Master of Applied Data Science

DATA601

Applied Data Science Project

February, 2019 – May, 2019

Application of a Machine Learning Algorithm，Self-Organizing Map, for Examining the consistency between atmospheric reanalyses about near-surface horizontal wind fields

Shun Li

Key points:

atmospheric reanalysis, Self-Organizing Map, python, EOF, ERA-Interim, ERA20C, Patterns visualization, contingency table analysis, chi-suqare, entropy

Table of Contents

Abstract

This project compares statistics of near-surface winds in two reanalyses, namely ERA-Interim and ERA20C datasets. This comparision uses daily scale zonal (East-West) and meridional (North-South) winds classified against 12 representative patterns derived using a Self-Organizing Map applied to ERA-Interim data. Changes in occurrence of representative patterns are collected into contingency tables for each region and each year. Measurements of chi-square and Entropy derived from the contingency tables are used to determine the consistency between ERA-Interim and ERA20C.

1. Introduction & motivation (problem definition)

In climate science, atmospheric reanalyse gives a numerical description of the recent atmospheric state, by combining models with a wide range of observations(Compo,2011).They contain estimates of atmospheric parameters such as air temperature, pressure and wind at different altitudes. The estimates are produced for all locations on earth, and they span decades or more. So, atmospheric reanalyses are large datasets that can take up several petabytes of space.

However, the atmosphere strongly interacts with the underlying ocean,sea ice, and land surface processes(Xiangdong, 2008) and these interactive processes are not well understood and represented in the models that are used in atmospheric reanalysis.This leads to a problem that the combining models might produce biases and errors in simulating and forecasting the atmospheric state(Xiangdong, 2008).

Furthermore, for different atmospheric reanalyses, because of the different types of observation used, like radiosondes、satellite、buoy、aircraft and ship, the biases and errors also exist in all observational data, including temporal and spatial coverage.

So, taking these two sides together, there can be some differences between various atmospheric reanalyses.

The central focus of this project is the quality of reanalyses over Antarctica, where the weather is harsher, fewer observation points than other areas are available, and the climate change is much more complex, it is very likely that the quality of these existing atmospheric reanalyses are too poor to support some types of atmospheric researches in Antarctica. The figure below shows the global radiosonde network to visualize the gaps of global observations between different areas.

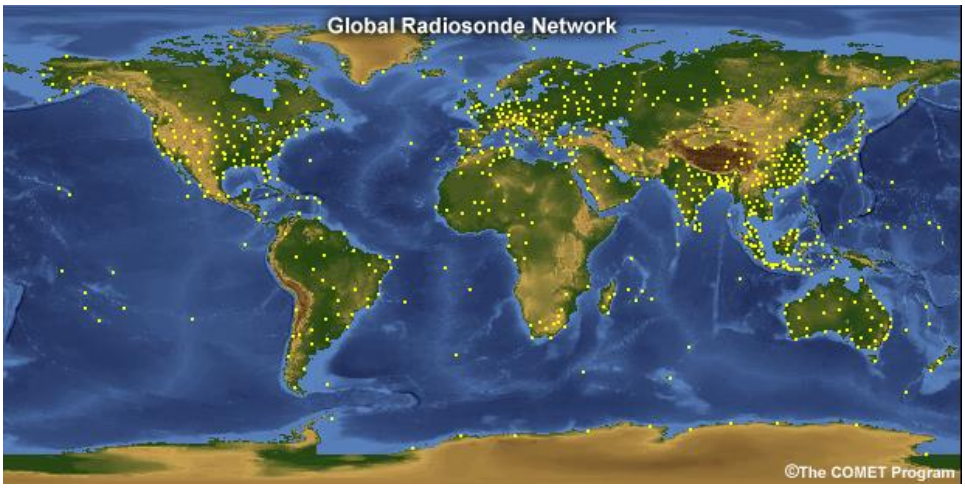


Figure1. Global Radiosonde Coverage – each dot represents a launch point. Map from University of Graz. Note the lack of observation around Antarctica

Figure 1 shows that, the observations in Antarctic are poor. But, it is still just a subjective judgment, several quantitative analyses need to be implemented to evaluate these differences between different regions.

Based on the above factors, Gateway Antarctica, the center for Antarctic studies and research at University of Canterbury, has set up this project, which aims to identify the quality of reanalyses around the earth, and compare the results with work completed around Antarctica already. More specifically, by carrying out his project, these questions could be answered :

- In which regions is the quality or consistency of different atmospheric reanalyses the best or worst?
- Which years is the quality or consistency for different atmospheric reanalyses the best or worst?
- How well do these atmospheric reanalyses characterize the atmospheric state in Antarctic regions, is it bad? And if so, how bad?

On methodology, to simplify the workload of the whole project, this project was to select one single atmospheric parameter, near-surface horizontal wind fields. And, also, this project will apply a new approach from data science to the wind fields. One of them is Self-Organizing-Maps, which is a pattern classification and clustering algorithm. Besides, some traditional statistical methods could be applied like contingency table analysis, which measures the dependency between different reanalysis products. And, one attractive point in this project is using python for the whole process, some common data analytic libraries like numpy、pandas、matplotlib、xarray, and so on. In addition, knowledge about matrix and linear algebra also could be used to solve some multidimensional data(multi-spatio-temporal data).

For innovation, this project introduces modern machine learning algorithms into

atmospheric and climate science, which could enhance the reliability of the results； this project fills in the research on the consistency of atmospheric reanalyses； this project could raise new topics about the quality of reanalyses, like air temperature, air pressure and so on to produce a complete quality analysis.

2. Data and dataset (data acquisition)

Usually, the atmospheric reanalysis products have been divided into three generations (Xiangdong Zhang,2008).

In the first generation, three international centers took initiative in middle 1990s.First,**European Centre for Medium-Range Weather Forecasts(ECMWF)**, it operates one of the largest supercomputer complexes in Europe and the world's largest archive of numerical weather prediction data; then, **National Aeronautics and Space Administration Data Assimilation Office(NASA/DAO)**, it conducts modeling and assimilation activities and uses data gathered by past and current observations to support NASA's current suite of Earth Systems; Third, **National Centers for Environmental Prediction(NCEP)**and **National Center for Atmospheric Research(NCAR)**.Here are the products from them:

--- **ERA-15**(1979-1993) from ECMWF

--- **NASA/DAO**(1980-1993) from USA

--- **NCEP/NCAR**(1948-present)from USA

In the second generation, the following first three centers, **Japan Meteorological Agency(JMA)** conducted the first Asian long-term global atmospheric reanalysis project, it collected and prepared observational and satellite data from many sources including ECMWF, National Climate Data Center(NCDA).In the USA, **Department of Energy(DOE)** cooperated with NCEP to develop the new reanalysis products. Some main products from them:

--- **ERA-40(19598-201)** from ECMWF

--- **JRA-25(1979-2004)** from JMA

--- **NCEP/DOE(1979-presesnt)** from USA

In the third generation, NCEP developed its third product called **Climate Forecast System Reanalysis (CFSR).Global Modeling and Assimilation Office(GMAO)** from NASA created Modern-Era Retrospective analysis for Research and Applications,Version 2(MERRA-2).Following are main products:

--- **ERA-interim(1979-present)** from ECMWF

--- **JRA-55(1958-2012)** from Japan

--- **NASA/GMAO-MERRA-2(1979-present)** from USA

--- **ECEP-CFSRP(1979-2008**) from USA

--- ERA20C(1900-2010) from ECMWF

In this project, the information about wind fields and topography have been selected as

the main research objects. And two datasets，**ERA-interim(1979-present)、ERA20C(1900-2010) have been selected.** The data about north-south velocity and East-west velocity has been downloaded from these datasets. But, one thing to be noticed is that the format for the data is NetCDF, which is an international standard of the Open Geospatial Consortium. In general, a NetCDF dataset contains three description types including dimensions、variables and attributes. Variables store actual data, dimensions give information about the dimensions of variables and attributions provide auxiliary information or metadata about variables. Since a NetCDF dataset has contained all information about data in itself, its format also can be called a "self-describing" data format. One example has been listed to explain this in Figure2:

```
topography_data_0
<xarray.Dataset>
Dimensions:  (lat: 91, lon: 180, time: 1)
Coordinates:
  * lat      (lat) float32 90.0 88.0 86.0 84.0 82.0 ... -84.0 -86.0 -88.0 -90.0
  * lon      (lon) float32 0.0 2.0 4.0 6.0 8.0 ... 350.0 352.0 354.0 356.0 358.0
  * time     (time) datetime64[ns] 1851-01-01
Data variables
    hgt      (time, lat, lon) float32 ...
Attributes:
    Conventions:            CF-1.2
    title:                  4x Daily NOAA-CIRES 20th Century Reanalysis V2c
    comments:               Data is from \nNOAA-CIRES 20th Century Reanalysi...
    platform:               Model
    institution:            NOAA ESRL Physical Sciences Division & CU/CIRES ...
    version:                2c
    history:                created 2015/02 by Hoop (chunked, deflated non-p...
    contact:                esrl.psd.data@noaa.gov
    source:                 20CRv2c 2014, Ensemble Kalman Filter, ocean (spe...
    forcing:                CO2, Sl, Vl, SST, Oz (Oz is prognostic, SODAsi.2...
    forcing_note:           Additional information on the external forcings ...
    name_of_model_used:     ensda_v351
    creation_date:          2014-06-13T18:30:07Z
    product:                reanalysis
    observations:           International Surface Pressure Databank version ...
    assimilation_algorithm: Ensemble Kalman Filter
    dataset_title:          NOAA-CIRES 20th Century Reanalysis (V2c)
    citation:               Compo,G.P. <https://www.esrl.noaa.gov/psd/people...
    References:             https://www.esrl.noaa.gov/psd/data/gridded/data...
```

Figure2. topography data in python xarray open

About topography data, it has three **dimensions**(latitude、lontitude、time)，and **variable** is height, it has many different **attributes** like title、version、source and so on.So, actually, each NetCDF data is just a multidimensional array.

And,in python,using xarray library, which can be used to analysis the Multidimensional array,to open the NetCDF file.

3. EOF / PCA (data cleaning and processing)

3.1 Introduction

Empirical Orthogonal Function technique(EOF),also referred as eigenvector Analysis, or principal component analysis(PCA) is used in the analysis applied. It is a method to analyze

the structure feature of matrix data and extract the main data features (A. Hannachi,2004). In other words, it aims at finding a new set of variables that capture most of the observed variation in the data. EOF was first proposed by the statistican, Pearson in 1902. And it has been introduced in atmospheric science since the 50's by Lorenz. But, because it was very computationally demanding, it was not widely used in practical work until 1960's.

In EOF analysis, the eigenvectors correspond to the spatial samples, and it reflects the spatial distribution of factor fields (in this project, the factor field is wind field); the principal components correspond to the time series, and reflects the changes of weights about each spatial mode. So, often, EOF analysis is also called the Decomposition of Time and Space, that is:

$$\boldsymbol{X = EOF_{m*m} \times PC_{m*n}}$$

m is spatial point, *n* is the length of time series

3.2 Principle of EOF

In this project, the observed data of wind field is given as a matrix form:

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ & \cdots & \cdots & \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$

The purpose of EOF analysis is to decompose $\boldsymbol{X}$ as time function $\boldsymbol{Z}$ and space function $\boldsymbol{V}$:

$$\boldsymbol{X = VZ}$$

Or:

$$x_{it} = \sum_{k=1}^{p} v_{ik} z_{kt} = v_{i1} z_{1t} + v_{i2} z_{2t} + \cdots + v_{ip} z_{pt}$$

$$i = 1,2,\cdots,m \qquad t = 1,2,\cdots,n \qquad k = 1,2,\cdots,p$$

And it means that the *tth* observation value of *ith* grid point can be regarded as a linear combination of $\boldsymbol{P}$ space function v_{ik} and time function z_{ki}. So,

$$\boldsymbol{V} = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1m} \\ v_{21} & v_{22} & \cdots & v_{2m} \\ & \cdots & \cdots & \\ v_{m1} & v_{m2} & \cdots & v_{mm} \end{pmatrix} \qquad \boldsymbol{Z} = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ & \cdots & \cdots & \\ z_{m1} & z_{m2} & \cdots & z_{mn} \end{pmatrix}$$

and $v_j = (v_{1j}, v_{2j}, \cdots, v_{mj})^T$ is a space typical field.

Based on above, the *tth* space field:

$$\begin{bmatrix} x_{1t} \\ x_{2t} \\ \vdots \\ x_{mt} \end{bmatrix} = \begin{bmatrix} v_{11} \\ v_{21} \\ \vdots \\ v_{m1} \end{bmatrix} z_{1t} + \begin{bmatrix} v_{12} \\ v_{22} \\ \vdots \\ v_{m2} \end{bmatrix} z_{2t} + \cdots + \begin{bmatrix} v_{1m} \\ v_{2m} \\ \vdots \\ v_{mm} \end{bmatrix} z_{mt}$$

or $\boldsymbol{x}_t = \boldsymbol{v}_1 z_{1t} + \boldsymbol{v}_2 z_{2t} + \cdots + \boldsymbol{v}_m z_{mt}$

3.3 Algorithm calculation steps

(1).Selecting the data, and preprocessing data to obtain the data matrix:

In this project, the data we have is a wind field(North-South field & East-West field). It is the value of the wind speed in each grid point examined and is recorded each day. So, firstly, because it has been collected for many years, we need to calculate the mean of each day, which is called the "climatology" in code. Then, calculating the difference between original data and "climatology".

However, this project is about near-surface horizontal wind field, which means that an altitude limit needs to be set. In this project, we decided to use only the data point where the topography height is below 500m as this is known to be more accurate. So, here, one job we need to do is data masking to remove the grid points where topography is above the limit(500m).

Finally, concatenating the North-South wind field and East-West wind field together, which is the matrix $\boldsymbol{X}$ in the formula.

(2).Computing the cross product of *X* with its transpose:

According to the knowledge from linear algebra, we know that

$$\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{V}\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{V}^T$$

So, here the cross product is $\boldsymbol{A}$,which satisfies $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{X}^T$

(3).Computing the eigenvalues $(\lambda_1, \ldots\ldots, \lambda_m)$ **and eigenvectors** V_{m*m} **of *A*. And satisfying** $A_{m*m} \times V_{m*m} = V_{m*m} \times E_{m*m}$ **, and *E is a diagonal matrix* :**

$$E = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_m \end{bmatrix}$$

(4).Arranging the eigenvalues from the largest ones to the smallest ones.

Because $\boldsymbol{X}$ is the real observation, so the related λ mustn't be less than 0.And each eigenvalue corresponds to an eigenvector, which is $\boldsymbol{V}$ that we want in the beginning.

$$V = v_1, v_2, \cdots, v_m$$

(5).Computing principal components *Z*:

By projecting V onto the original data X, the corresponding time coefficients of all eigenvectos,namey the principal components, can be obtained.

$$Z = V^T X$$

(6).Computing contribution rate and removing noisy that covers less important information:

In the python code, it is the step 9 (Truncating datasets) in pre-processing data part. Usually, if λ is larger, it is more important.

$$\frac{\lambda_k}{\sum_{i=1}^{m} \lambda_i} \times 100\%$$

In this project, the threshold is set as 0.9, which means we just need the first 90 percent of λ.

4. SOM (data modeling)

4.1 Introduction

In this project, we use the Self-Organizing-Map(SOM) technique on ERA-interim output between 1979-2015 to derive representative surface wind patterns, output from other reanalysis are then attributed to one of these patterns using a Euclidean distance metric.

A Self-Organizing-Map(SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional),discretized representation of the input training samples, called a map. This method was introduced by the Finnish professor Teuvo Kohonen in the 1980s, so sometimes, it is also called as Kohonen map (kohonen,T, 1990). Self-Organinzing-Map differs from other artificial neural networks as it applies competitive learning, which will be explained later, and in the sense that it uses a neighborhood function to preserve the topological properties of the input training samples.

In deep learning, some common algorithms, like artificial neural network、convolutional neural network、K-means and so on, have been familiar for us. Here, the differences between different algorithms has been listed to provide a brief review about Machine Learning:

Supervised Learning	Artificial Neural Network	Used for Regression & Classification
	Convolutional Neural Networks	Used for Computer Vision
	Recurrent Neural Networks	Used for Time Series Analysis
	Support Vector Machines	Used for Regression & Classification
Unsupervised Learning	Self-Organizing Maps	Used for pattern Detection
	K-means clustering	Used for cluster analysis
	Deep Boltzmann Machines	Used for Recommendation Systems

Figure3. Comparisons between some common machine learning algorithms

4.2 K-means clustering

In this project, even though we have not used k-means clustering algorithm, it is extremely helpful to understand Self-Organizing Maps. And since they are both clustering algorithms, some similar logical thinking exists between them.

In data mining, we use K-means to categorize datasets. First, before processing the dataset into categories, the number of clusters has to be agreed upon; Then, randomly select K points to act as centroids; After, assign each data point to the centroid closest to it, just to reminding, here Euclidean distances has been chosen to measure distances; Next, compute and place the new centroid of each cluster; Reassign the data point to the new centroid that is closest to them; Repeat the steps until getting the ideal state.

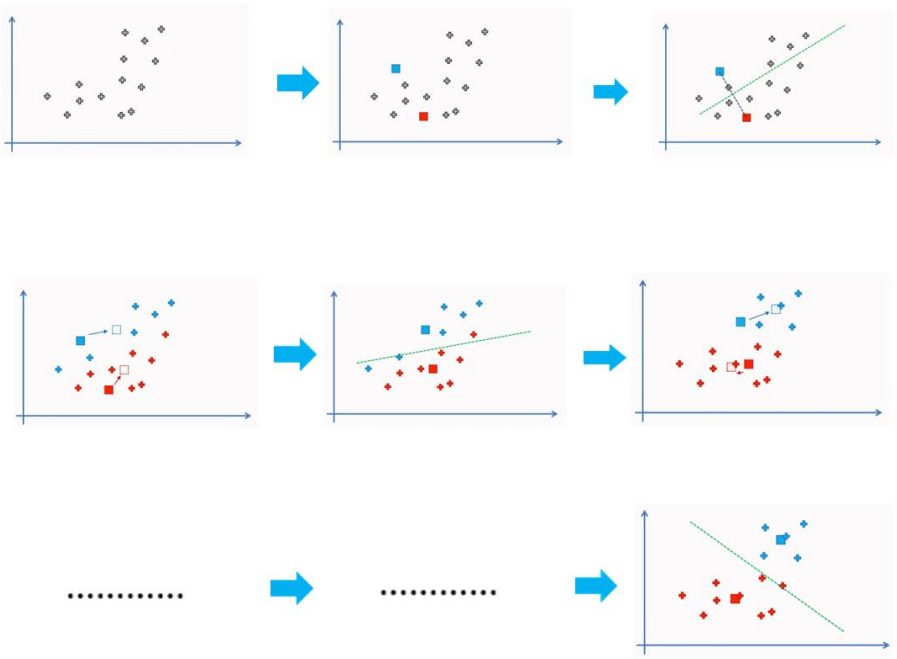


Figure4. K-means clustering algorithms

4.3 SOM algorithm principle

4.3.1 weights and best-matching-unit(BMU) in SOM

Frist, we will start by giving an example of how this algorithm works. Assuming that we have three input nodes that represent three columns(dimensions) in dataset, and each of these columns can contain thousands of rows.

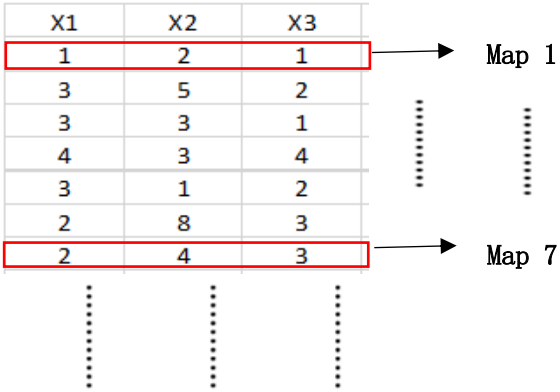


Figure5. Example about SOM maps

And in this example, 9 different output nodes have been selected. However, in this project, the number of output nodes was identified as 12, which is based on experience with pattern recognition methodology in this climatological context. Since one of the benefits with Self-Organizing Maps is reducing dimensionality, the output nodes are two-dimensional.

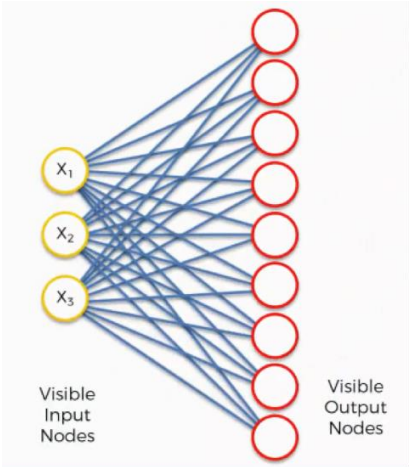


Figure6. SOM visible input nodes and output nodes

Then, if we take the topmost output node and focus on its connections with the input

nodes. As we can find, a weight has been assigned to each of these connections.

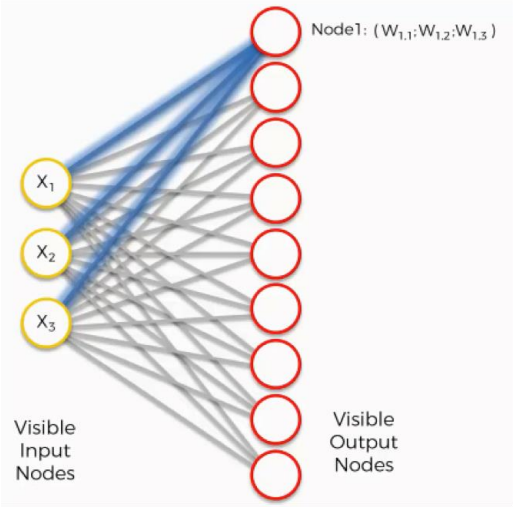


Figure7. Assigning weights into SOM topmost output node

However, the meaning of weight is total different with those weights used in artificial or convolutional neural networks. In artificial neural networks, multiplying the input node's value by the weight and applying an activation function ($\boldsymbol{Y} = \sum Weight * input) + bias$). But, on the contrary, with SOM, no activation functions. The weights are not separate from the nodes ,they belong to the output node itself. Instead of being the result of adding up these weights, each output node in an SOM contains the weights as its corrdinates.

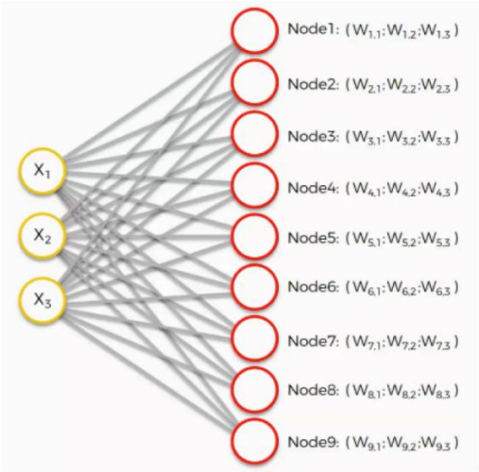


Figure8. Assigning weights into SOM output nodes

Next, using the following equation to calculate the closest node with map1 (row 1):

$$\textbf{Distance} = \sqrt{\sum(\mathbf{x_i} - \mathbf{w_{j,i}})^2}$$

i : input node number **j** : output node number

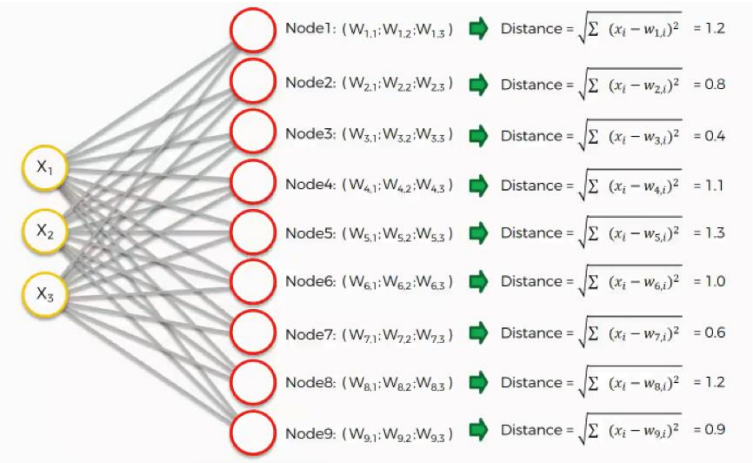


Figure9. Calculating distances for SOM output nodes

In this example, node3 is the closet one with the distance of 0.4.And this node is called as **best-matching-unit(BMU**) in Self-Organizing Maps.

4.3.2 Dragging the BMU closer to data point

In the last section, we used 9 output nodes, then we use a lager SOM to understand how to adjust the weights to obtain the BMU. In this section, we consider 9 x 12 output nodes. Supposedly, the green circle represents the map1 ’s BMU. But because the input nodes cannot be updated since they are the fixed dimensions of data, whereas we have to update the weights so that the output node is still the closest one to the map1 in dataset. So, in simple terms, SOM is drawing closer to the data point by stretching the BMU towards it.

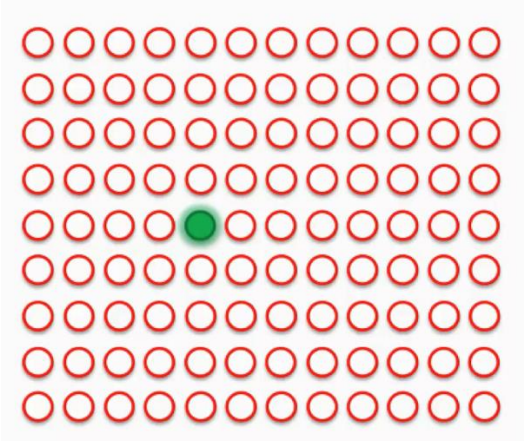


Figure10. SOM best-matching-unit(BMU)

As we can see in Figure 10, the BMU(green circle) is the closest one to the data point(the small white circle). But each time, when we drag the BMU closer to the data point, the nearby nodes are also pulled closer to that point.

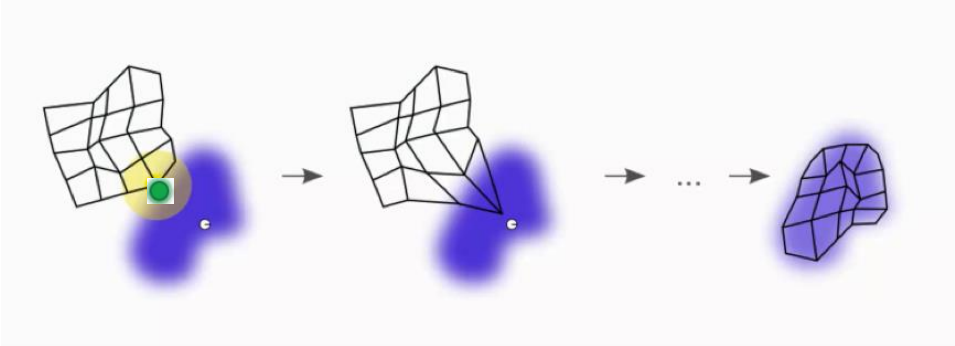


Figure11. Adjusting SOM best-matching-units

To determine which nearby nodes need to be adjusted, a radius around the BMU need to be drawn, and any node that falls into that radius would have its weight updated to pulled it closer to the data point that it has been compared against. In addition, the closer a node is to the BMU, the heavier the weight that will be added to its update. Supposedly, in this dataset, it contains multiple best-matching units, and each of them will be assigned a radius like below.

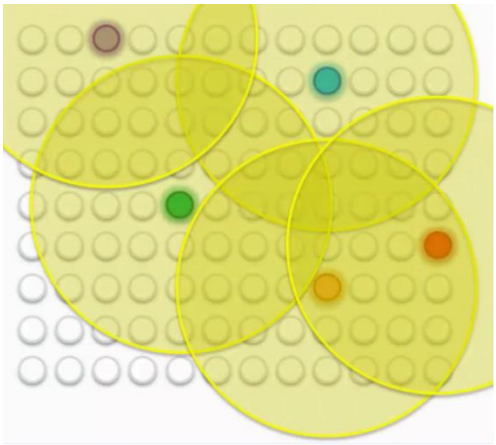


Figure12. Assigning each BMU with a radius

Each time, when updating one BMUs, the peripheral nodes are going through some push and pull process because initially many of them fall within the radius of more than one BMUs. They are updated by the combine forces of these BMUs, with the nearest BMU being the most influential. As the process is repeated, the radius shrinks and this means that the SOM patterns are most strongly affected early on in the process. This mechanism of a radius also ensures that the nodes that are most alike are clustered together.

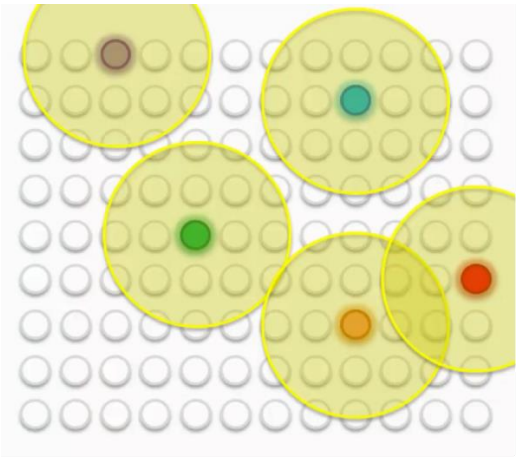


Figure13. changing the size of BMU' radius

After all the push and pull between the nodes and the different BMUs, we have come to a point where each node has been assigned a BMU.

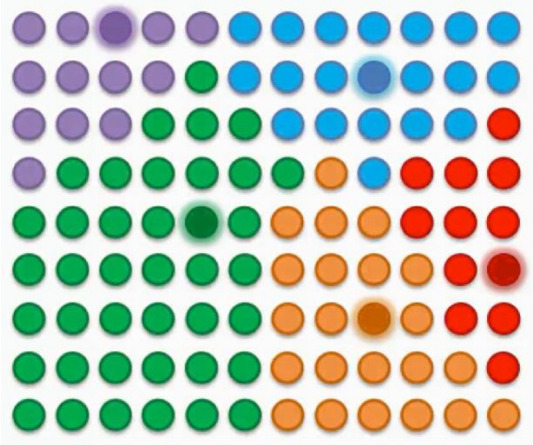


Figure14. creating SOM patterns

4.3.3 Patterns visualization

In order to determine the consistency of surface winds between ERA-interim and ERA20C globally, we have to intercompare these datasets. Before applying Self-Organizing Maps algorithm, we need to split the whole world into different regions. In this project, the whole world has been divided into 36 equal-areas. The latitude has been divided into 6 pieces from 90°N to 90°S by 30 degrees, and the longitude also has been divided into 6 pieces by 60 degrees with the 20°E as the starting point.

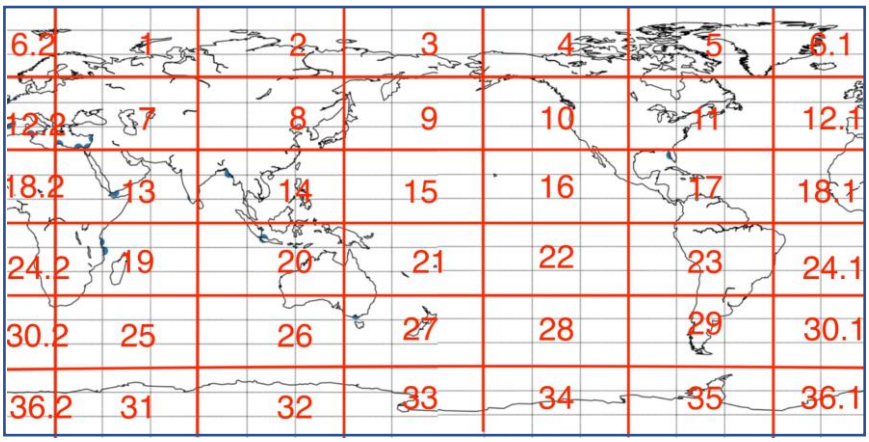


Figure15. segmentation maps with 36 equal-size regions

In this project, we used the SOMPAK(v 3.1) software(obtained from http://www.cis.hut.fi/research/som_pak) to derive the SOM from the most important PCs (related to top 90% of variance) over the period 1979-2015. And the learning process adjusts a set of reference vectors based on the difference between the reference vector and each input record. A learning rate determines how the adjustment is related to the difference between the reference vector and input data. Training then consists of many iterations of reference vector adjustment until stable values are reached. In each iteration, the best matching reference vector is identified, using the Euclidean distance.

Based on past experience, 12 representative patterns (or output nodes) for the wind fields has been displayed by application of the SOM technique to ERA-Interim output, corresponding zonal(East-West) and meridional (North-South) wind patterns are displayed in the supplementary information. A 3 x 4 SOM was selected as it minimized quantization error and represented a good balance in terms of representation of wind patterns over different regions.(See Figure 16 and 17).

Because there are 36 different regions, so the related 36 SOM has been obtained. Here we just pick one SOM, region35(Antarctic Peninsula) for an example, from the result to give a brief explanation.

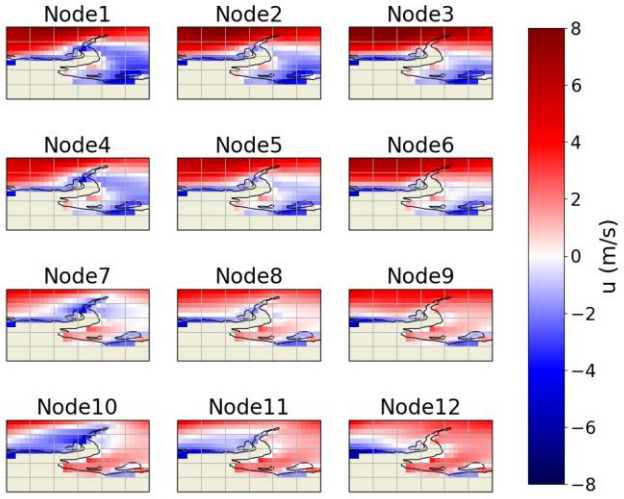


Figure16. surface zonal winds in region35 for each node in SOM derived from ERA-Interim reanalysis output

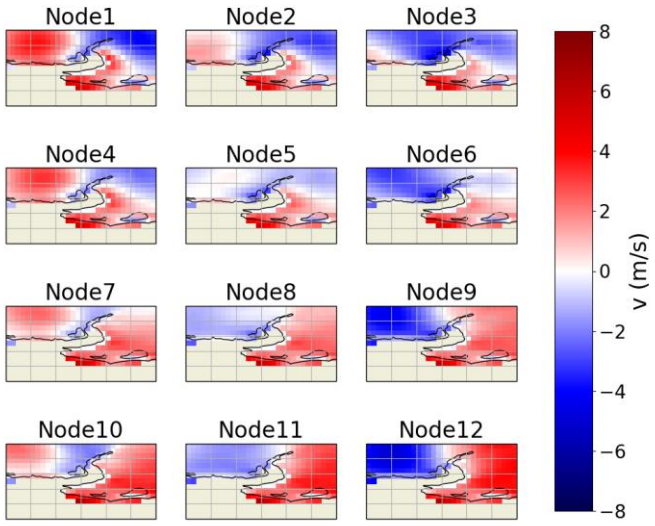


Figure17. surface meridional winds in region35 for each node in SOM derived from ERA-Interim reanalysis output

In region35(Antarctic Peninsula), all results indicate that the zonal (East-West) velocity is strongest at the top the Antarctic peninsula(See Figure 16). From Node 1 to Node 6, the velocity is weakest in Ronne Ice Shelf, and from Node7 to Node 12, the result is on the contrary; all results indicate that meridional (North-South) velocity is strongest in

Ronne Ice Shelf, especially the part that is closest to the contient. Node 9 and 12 show that in Belling-Shausen sea, the meridional (north-south) velocity is weakest.

5. Model adjustment

After initial work, it was necessary to improve the analsysis by completing some model adjustements. Before, we split the whole world into 36 different region boxes and applied the Self-Organizing Maps algorithm to each of them to obtain the related patterns. And each region boxes size is 30 degrees latitude by 60 degrees longitude. But, each region boxes has no overlapping points with each other, which may cause random variations for the results of each pattern. So, we decided to offset the longitude so that the region boxes overlap and the longitude also overlap to avoid this randomness. We lose independence between regions, but this is offset by having a higher resolution spatial analysis which might provide interesting detail. In addition, we also needed to ensure that the discontinuity at 360 degrees longitude was removed from our previous analysis.

Since each region box size is 60 degrees longitude, it means each region just moves 6 times and it can offset with the next region box to form a cycle in whole 360 degrees longitude. And only 6-times movements can make sure the discontinuity.

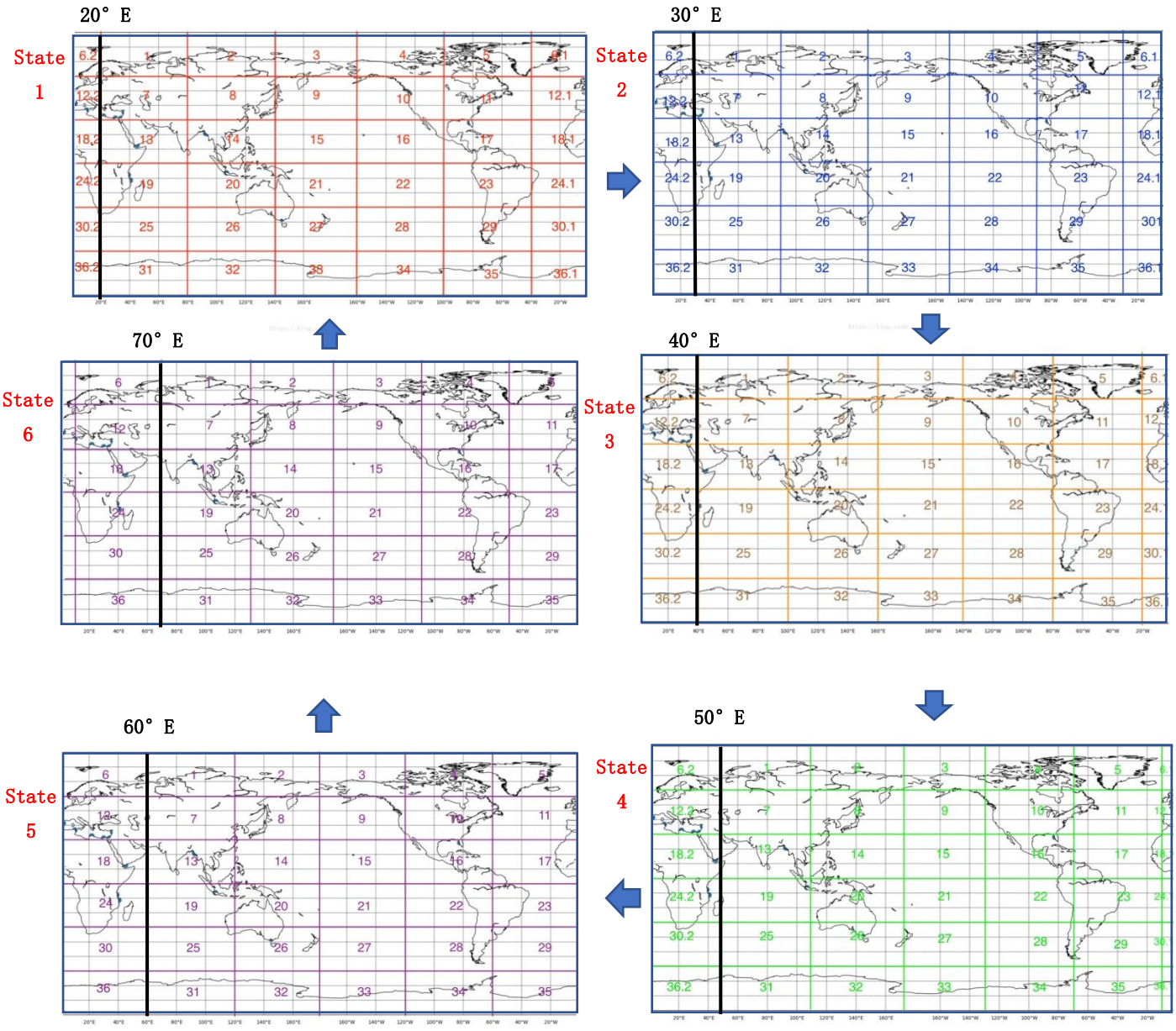


Figure18. Model adjustment to create the continuity and circulation

6. Evaluation

6.1 Histogram

Until now, we have got all of the results for each pattern of each region in the above 6 different states. The next step is to calculate the time series of BMU using a Euclidean distance measure for ERA20C and ERA-Interim data respectively. Here, listing the result for region1 in state1 from ERA-Interim data in Figure 19.

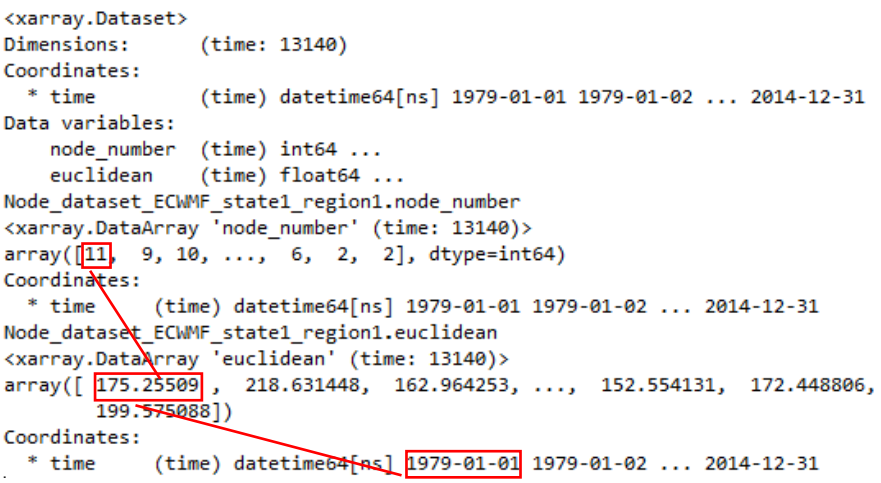


Figure19. Node dataset for ERA-Interim about region1, state1 in python xarray open

From this example, we can know that for region1 in state1,the pattern node number that each day belongs to and the related Euclidean distance for it. For instance,1979-01-01, it belongs to node 11 and the Euclidean distance for its BMU is 175.25509.

Then, the next thing is that we need to statistic: for each node, how many days in each year belong to? and also its proportion? Academically, this proportion is called relative frequency of occurrence. In this project, we use the **np.histogram** function to create the histogram results of the relative frequency of occurrence, and the result has been saved in the file---3.1 histogram. Since the format of result is netCDF, it is not very easy to browse all the result, so I was introduced to a tool called **panoply** (obtained from https://www.giss.nasa.gov/tools/panoply/), which is a cross-platform application designed by NASA. It can plot geo-referenced and other arrays from netCDF, HDF, GRIB and other datasets.Using this tool, I can plot some pieces of data for region1, state1 for ERA-Interim for example(See Figure 20):

X Axis: node_number ()

Y Axis: time ()	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
1979.0	78	32	37	32	13	34	41	5	24	32	14	23
1980.0	51	22	31	34	26	26	36	22	30	39	19	29
1981.0	59	14	35	34	14	14	37	11	25	69	27	26
1982.0	62	35	40	24	14	26	30	12	28	43	22	29
1983.0	56	21	34	39	19	31	29	15	26	36	24	35
1984.0	82	18	30	47	7	28	39	13	15	41	20	25
1985.0	68	23	37	40	17	31	29	11	32	37	16	24
1986.0	60	26	31	45	16	13	38	10	26	57	20	23
1987.0	59	15	27	31	18	32	37	21	26	50	17	32
1988.0	43	16	21	30	13	24	39	21	26	78	34	20
1989.0	39	20	39	36	10	16	44	15	20	74	17	35
1990.0	95	17	23	45	11	13	28	18	19	56	16	24
1991.0	42	27	29	46	7	20	34	7	15	83	29	26
1992.0	71	21	31	36	9	19	35	15	11	63	20	34
1993.0	45	13	33	41	15	20	40	18	25	59	28	28

Figure20. Frequency of occurrences about region1, state1, ERA-Interim in panoply open

From the result, we can get the information about the number of each nodes in each year. For example, in year 1979, 78 days belonged to node1, which have the largest proportion, meanwhile, only 5 days belong to node8, which is the minimum one. Also, according to the result, the relative frequency of occurrence of the different patterns can be easily obtained:

X Axis: node_number

Y Axis: time	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
1979.0	21.37	8.77	10.14	8.77	3.56	9.32	11.23	1.37	6.58	8.77	3.84	6.30
1980.0	13.97	6.03	8.49	9.32	7.12	7.12	9.86	6.03	8.22	10.68	5.21	7.95
1981.0	16.16	3.84	9.59	9.32	3.84	3.84	10.14	3.01	6.85	18.90	7.40	7.12
1982.0	16.99	9.59	10.96	6.58	3.84	7.12	8.22	3.29	7.67	11.78	6.03	7.95
1983.0	15.34	5.75	9.32	10.68	5.21	8.49	7.95	4.11	7.12	9.86	6.58	9.59
1984.0	22.47	4.93	8.22	12.88	1.92	7.67	10.68	3.56	4.11	11.23	5.48	6.85
1985.0	18.63	6.30	10.14	10.96	4.66	8.49	7.95	3.01	8.77	10.14	4.38	6.58
1986.0	16.44	7.12	8.49	12.33	4.38	3.56	10.41	2.74	7.12	15.62	5.48	6.30
1987.0	16.16	4.11	7.40	8.49	4.93	8.77	10.14	5.75	7.12	13.70	4.66	8.77
1988.0	11.78	4.38	5.75	8.22	3.56	6.58	10.68	5.75	7.12	21.37	9.32	5.48
1989.0	10.68	5.48	10.68	9.86	2.74	4.38	12.05	4.11	5.48	20.27	4.66	9.59
1990.0	26.03	4.66	6.30	12.33	3.01	3.56	7.67	4.93	5.21	15.34	4.38	6.58
1991.0	11.51	7.40	7.95	12.60	1.92	5.48	9.32	1.92	4.11	22.74	7.95	7.12
1992.0	19.45	5.75	8.49	9.86	2.47	5.21	9.59	4.11	3.01	17.26	5.48	9.32
1993.0	12.33	3.56	9.04	11.23	4.11	5.48	10.96	4.93	6.85	16.16	7.67	7.67

Figure21. Relative Frequency of occurrences about region1, state1, ERA-Interim in panoply open

6.2 Contingency Table

Now, for each region in each state, we have the all the frequency of occurrences for each year. For ERA-Interim dataset, the result of frequency of occurrences is from 1979 to 2015; For ERA20C dataset, the result of frequency of occurrences is from 1900 to 2010.The common years that they both have is form 1979 to 2010. So, based on this, we can make quantitative

measure of the strength of associations for both this two distributions.

The best approach for this is contingency tables. In this project, using **pandas.crosstab** to catch the result. All the result has been produced and saved in file--- 4.1 contingency_table. The logic and usage can be found in crosstab cheatsheet from appendix**.** Actually, we have got 6480 different contingency tables (30 common year * 36 regions * 6 states). Due to the limitations of the article, I just pick one year (1980) from the specific region(region1) in specific state(state1) to give an explanation:

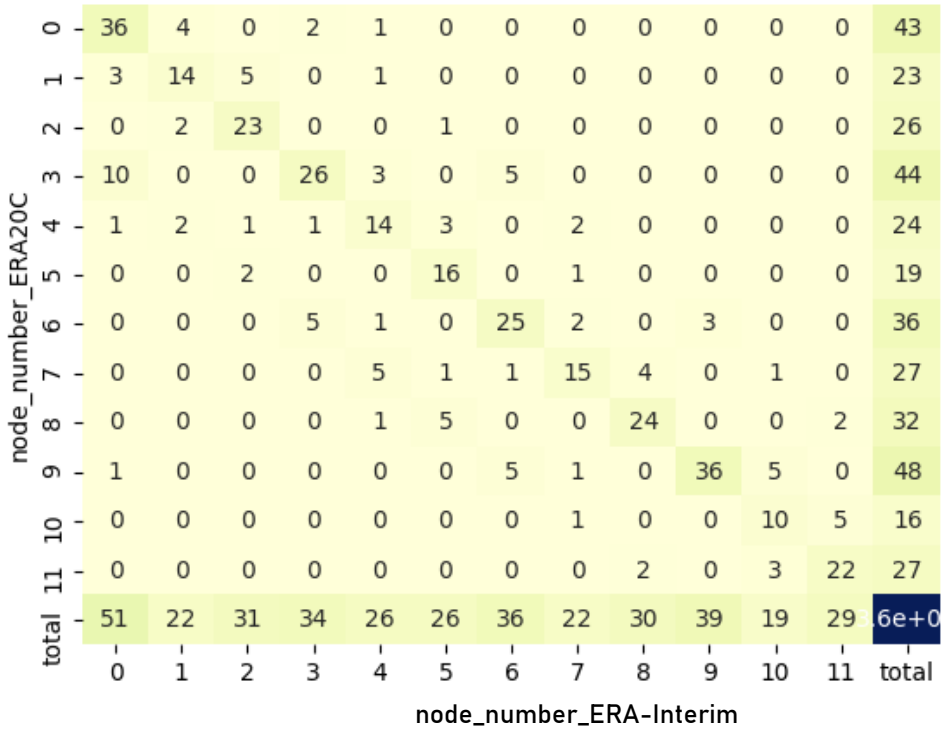
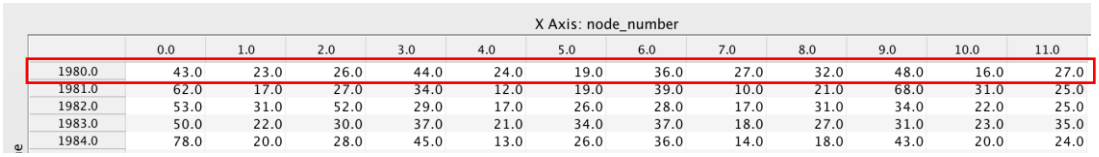


Figure22. contingency table about region1, state1 in 1980

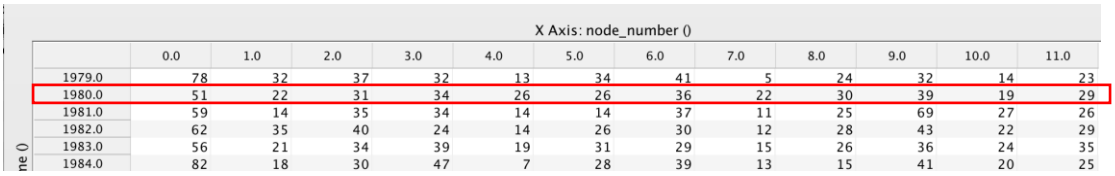
In the contingency table, the rows are labeled by the values of ERA20C, the columns are labeled by the values of ERA-Interim, and the entries are nonnegative integers giving the number of days that both ERA20C and ERA-Interim have. So, from the above result, we can know that total 36 days in 1980 from both ERA-Interim and ERA20C belong to node1, 14 days are node2 and so on. Also, for ERA-Interim, total 51 days belong to node1 and 22 days belong to node2 and so on; for ERA20C, 43 days belong to node1 and 23 days belong to node2 and so on. To verify the correctness of the result, we can compare it with results of the frequency of occurrences that we have calculated before. frequency of occurrences.



X Axis: node_number

	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
1980.0	43.0	23.0	26.0	44.0	24.0	19.0	36.0	27.0	32.0	48.0	16.0	27.0
1981.0	62.0	17.0	27.0	34.0	12.0	19.0	39.0	10.0	21.0	68.0	31.0	25.0
1982.0	53.0	31.0	52.0	29.0	17.0	26.0	28.0	17.0	31.0	34.0	22.0	25.0
1983.0	50.0	22.0	30.0	37.0	21.0	34.0	37.0	18.0	27.0	31.0	23.0	35.0
1984.0	78.0	20.0	28.0	45.0	13.0	26.0	36.0	14.0	18.0	43.0	20.0	24.0

Figure23. Checking results from contingency table with panoply open about region1, state1 in 1980 in ERA20C datasets



X Axis: node_number ()

	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
1979.0	78	32	37	32	13	34	41	5	24	32	14	23
1980.0	51	22	31	34	26	26	36	22	30	39	19	29
1981.0	59	14	35	34	14	14	37	11	25	69	27	26
1982.0	62	35	40	24	14	26	30	12	28	43	22	29
1983.0	56	21	34	39	19	31	29	15	26	36	24	35
1984.0	82	18	30	47	7	28	39	13	15	41	20	25

Figure24. Checking results from contingency table with panoply open about region1, state1 in 1980 in ERA-Interim datasets

By comparison, the results of the two approaches are the same. So, the result of the contingency table is correct definitely.

6.2.1 chi-square

Until now, all the 6480 different contingency tables have been calculated. The next job is how to make measures of Associations between ERA20C and ERA-Interim according to these contingency tables.

One approach is Chi-Square(David C. Howell,2014), which is a basic method in statistic to make measure about different variables.

	node_number_ERA-Interim													
node_number_ERA20C		0	1	2	3	4	5	6	7	8	9	10	11	**Total**
	0													*N1.*
	1													*N2.*
	2													*N3.*
	3													*N4.*
	4													*N5.*
	5													*N6.*
	6													*N7.*
	7													*N8.*
	8													*N9.*
	9													*N10.*
	10													*N11.*
	11													*N12.*
	Total	*N.1*	*N.2*	*N.3*	*N.4*	*N.5*	*N.6*	*N.7*	*N.8*	*N.9*	*N.10*	*N.11*	*N.12*	*N*

Figure25. example about contingency table

Some notation first:

Let ***Nij*** denote the number of events that occur with the first variable ***x*** taking on its ***ith*** value, and the second variable ***y*** taking on its ***jth*** value.

Let ***N*** denote the total number of events, the sum of all the ***Nij***.

Let ***Ni·*** denote the number of events for which the first variable ***x*** takes on its ***ith*** value regardless of the value of y;

Let ***N·j*** is the number of events with the ***jth*** value of y regardless of ***x***.

Then, we have:

$$N_{i\cdot} = \sum_{j} N_{ij} \qquad N_{\cdot j} = \sum_{i} N_{ij}$$

$$N = \sum_i N_{i.} = \sum_j N_{.j}$$

The null hypothesis is that the two variables ***x*** and ***y*** have no association. In this case, the probability of a particular value of ***x*** given a particular value of ***y*** should be the same as the probability of that value of ***x*** regardless of ***y***. Therefore, in the null hypothesis, the expected number for any ***Nij*** , which we will denote ***nij*** ,can be calculated from only the row and column totals,

$$\frac{n_{ij}}{N_{.j}} = \frac{N_{i.}}{N}$$

Which implies $n_{ij} = \frac{N_{i.}\ N_{.j}}{N}$. The chi-square statistic is now given ,which is summed over all entries in the table:

$$x^2 = \sum_{i,j} \frac{(N_{ij} - n_{ij})^2}{n_{ij}}$$

And it can implemented in the python code.

Then we need to find some reparametrization of ***χ2*** which maps it into some convenient interval, like 0 to 1, where the result is not dependent on the quantity of data, but rather depends only on the underlying population from which the data were drawn. Here, we introduce ***Cramer's V***(William H. Press & Brian P.Flanner,1992).

$$\mathrm{V} = \sqrt{\frac{x^2}{N_{mim(I-1,\ J-1)}}}$$

Cramer's V has the pleasant property that it lies between zero and one inclusive: Equals zero when there is no association; Equals one only when the association is perfect.

6.2.2 entropy

Another way to measure the association is based on Entropy. So, what is Entropy? To help to understand it, let us play a game: Imagining two machines, they both output messages from an alphabet of **A**,**B**,**C**,**D**. Machine A generates each symbol randomly, they all occur 25% of the time; Machine B generates symbols according to the following probabilities (A;50%;B:12.5%;C:12.5%; D:25%).

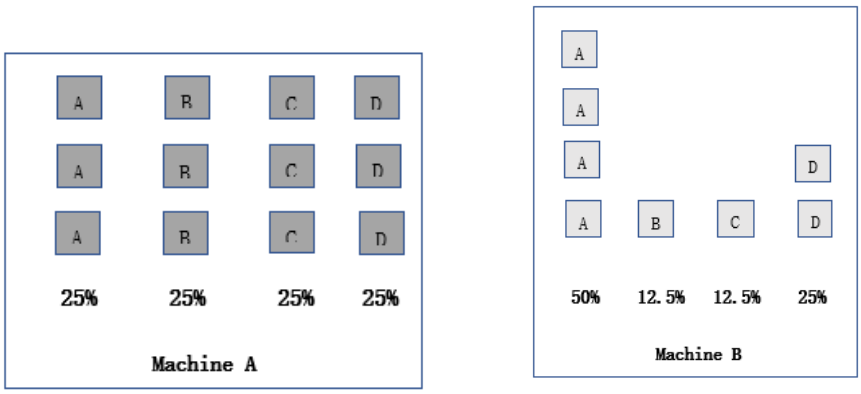


Figure26. two machines models with different--probabilities symbols

The question is that: which machine is producing more information? Claude Shannon cleverly rephrase the question: If we have to predict the next symbol from each machine, which is the minimum number of yes-or-no questions to ask?

For machine A, the most efficient way is to pose a question which divides the possibilities in half. Here, the minimum question number is two.

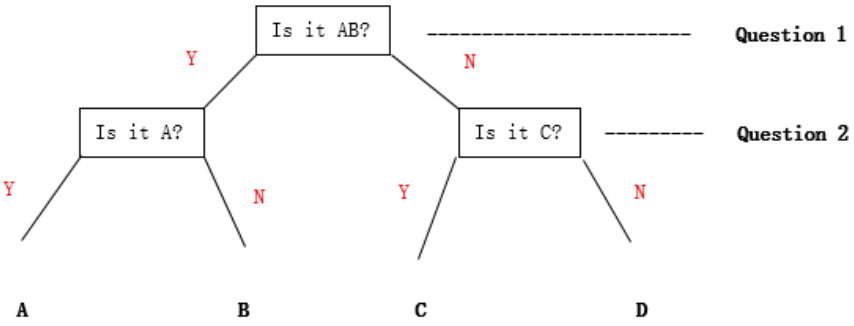


Figure27. Asking questions for machine A

For machine B, usually the symbols which have the maximum probabilities could be asked firstly. Here, the efficient way has been put:

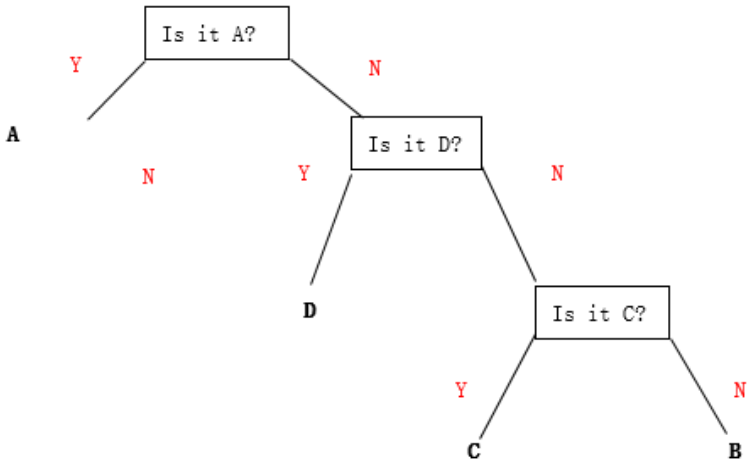


Figure28. Asking questions for machine B

On average, the number of questions that be asked to determine a symbol can be explained nicely with an analogy. In one of two equally likely directions, based on which way it falls, we can generate a symbol. So, machine A, two levels or bounces need to be added; for machine B, three levels or bounces need to be added. Taking a weighted average, the excepted number of bounces:

bounces = PA * 1 + PB * 3 + PC * 3 + PD * 2

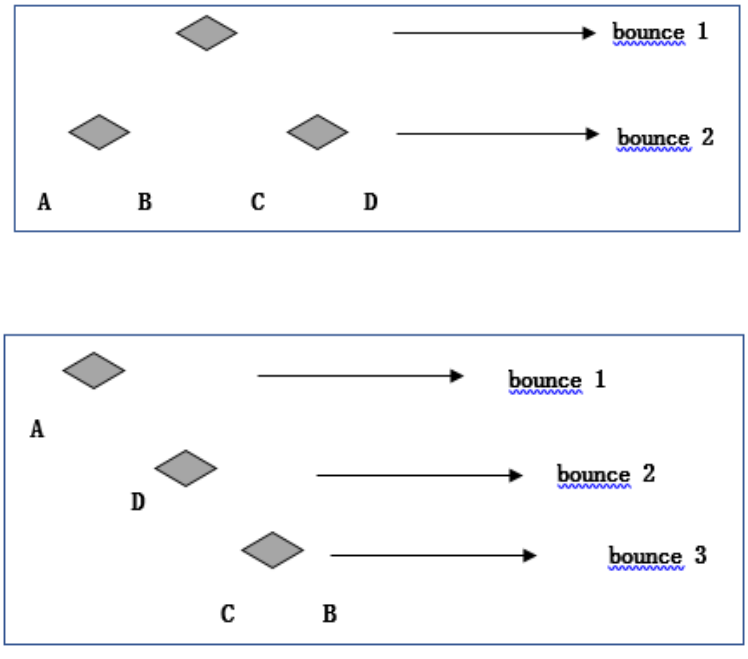


Figure29. setting bounces for each Machine

By calculating, for machine A, the bounces is 2; for machine B, the bounces is 1.75.

This means machine B is producing less information, because there is less uncertainty. And Claude Shannon called this measure of average uncertainty as Entropy(William H. Press & Brian P.Flanner,1992) and use letter H to represent it. Entropy or H is the summation of each symbol:

$$H = \sum_{i=1}^{n} Pi * \ bouncesi$$

$$bouncesi = log_2(outcomesi)$$

$$outcomesi = \frac{1}{Pi}$$

By the transformation of the above formulas, we get:

$$H = \sum_{i=1}^{n} pi * \ log_2(\frac{1}{Pi})$$

$$H = -\sum_{i=1}^{n} Pi * \ log_2(pi)$$

In actual usage, for easier calculating, we use ***In*** to replace **log*2***

Then, we get the final equation:

$$H = -\sum_{i=1} P_i * \ ln(p_i)$$

Suppose that one question, **x**, has ***I*** possible answers, labeled by ***i***, and that the other question,***y***,as ***J*** possible answers, labeled by ***j***. Then the possible outcomes of asking both questions form a contingency table whose entries ***Nij*** ,when normalized by dividing by the total number of remaining possibilities ***N***, give all the information about the ***p***:

$$p_{ij} = \frac{N_{ij}}{N}$$

$$P_{i.} = \frac{N_{i.}}{N}$$

$$p_{.j} = \frac{N_{.j}}{N}$$

The entropy of the questions ***x*** and ***y*** are, respectively:

$$H(x) = -\sum_{i} p_{i.} \ln(p_{i.}) \qquad H(y) = -\sum_{j} p_{.j} \ln(p_{.j})$$

and the entropy of the two questions together:

$$H(x,y) = -\sum_{i,j} p_{ij} \ln(p_{ij})$$

the entropy of the question y given x：

$$H(y|x) = -\sum_{i} p_{i.} \sum_{j} \frac{p_{ij}}{pi.} \ln(\frac{p_{ij}}{p_{i.}}) = -\sum_{i,j} p_{ij} \ln(\frac{P_{ij}}{p_{i.}})$$

and the entropy of x given y:

$$H(x|y) = -\sum_{j} p_{.j} \sum_{i} \frac{p_{ij}}{p_{.j}} \ln\left(\frac{p_{ij}}{p_{.j}}\right) = -\sum_{i,j} p_{ij} \ln(\frac{p_{ij}}{p_{.j}})$$

Based on above, the measure of association, here we call “dependency”:

$$U(x,y) \equiv 2\left[\frac{H(y) + H(x) - H(x,y)}{H(x) + H(y)}\right]$$

Similarly like ***Cramer's V***, the value of “dependency” also lies between zero and one, equals zero when there is no association, and equals to one when the association is perfect.

7. Results

7.1 Association plot with regions

For each region in each stale, it has the different values of “dependency” from 1980 to 2010, so we need to plot the time series of the change about theses dependencies' value. There are 216(36 regions * 6 states) different plots. Here are two example pictures that plot the “dependency” for region1 and region33 in state1.

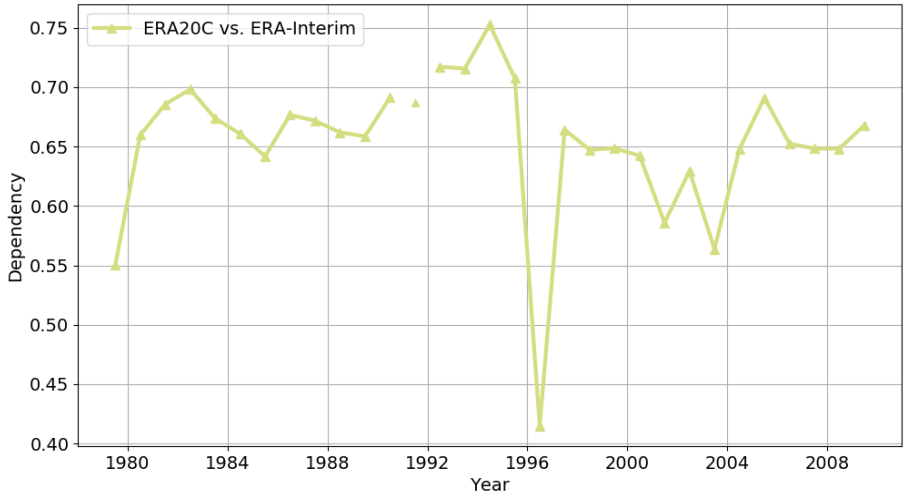


Figure30. time series of Dependency in region1,state1

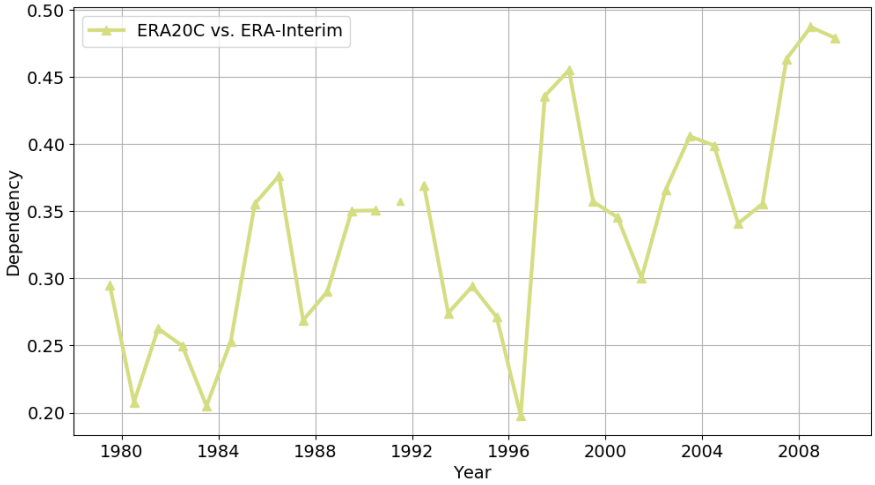


Figure31. time series of Dependency in region33,state1

For figure 1, its expectation of "dependency" value is around 0.68, and the trend is relatively stable; for figure2, its expectation of "dependency" value is not stable, the change of it relatively big, but the general trend of it is upwards. For all of the results, the minimum of value for all regions is at 1997. And because of the original data of 1992 is same as 1993, so when plotting it, the value of "dependency" in 1993 has been removed from the result.

7.2 Association plot with time

In addition, we can use another way to plot the association about ERA-Interim and ERA20C, such as global heatmap. In final, we have 180(30 years * 6 states) different heatmaps. Here are two example pictures that plot the global heatmaps for ***Cramer's V*** and Entropy's **dependency** in state1.

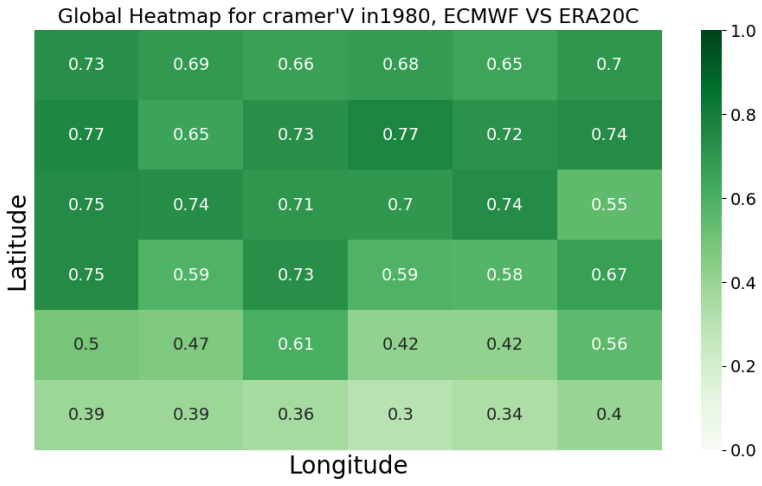


Figure32. Heatmap for cramer'v in 1980 in state1.

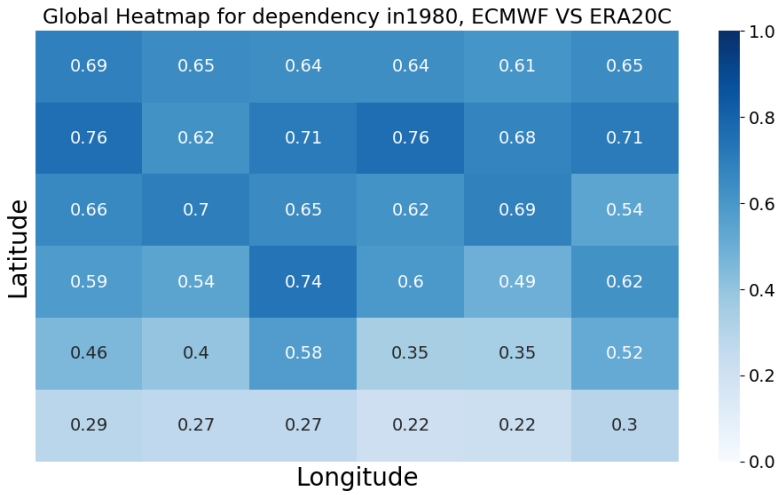


Figure33. Heatmap for dependency in 1980 in state1.

In the heatmaps, there are 36 different blocks, and each block corresponds to a region in state1.For example, in Global Heatmap for Cramer's V in 1980, the top-left value, 0.73, is the Cramer's V in region 1 in state1. And the value is represented by the shade of the color. In these two pictures, the whole trend for them is similar, it means the result of associations between ERA20C and ERA-Interim is correct by using the two different methods.

8. Conclusions

In final, we have obtained all the results that we need to help us answer the begin questions in the project:

- Which regions the quality of consistency for different atmospheric reanalyses is best、worst or others?
- Which years the quality of consistency for different atmospheric reanalyses is best、worst or others?
- How these atmospheric reanalyses characterize the atmospheric state in Antarctic regions, is it bad? And if so, how bad?

For question1,firstly we pick each state to check the results by calculating the average value of Cramer's V and dependency between 1980 and 2009 to create the final result:

For state1, the average of **Cramer's V**:

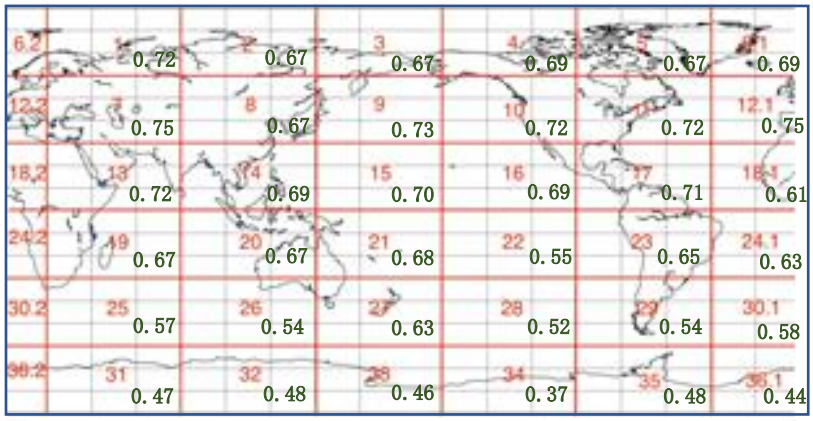


Figure34. Heatmap for expectations of **Cramer's V** between 1980 and 2009 in state1

According to the result, we can set six different levels:

Grade A (>= 0.7) : 1 7 9 10 11 12 13 15 17
Grade A- (0.65 - 0.7) : 2 3 4 5 6 8 14 16 19 20 21
Grade B+ (0.60 – 0.65) : 18 23 24 27
Grade B (0.55 - 0.60) : 25 30
Grade B- (0.50 - 0.55) : 22 26 28 29
Grade C (=< 0.50) : 31 32 33 34 35 36

For state1, the average of **Dependency**:

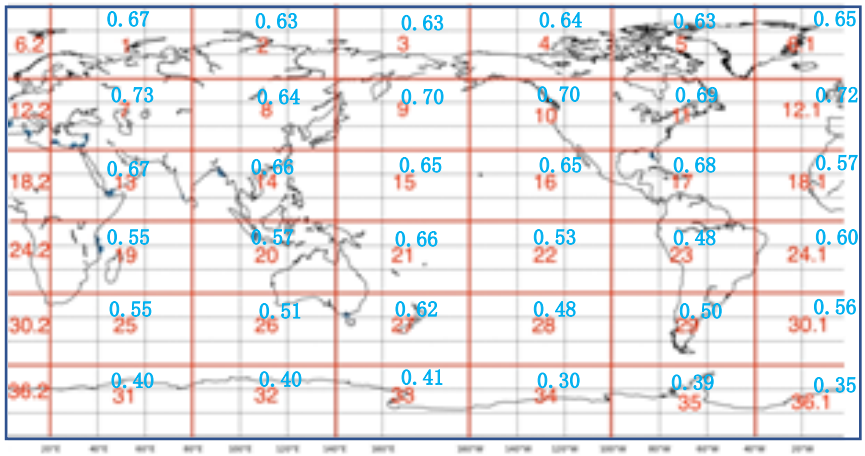


Figure35. Heatmap for expectations of dependency between 1980 and 2009 in state1

According to the result, we also can set six different levels:

Grade A (>= 0.7) : 7 9 10 12
Grade A- (0.65 - 0.7) : 1 6 11 13 14 15 16 21
Grade B+ (0.60 – 0.65) : 2 3 4 5 8 17 24 27
Grade B (0.55 - 0.60) : 18 19 20 25 30
Grade B- (0.50 - 0.55) : 22 26
Grade C (=< 0.50) : 23 28 29 31 32 33 34 35 36

If the regions get Grade A, it means that there has a very high consistency between ERA20C and ERA-Interim; if the regions get Grade A-, it means the consistency is great; if the regions get Grade B+ or B, it means the consistency is high; but if the result is Grade C, here it indicates the consistency between ERA20C and ERA-Interim is relatively poor.

From the statistical results of Cramer's V, we can find that almost all regions in the northern hemisphere have very high or high consistencies between ERA20C and ERA-Interim, and in southern hemisphere, the results about consistency gradually decrease according to the latitudes, almost regions between 0 and 30°S have a Grade B+ and B; almost regions between 30°S and 60°S have a grade B- ; all the regions below 60°S have Grade C, and from the map, we can know there regions actually are the parts of Antarctic.

From the statistical results of dependency, we can know that the some regions get the different Grades when comparing them with Cramer's V. The main difference is in Arctic, according to Cramer's V, the regions which in Arctic get the Grade A-; but according to Dependency, these regions just get B+.

Even there have some different Grades according to the two different measures, but the whole global statement is very similar. And if we apply the same method to other states, like state2 state3 state4 state 5 or state6, we also can draw the similar conclusions.

For question2,because we have get the time series of the Dependency's value for the 36 different regions respectively. So, next , we can merge the 36 different lines into one plot to compare them with each other. And since we have make Grades for each regions, so we can group the regions before merge all of them together. In this project, Grouping them according to Cramer's V, but we also can use the Dependency's value to group them alternatively.

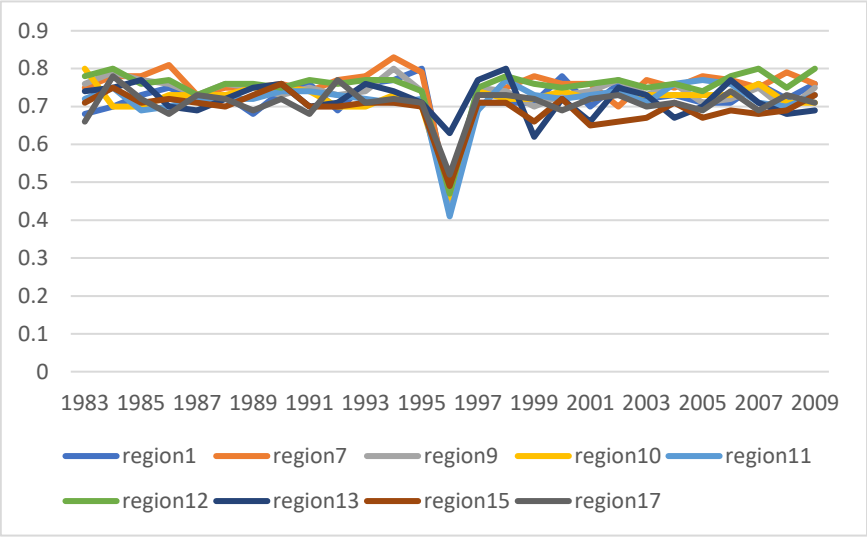


Figure36. Time Series of Cramer's V between ERA20C and ERA-Interim (Grade A+)

In Grade A group, the regions have the very similar volatility, and the values are relatively stable, which hover around 0.75. But the 1996 seems very strange, the values for all regions is the lowest ones.

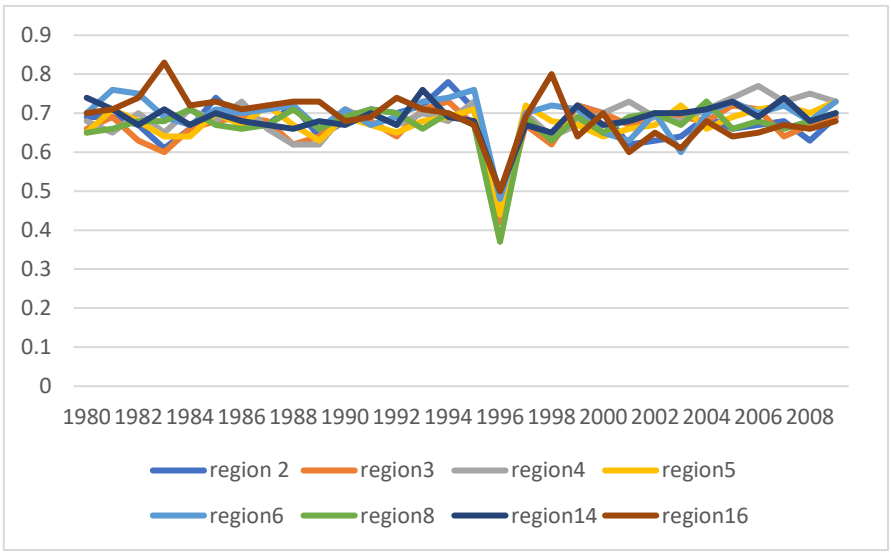


Figure37. Time Series of Cramer's V between ERA20C and ERA-Interim (Grade A)

In Grade A, the whole trend is very similar like Grade A+, just the expectations for all regions are lower than those in Grade A+. In addition, for region16, it has two peaks, which are 1983 and 1998 respectively. It means in the two year, quality of consistency between ERA20C and ERA-Interim is best. But, same as Grade A+, the lowest values also happened in 1996, which means quality of consistency is the worst one.

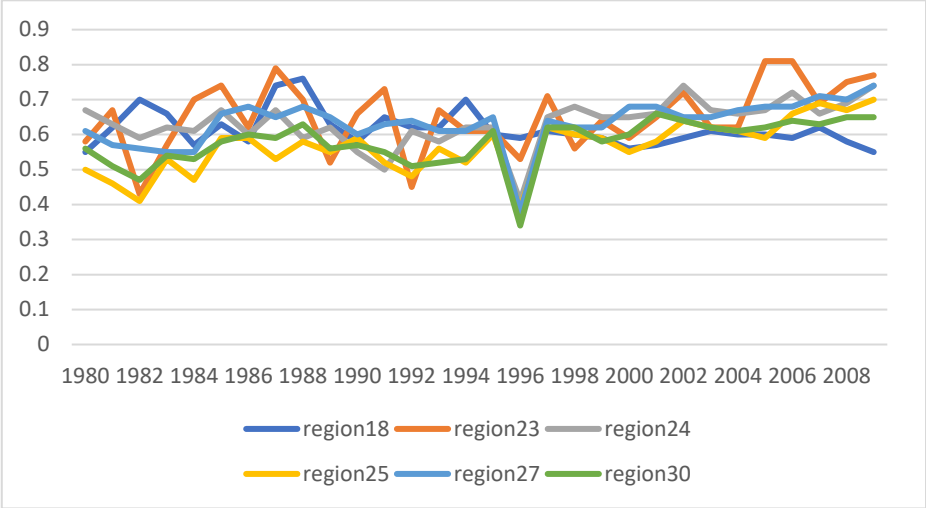


Figure38. Time Series of Cramer's V between ERA20C and ERA-Interim (Grade B+ & Grade B)

In Grade B+ and Grade B groups, all of the regions are not stable, especially for the region23, the change of Cramer's V is bigger than others. In 1982 and 1992, it had the lowest values, which are different with other regions. For region18, at early years, its Cramer's V had some change, but after 1990, the change is very small. This indicates that in region18, quality of consistency between ERA20C and ERA-Interim was stable after 1990.

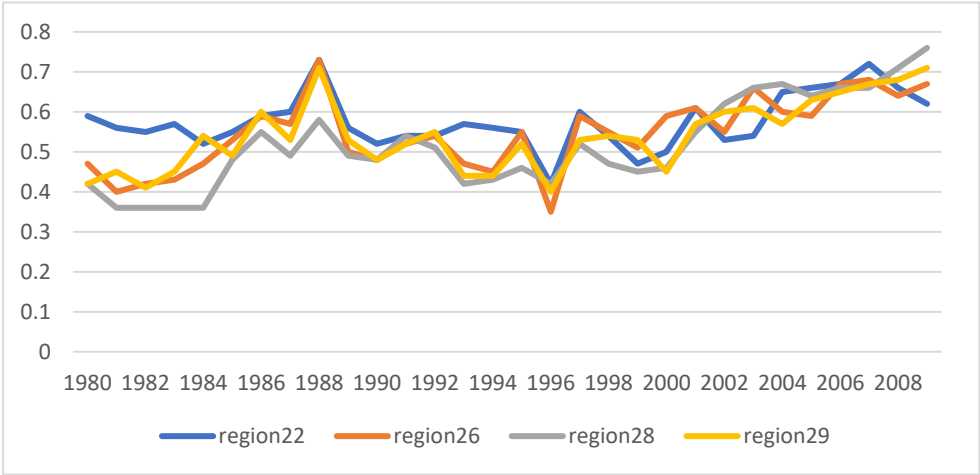


Figure39. Time Series of Cramer's V between ERA20C and ERA-Interim (Grade B-)

In Grade B- group, the regions have same trend, all of them have the unstable Cramer's V before 1988, then they reached at the peak in 1988.After, there were unstable Cramer's V

until 1996, when they got the lowest values. But after 1996, all the regions showed the same rising trends.

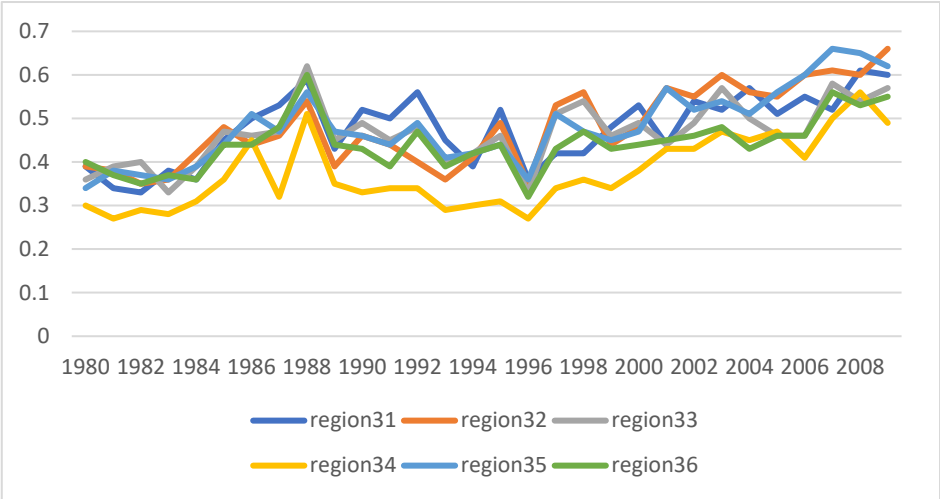


Figure40. Time Series of Cramer's V between ERA20C and ERA-Interim (Grade C)

In Grade C group, all the regions has their own trends. But by comparing all of them, the region34 has the worst values. In real geography, region34 is the part of West Antarctic.

For question3,based on the figures and charts in question1 and question2,we can determine the fact that the atmospheric reanalyses indeed characterize worst in Antarctic regions. Almost all the region blocks are graded with C, which is lowest ones; and the range of changes about Cramer's V in these regions also are larger than other regions.

9. Limitations and future expectations

This project using Self-Organizing Map algorithm to apply the near surface horizontal wind fields to determine the consistency between ERA20C and ERA-Interim from the direction of time and space respectively. But, in fact, there are more various atmospheric reanalysis products like ERRA2、JRA55、20th Century Reanalysis V2c and so on. Due to content-based restriction, the comparisons between these reanalysis products have not been implemented yet. But in future, we can apply the same approaches or other machine learning methods to do more research on them to get more comprehensive results.

In addition, in this project, the object that we analysis is just the wind fields, but in real life, the whole atmosphere system is a very complex process, it involves with many factors,

and it also strongly interacts with the underlying ocean, sea ice, and land surface processes. So, next stage, we can pick more elements like precipitation、illumination, not just wind to give more useful information.

This project focus on the consistency of wind fields globally, and the size of each region blocks that we separated is a bit large, we can make it smaller. We also can analysis it according to each countries, but it may consume much more time.

10. Summary

Because this project is from Gateway Antarctica, our main research should focus on this area. And, actually since this project is too big and we have divided it into two major parts. One part is focusing on Ross Sea/Ross Ice Shelf region, which is a very important region in the whole Antarctica； Another part is the Comparisons from the global perspective, which is what I have done in the project.

References

Andrew J. Monaghan, David H. Bromwich & Julien P. Nicolas. (2011). An Assessment of Precipitation Changes over Antarctica and the Southern Ocean since 1989 in Contemporary Global Reanalyses. *Journal of Climate, 24*(16):4189-4209. doi: 10.1175/2011JCLI4074.1

A. Hannachi. (2004). A Primer for EOF Analysis of Climate Data. UK: Department of Meteorology, University of Reading Press.

Bruce Hewitson. (2008). Climate Analysis, Modelling, and Regional Downscaling Using Self-Organizing Maps. Self-Organising Maps: Applications in Geographic Information Science, 137 – 153. doi: 10.1002/9780470021699.ch8

Compo, G. P., Whitaker, J. S., Sardeshmukh, P. D., Matsui, N., Allan, R. J., Yin,X., . . . Worley, S. J. (2011). The twentieth century reanalysis project. Quarterly Journal of the Royal Meteorological Society, 137 (654), 1-28.

David C. Howell. (2014). Chi-Square Test: Analysis of Contingency Tables. *International Encyclopedia of Statistical Science*. doi: 10.1007/978-3-642-04898-2_174

Ellen-Wien Augustijn & Raul Zurita-Milla. (2013). Self-organizing maps as an approach to exploring spatiotemporal diffusion patterns. *International Journal of Health Geographics .vol.12,*1-14. doi: 10.1186/1476-072X-12-60

Edmond Marks. (1975). Methods for analyzing multidimensional contingency tables. Resear-*ch in Higher Education*,vol3.217 – 231. doi: 10.1007/BF00991211

Fisher, Ronald A. (1922). On the Interpretation of chi-squared from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*,85, 87–94. doi:10.2307/2340 521. JSTOR 2340521

Gennady Andrienko , Natalia Andrienko , Peter Bak , Sebastian Bremm , Daniel Keim , Tatiana von Landesberger, … , Tobias Schreck. (2010). A framework for using self-organising maps to analyse spatio-temporal patterns, exemplified by analysis of mobile phone usage. *Journal of Location Based Services,vol.4*,200-221 doi: 10.1080/17489725.2010.532816.

G. Andrienko, N. Andrienko, S. Bremm, T. Schreck, T. von Landesberger, P. Bak, D. Keim. (2010). Space-in-time and time-in-space self-organizing maps for exploring spatiotemporal patterns. *IEEE - VGTC conference on Visualization*, 913-922

J. Vesanto & E. Alhoniemi. (2000). Clustering of the self-organizing map. *IEEE Transactions on Neural Networks, vol.*11 ,86 – 600.

Juha Vesanto. (1999). SOM-Based Data Visualization Methods. *Intelligent Data Analysis,vol-.l3*,111-126

Jolly, B., McDonald, A. J., Coggins, J. H. J., Zawar-Reza, P., Cassano, J., Lazzara,M., . . . Dale, E. (2016). A validation of the antarctic mesoscale prediction system using self-organizing maps and high-density observations from snowweb. *Monthly Weather Review, 144* (9), 3181-3200. doi: 10.1175/mwr-d-15-0447.1

Kohonen, T. (1990). The self-organizing map. *Proceedings of the Ieee, 78* (9), 1464-1480. doi: 10.1109/5.58325

Kadim Tas¸demir & Erzsébet Merényi. (2012). SOM-based topology visualization for interactive analysis of high-dimensional large datasets. *Machine Learning Reports, ISSN: 1865-3960.*

Stephenson,David B,Benestad & Rasmus E. (2000). Empirical Orthogonal Function analysis. *Environmental statistics for climate researchers*. Retrieved 2013-02-28.

Seefeldt, M. W., & Cassano, J. J. (2012). A description of the ross ice shelf air stream (ras) through the use of self-organizing maps (soms). *Journal of Geo physical Research Atmospheres*, *117* . doi: 10.1029/2011jd016857

Vincent Vigneron. (2006). Entropy based principle and generalized contingency tables. *ESANN'2006 proceedings - European Symposium on Artificial Neural Networks Bruges (Belgium),* 26-28. ISBN 2-930307-06-4

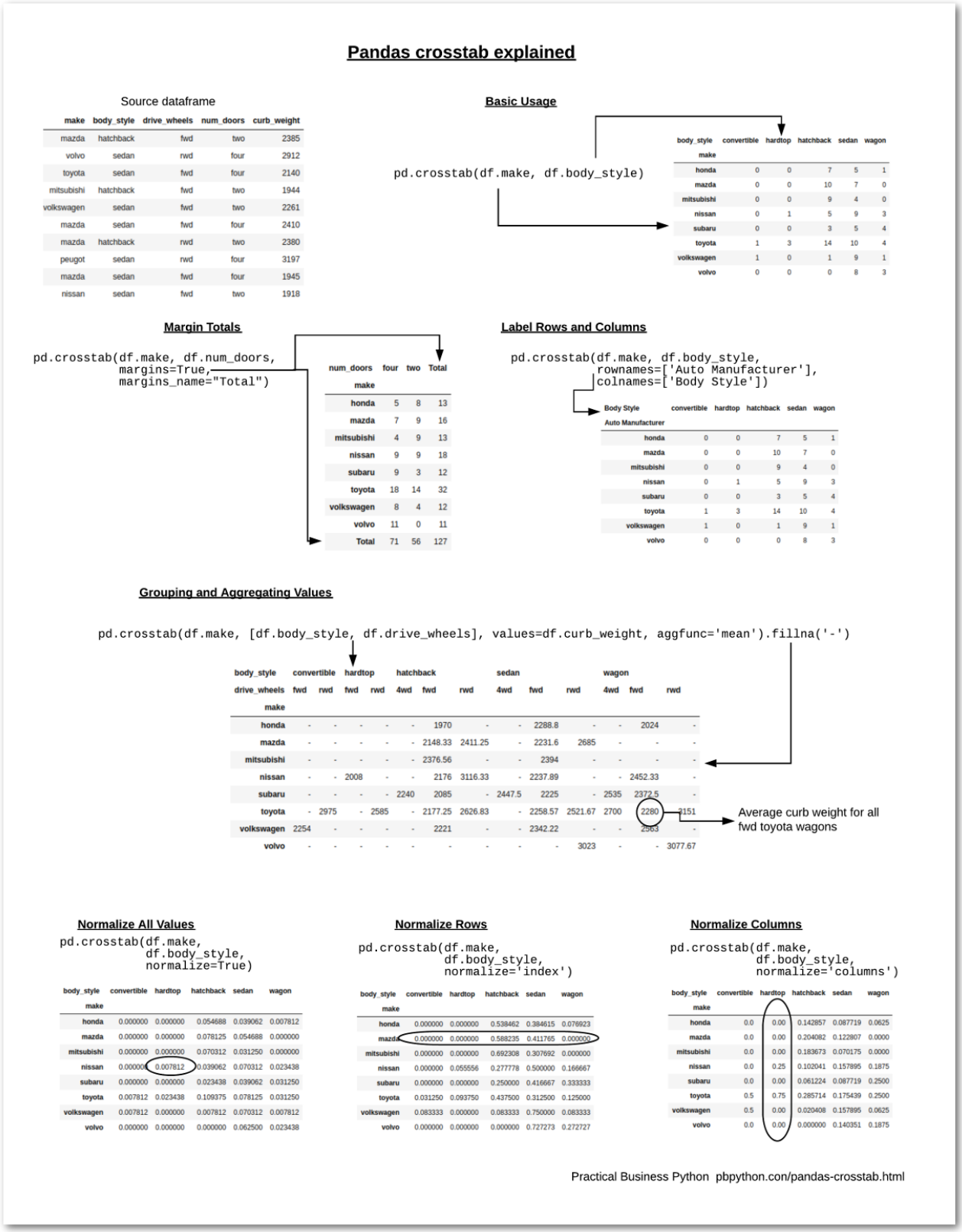
William H. Press, Saul A. Teukolsky, William T.Vetterling & Brian P.Flannery. (1992). Contingency Table Analysis of Two Discributions. *Numerical Recipes in C, The Art of Scientific Computing Second Edition,* 628—631

Xiangdong Zhang,Asgeir Sorteberg,Jing Zhang,Rudiger Gerdes & Josefino C.Comiso. (2008). Atmospheric Reanalysis Data: Its Application for Detection and Attribution of Arctic Climate Change. *Geophysical Research Letters,vol*. 35,L22701. doi:10.1029/2008GL035607,208

Xuegong Zhang & Yanda Li. (1993). Self-organizing map as a new method for clustering and data analysis. *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), vol.3,*2448 – 2451. doi:10.1109/IJCNN.1993.714219

Yates, Frank. (1934). Contingency table involving small numbers and the $\chi 2$ test. *Supplement to the Journal of the Royal Statistical Society. 1* (2): 217–235. JSTOR 2983604

Appendix


Pandas crosstab explained
Source dataframe
Basic Usage
pd.crosstab(df.make, df.body_style)
Margin Totals
pd.crosstab(df.make, df.num_doors,
margins=True,
margins_name="Total")
Label Rows and Columns
pd.crosstab(df.make, df.body_style,
rownames=['Auto Manufacturer'],
colnames=['Body Style'])
Grouping and Aggregating Values
pd.crosstab(df.make, [df.body_style, df.drive_wheels], values=df.curb_weight, aggfunc='mean').fillna('-')
Average curb weight for all fwd toyota wagons
Normalize All Values
pd.crosstab(df.make,
df.body_style,
normalize=True)
Normalize Rows
pd.crosstab(df.make,
df.body_style,
normalize='index')
Normalize Columns
pd.crosstab(df.make,
df.body_style,
normalize='columns')
Practical Business Python pbpython.con/pandas-crosstab.html