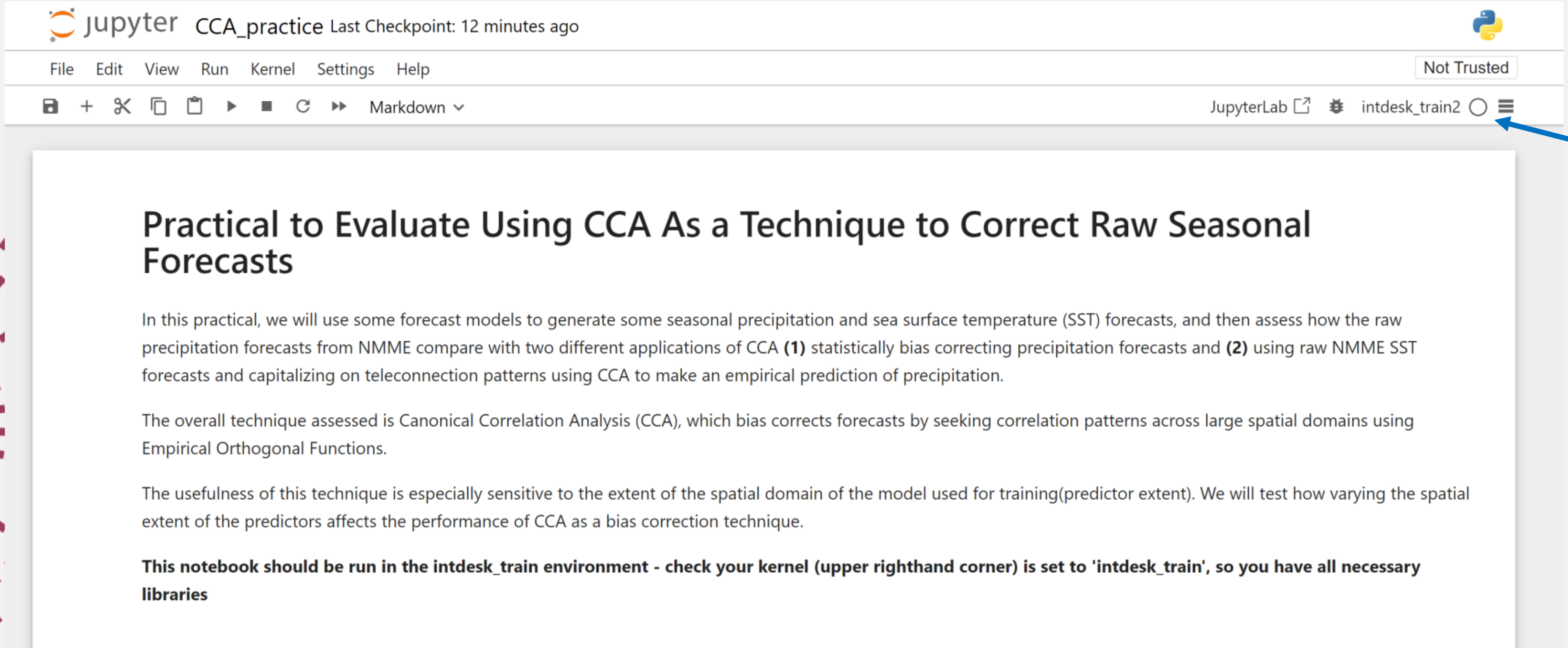
The notebook should launch automatically via your default Internet browser.

Practical Session Setup (cont.)



Click **CCA_practice.ipynb** to open the Canonical Correlation analysis (CCA) code for the seasonal practical

Practical Session Setup (cont.)



Practical to Evaluate Using CCA As a Technique to Correct Raw Seasonal Forecasts

In this practical, we will use some forecast models to generate some seasonal precipitation and sea surface temperature (SST) forecasts, and then assess how the raw precipitation forecasts from NMME compare with two different applications of CCA **(1)** statistically bias correcting precipitation forecasts and **(2)** using raw NMME SST forecasts and capitalizing on teleconnection patterns using CCA to make an empirical prediction of precipitation.

The overall technique assessed is Canonical Correlation Analysis (CCA), which bias corrects forecasts by seeking correlation patterns across large spatial domains using Empirical Orthogonal Functions.

The usefulness of this technique is especially sensitive to the extent of the spatial domain of the model used for training(predictor extent). We will test how varying the spatial extent of the predictors affects the performance of CCA as a bias correction technique.

This notebook should be run in the intdesk_train environment - check your kernel (upper righthand corner) is set to 'intdesk_train', so you have all necessary libraries

Before running anything, check that you have the right environment activated in your Jupyter kernel, should be 'intdesk_train'

Practical Session Setup (cont.)

Libraries

```python
import xcast as xc
import datetime as dt
from datetime import datetime
import numpy as np
from pathlib import Path
import xarray as xr
import os
import time
import cartopy.crs as ccrs
import cartopy.feature as cf
import matplotlib.pyplot as plt
import glob
from cartopy.feature import NaturalEarthFeature
import urllib.request
from zipfile import ZipFile
```

1. **Import the required python libraries**

NOTE: You can quickly run a 'cell' in a Jupyter Notebook by click on the cell and then clicking 'Shift' + 'Enter'

Project Directory Setup ¶

```python
# setup the folder where you want to work for this project
# either type in the location of your working directory below, or place this notebook in the folder where you
project_dir = os.getcwd()#"/Users/katie/Desktop/trial_pacisl"
print('Project Directory is Located in ' + project_dir)

#makes subdirectores to organize your work within the project if they don't already exist
os.makedirs(os.path.join(project_dir, 'practical_data'), exist_ok = True)
data_dir = os.path.join(project_dir, 'practical_data')
os.makedirs(os.path.join(project_dir, 'practical_figures'), exist_ok = True)
figure_dir = os.path.join(project_dir, 'practical_figures')
```

```
Project Directory is Located in /cpc/int_desk/pac_isl/analysis/xcast/seasonal/practical_notebooks
```

2. **Setup the project directories**

- .ipynb_checkpoints
- __pycache__
- practical_data
- practical_figures
- CCA_practice.ipynb
- onehotupdate.py
- pacific_mask.nc

Practical Session Setup (cont.)

Setup Your Constants: Spatial Extents and Dates

(1) initialization date: Make sure this date is not set for a future month but in a present or past month, and then seasonal forecasts will be created for 3 target periods following that month. For instance if you pick (2023,8,1) as your initialization date, you will evaluate forecasts over Sep-Nov, Oct-Dec, and Dec-Feb.

```
## PICK YOUR DATE you want to initialize the model, e.g. your current date
initial_date = (2023, 9, 1)
```

(2) region of interest: Several coordinates have been setup in this cell. Make sure your region_coords variable is equal to the name of one of the coordinate dictionary entries in this cell, and then name that region as you like, e.g. region_coords = solomon_coordinates, region_of_interest = 'Solomon Islands'. You can adjust the coordinate values as you like in the dictionaries if you want to play around with the predictand extent, just remember to keep the naming consistent.

```
### PICK YOUR TARGET REGION OF INTEREST
region_of_interest = 'Pacific Islands' #how you want to name your region (can include spaces)
region_coords = pacislands_coordinates #name of the coordinates to use for your region, as de
```

1. **Choose month based on your 1st season of interest**
2. **Look at the region of interest coordinate options in next cell**
3. **Name your region of interest and select coordinates to match name as defined in previous cell**

NOTE: if you have 2 regions of interest in your group you can adjust the code as follows:

region_of_interest = [‘Pacific Islands’ , ‘Tuvalu’]
region_coords = [pacislands_coords, tuvalu_coords]

Practical Session Setup (cont.) – by Team Role

(3) type of predictor: This will either be set to 'sst' or 'precip', which will affect the type of forecast you generate, either using raw NMME SST predictions and statistically relating those to rainfall observations over a training period or raw NMME rainfall predictions and statistically relating those to rainfall observations over a training period.

1. Pick predictor based on your role

```
### PICK YOUR PREDICTOR, either choose 'precip' or 'sst'
predictor_type = 'sst'
```

```
### PICK YOUR PREDICTOR TRAINING ZONE

#some options outlined below
#Pacific region, encompassing all islands
pacific_extent = {
    'west': 120,
    'east': 210,
    'north': 10,
    'south': -30
}

#Global oceans, including Pacific and Atlantic
global_extent = {
    'west': 100,
    'east': 270,
    'north': 30,
    'south': -30
}

predictor_train_extent = pacific_extent
predictor_train_extent_name = 'pacific'

#choose extent you want to train your models -
#'whole region' will be all of Pacific Islands,
#'subregion' will train the model on a buffer zone slightly large than your island
predictand_train_extent = 'whole region'
```

Make sure this name exactly matches one of the coordinate dictionaries above

2. Pick predictor and predictand training extents based on your role

Role	Predictor	Predictor Extent	Predictand Extent
CCA - SST global	sst	global_extent	'whole region'
CCA - SST pacific	sst	pacific_extent	'whole region'
CCA - Precip large	precip	pacific_extent	'whole region'
CCA - Precip region	Precip	pacific_extent	'subregion'

Practical Session Data Prep

1. **Run the next cell and check the print out to make sure the seasons you are predicting is what you want**

```
#this cell is setup to calculate your target forecast months based on your initiali
#the forecast months are currently seto to be 1-3, 2-4 and 3-5 months ahead
number_to_month_name_dictionary = {
    1: 'Jan',
    2: 'Feb',
    3: 'Mar',
    4: 'Apr',
    5: 'May',
    6: 'Jun',
    7: 'Jul',
    8: 'Aug',
    9: 'Sep',
    10: 'Oct',
    11: 'Nov',
    12: 'Dec',
    0: 'Dec'
}

leads = [['1', '3'],['2', '4'], ['3','5']]
initial_month = dt.datetime(*initial_date).month
initial_month_name = number_to_month_name_dictionary[initial_month]
target_seas = []
for l in leads:
    target_low = number_to_month_name_dictionary[(initial_month + float(l[0]))%12]
    target_high = number_to_month_name_dictionary[(initial_month + float(l[1]))%12]
    target_seas.append('-'.join([target_low, target_high]))
print('Target seasons to forecast')
print(target_seas)
```

```
Target seasons to forecast
['Oct-Dec', 'Nov-Jan', 'Dec-Feb']
```

2. **Download your data for your season of interest, print out will give you the name of the file based on initialization month**

```
# Download data
def download_file(url, output_filename):
    try:
        # Open the URL and download the content
        with urllib.request.urlopen(url) as response:
            # Read the content
            file_content = response.read()

            # Save the content to a file
            with open(output_filename, 'wb') as f:
                f.write(file_content)

            print(f"File downloaded successfully as {output_filename}")

    except Exception as e:
        print(f"Failed to download file from {url}: {e}")

url1 = "https://ftp.cpc.ncep.noaa.gov/International/PREPARE_Pacific/seasonal/" + initial_mont

os.chdir(data_dir)
download_file(url1, initial_month_name + '_data.zip')

# loading the temp.zip and creating a zip object
with ZipFile(initial_month_name + '_data.zip', 'r') as zObject:

    # Extracting all the members of the zip
    # into a specific location.
    zObject.extractall(
        path=data_dir)
os.chdir(project_dir)

download_data_dir = os.path.join(data_dir, initial_month_name + '_data')
```

```
File downloaded successfully as Sep_data.zip
```

Practical Session Data Prep (cont.)

1. Run this very long cell to load all your observed and model data

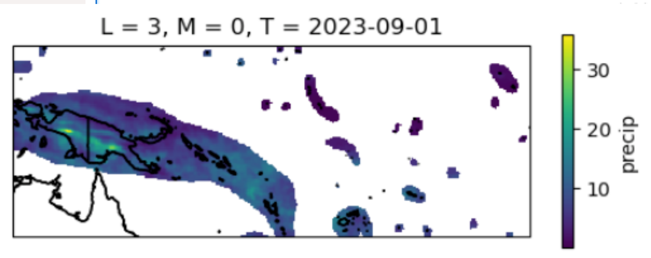
Load Observations and Model Data

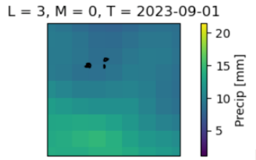
```
obs_leads = xr.open_dataset(os.path.join(data_dir, initial_month_name + '_da

#train the model on observations over a grid slightly larger than the region
#this could be updated for later fine-tuning, but calculated here to keep it
if predictand_train_extent_name == 'whole region':
    predictand_train_extent = pacislands_coordinates.copy()
elif predictand_train_extent_name == 'subregion':
    if type(region_of_interest) is list:
        predictand_train_extent = []
        for r in region_coords:
            predictand_train_extent.append({
            'west':  r['west']-5,
            'east': r['east']+5,
            'north': r['north']+3,
            'south': r['south']-5
    })
    else:
        predictand_train_extent = {
        'west':  region_coords['west']-5,
            'east': region_coords['east']+5,
            'north': region_coords['north']+3,
            'south': region_coords['south']-5
    }

if type(predictand_train_extent) is list:
    obs_crop = []
    for p in predictand_train_extent:
        obs_crop.append(obs_leads.sel(X=slice(p['west'],
                                   p['east']),
                          Y=slice(p['south'],
```

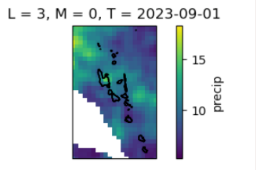
2. Check your obs / model data is what you want by running code to plot data in following cells

Check your region of interest is what you want

```
if type(obs_crop) is list:
    for o in obs_crop:
        obs_check = obs_crop.sel(X=slice(predictand_train_extent['west'],
                                 predictand_train_extent['east']),
                         Y=slice(predictand_trai
                                 dictand_trai
                         ols=1, figsi
                         ojection': c
```

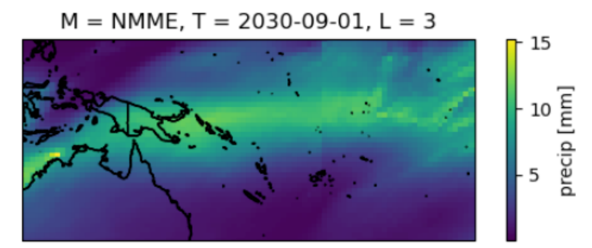


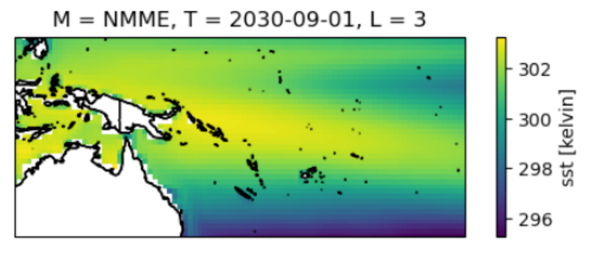




Check your model data is what you want

```
model_check = hindcast_data.sel(X=slice(predictor_train_extent['west'], pred
                         Y=slice(predictor_train_extent['south'], predictor

fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(6,2),
                         subplot_kw={'projection': ccrs.PlateCarree(central_
```





Practical Session Analysis

Analysis

Correct raw model outputs using Canonical Correlation Analysis (CCA)

```
start_time = time.time()
cca_fcsts_prob, cca_fcsts_det, cca_hcasts_det, cca_hcasts_prob, obs_to_test = [],
y_cca_loadings_test, x_cca_loadings_test, y_eof_loadings_test, x_eof_loadings_tes

for l in np.unique(hindcast_data.L):
    obs = obs_leads.sel(L=l).precip
    model = hindcast_data.sel(L=l)[predictor_type]
    fmodel = forecast_data.sel(L=l)[predictor_type]

    #run CCA
    hindcasts_det, hindcasts_prob, obs_test, y_cca_loadings, x_cca_loadings, y_eo
    i=1
    for xtrain, ytrain, xtest, ytest in xc.CrossValidator(model, obs, window=5):
        print("window {}".format(i))
        i += 1
        reg = xc.CCA(search_override=(5,
                                      5,
                                      3))
```

Run the CCA calculation

(this may take several minutes)

This algorithm is run on 75 windows across 3 seasons (will count 1 to 75 3 times per region/season tested)

NOTE: Katie setup a leave 5 years out testing window here, you could change this later

NOTE: Katie chose predictor modes as 5, predictand modes as 5, And CCA correlation modes as 3, you could change this later

While CCA is running...

Nominate a power point leader in your team
Consider the questions you will want to assess, make a powerpoint to aggregate your group's results -

(1) What does the **analysis** look like spatially when you run CCA across your 4 test cases?

- EOF loadings for X (predictors)
- EOF loadings for Y (predictands)

(2) How does CCA **compare** when applied using the **different training options**?
(3) How does CCA **compare as lead time increases**?
(4) How does CCA **compare seasonally** (if you have time to run multiple seasons)

Practical Session Analysis (cont.)

Pick Which Loadings to Examine

Select which loadings you want to examine, loadings can be either

x_pc_var_test: Percentage of the Variance Explained by a PC in Predictor Space (time series variable)

y_pc_var_test: Percentage of of the Variance Explained by a PC in Predictand Space (time series variable)

x_eof_loadings_test: EOF Loadings for Model Over the Predictor Space (spatial variable)

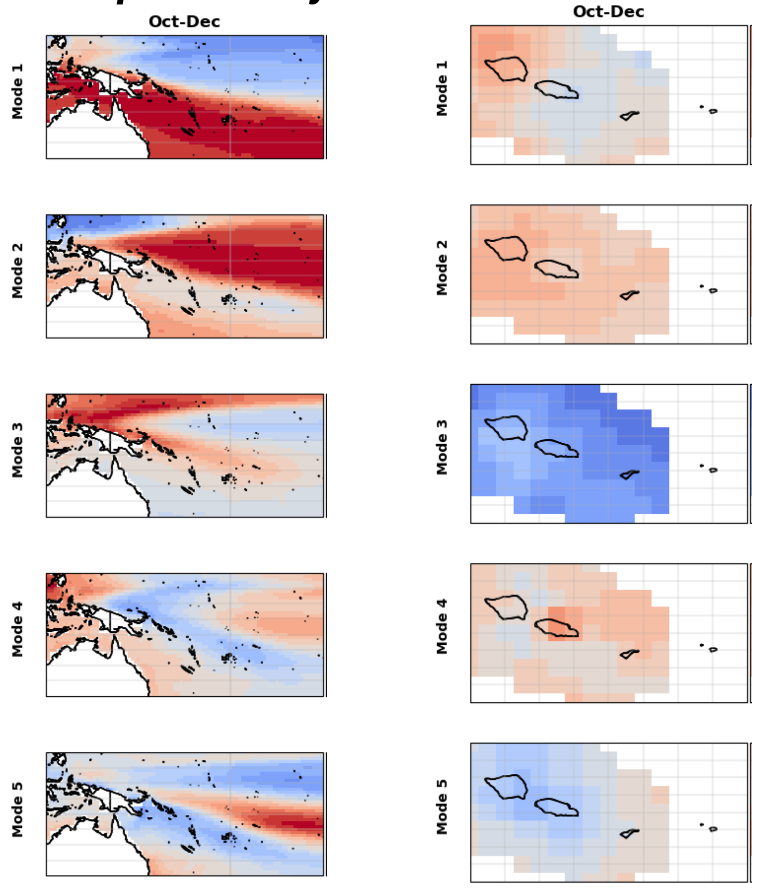
y_eof_loadings_test: EOF Loadings for Observations Over the Predictand Space (spatial variable)

x_cca_loadings_test: CCA Coefficients (Loadings) for the Modelled Time Series Projected back onto the Model EOF Loadings over Predictor Space (spatial variable)

y_cca_loadings_test: CCA Coefficients (Loadings) for the Observed Time Series Projected back onto the Observed EOF Loadings over Predictand Space (spatial variable)

```
#PICK WHAT ANALYSIS YOU WANT TO PLOT
analysis_type = x_pc_var_test
# modelled predictors ('x') or predicted precip observations ('y')
target_focus = 'x'
```

Example Figures:
EOF Spatial Analysis with SSTs and Samoa



X EOF **Y EOF**

Practical Session Evaluation with Pearson

1. **Run the Pearson correlation cell**

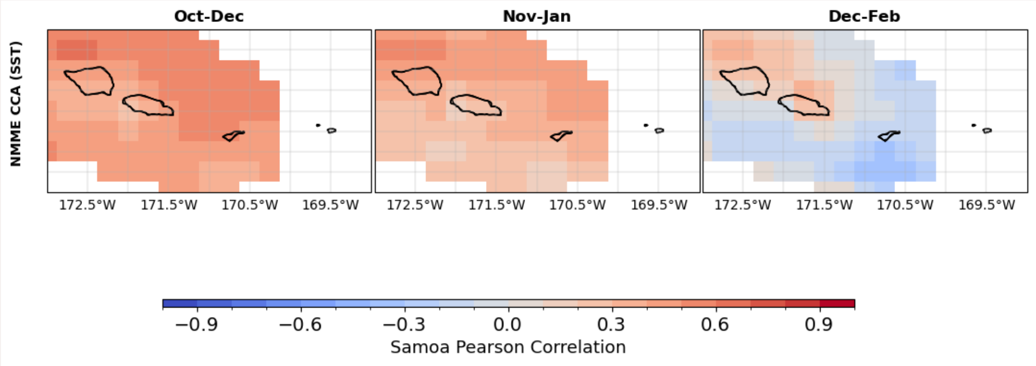
Evaluate Performance of Raw vs Bias Corrected Models

Pearson Correlation Coefficient

```
start_time = time.time()

if type(obs_crop) is list:
    obs_test = obs_crop.copy()
else:
    obs_test = [obs_crop]

#calculate pearson correlation score for hindcasts
pearsons = []
for c, cca_hcast_test in enumerate(cca_hcasts_det):
    pearson_cca_obs, pearson_raw_obs = [], []
    for l, lead in enumerate(np.unique(hindcast_data.L.values)):
        cca_pearson_calc = xc.Pearson(cca_hcasts_det[c].isel(L=l),obs_to_test[c].isel(L=l))
        cca_pearson_calc = cca_pearson_calc.expand_dims({'M':['NMME CCA (' + predictor_type.upper() +

        #regrid raw data for pearson calculation on one to one grid
        raw_regrid = xc.regrid(hindcast_data.isel(L=l)[predictor_type], obs_test[c].X, obs_test[c].Y)
        if region_coords[c] == chuuk_coordinates:
            hindcast_raw = raw_regrid.copy()
```

2. **Run the following cell to plot the Pearson correlations**

```
for p, pearsons_test in enumerate(pearsons):
    for r, region in enumerate(region_of_interest):
        models = np.unique(pearsons_test.M.values)
        models = np.flip(models, axis = 0)

        fig, axes = plt.subplots(nrows=len(models), ncols=len(target_seas), figsize=(12,
                                 subplot_kw={'projection': ccrs.PlateCarree(central_long

        # Set the extent to cover the entire world
        for ax in axes.flat:
            ax.set_global()
            ax._autoscaleXon = False
            ax._autoscaleYon = False

        for j, model in enumerate(models):
            for i, season in enumerate(target_seas):
                if predictor_type == 'sst':
                    ax = axes[i]
                else:
                    ax = axes[j, i]
                # Your plotting code here using the specific model and season
                xplot = pearsons_test.isel(L=i, M=j).plot(ax=ax,
                                          transform=ccrs.PlateCarree(),
                                          cmap='coolwarm', levels=21, vmin=-
```



NOTE – raw NMME values will only be compared if you are testing raw Precip NMME data, otherwise only CCA scores will be plotted

Practical Session Evaluation with GROCS

1. Calculate the GROCS Score for your Test Case

GROCS - Generalized Receiver Operating Characteristics Score

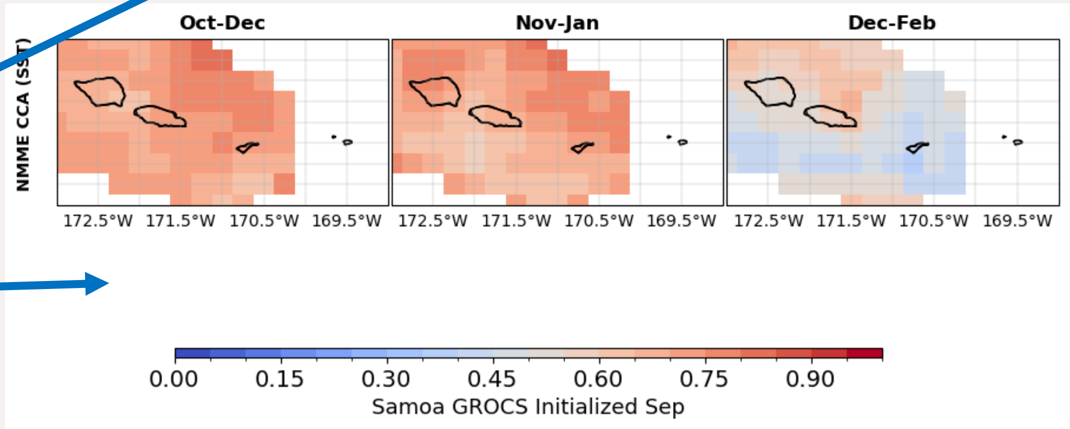
```python
start_time = time.time()
grocs = []
for c, cca_hcasts_test in enumerate(cca_hcasts_prob):
    grocs_cca = []
    for l, lead in enumerate(np.unique(hindcast_data.L.values)):

        hind_prob = xc.gaussian_smooth(cca_hcasts_prob[c].isel(L=l), kernel=3)
        obs = xc.gaussian_smooth(obs_to_test[c].isel(L=l), kernel=3)

        #transform obs into tercile based categories
        ohc = onehot.OneHotEncoder()
        ohc.fit(obs)
        T = ohc.transform(obs)
        clim = xr.ones_like(T) * 0.333

        grocs_cca_l = xc.GROCS(hind_prob, T)
        grocs_cca_l = grocs_cca_l.expand_dims({'M':['NMME CCA (' + predictor_type.upper() +
        grocs_cca.append(grocs_cca_l)

    grocs_cca = xr.concat(grocs_cca, dim = 'L')
    grocs.append(grocs_cca)
print('GROCS processing time is ' + str(time.time() - start_time))
```

2. Run the following cell to plot the results

```python
for g, grocs_test in enumerate(grocs):
    for r, region in enumerate(region_of_interest):
        models = np.unique(grocs_test.M.values)
        print(target_seas)
        fig, axes = plt.subplots(nrows = len(models), ncols=len(target_seas), figsize=(10, (1)*2 + 2)
                                 subplot_kw={'projection': ccrs.PlateCarree(central_longitude=180)})

        # Set the extent to cover the entire world
        for ax in axes.flat:
            ax.set_global()
            ax._autoscaleXon = False
            ax._autoscaleYon = False
        for j, model in enumerate(grocs_test.M.values):
            for i, season in enumerate(target_seas):
                ax = axes[i]
                # Your plotting code here using the specific model and season
                xplot = grocs_test.isel(L=i, M=j).plot(ax=ax, transform=ccrs.PlateCarree(),
                                                       cmap='coolwarm', levels=21, vmin=0, vmax=1, add
                ax.coastlines()
```

height

width



NOTE – if you don't like the width or height of your figure, play around with the 'figsize' parameter at the top of the cell, you can play around with how much white space is in your figure here

Practical Session Probabilistic Forecast Generation

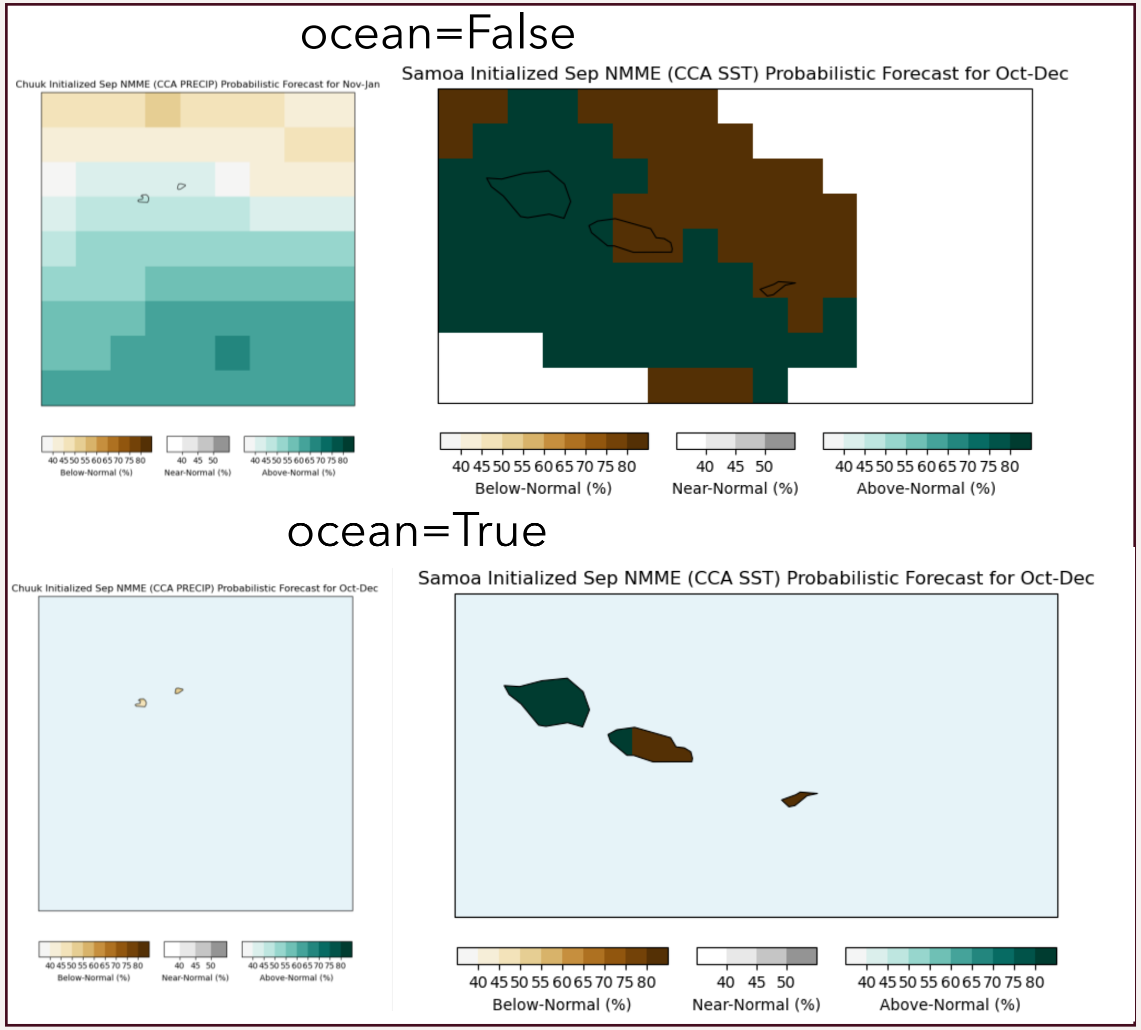
Plot the bias corrected probabilistic tercile forecast based on your test case

Plot the Bias Corrected Forecasts using CCA

Note, to mask the Oceans and only plot as blue, set 'ocean' = True, otherwise set 'ocean' equal to false to see the whole masked data

```
for c, cca_fcst_test in enumerate(cca_fcsts_prob):
    for r, region in enumerate(region_of_interest):
        for l, lead in enumerate(np.unique(cca_fcsts_prob[c].L)):
            im = xc.view_probabilistic(cca_fcsts_prob[c].isel(T=0, L=l).sel(X=slice(region_coords[r][
                                                                      region_coords[r]['ea
                                                                 Y=slice(region_coords[r]['
                                                                      region_coords[r]['
                                    cross_dateline=True,
                                    ocean=False,
                                    title= region_of_interest[r] + ' Initialized ' + initial_month_na
                                    savefig=os.path.join(figure_dir, '_'.join(['im' + initial_month_
```

NOTE – set ocean=True if you want to mask out the ocean data and plot it as blue, or False if you want to see the whole ocean mask outside of the island features



Summarizing findings…

Consider the questions you will want to assess, make a powerpoint to aggregate your group's results -

(1) What does the **analysis** look like spatially when you run CCA across your 4 test cases?
- EOF loadings for X (predictors)
- EOF loadings for Y (predictands)

(2) How does CCA **compare** when applied using the **different training options**?
(3) How does CCA **compare as lead time increases**?
(4) How does CCA **compare seasonally** (if you have time to run multiple seasons)

slido

Please download and install the
Slido app on all computers you use

Session Recap
Join at slido.com
#2316093

ⓘ Start presenting to display the joining instructions on this slide.

slido

Please download and install the
Slido app on all computers you use

Which bias correction technique generally worked best for your region?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

How did increasing the lead time between your model initialization and target prediction period affect your forecast's performance?

ⓘ Start presenting to display the poll results on this slide.

slido

Please download and install the Slido app on all computers you use

In which 3-month periods that you tested did you get the best performance?

ⓘ Start presenting to display the poll results on this slide.