WEATHER PREDICTION DATASET

CS 579 – FINAL PROJECT – PROGRESS PRESENTATION

- Yang, Tzu – Hua
- Vignesh J Muralidharan
- Abhinav Prabhu Adarapuram
- Fadhil Salih

OVERVIEW

- INTRODUCTION
- DATA DESCRIPTION
- METADATA – ORIGINAL
- PRE PROCESSING
- METADATA – RECORDED AFTER PREPROCESSING
- DATA VISUALIZATION (TABLEAU & R)
- DATA MANAGEMENT (UPDATED)
- ANALYSIS – MACHINE LEARNING – 1 – TIME SERIES
- ANALYSIS – MACHINE LEARNING – 2 – LOGISTIC REGRESSION
- CONCLUSION
- FUTURE WORK

INTRODUCTION

- DATA COLLECTION: DATA GATHERED FROM WEATHER UNDERGROUND WEBSITE – FOR 5 MAJOR CITIES IN WORLD FOR THE YEAR 2016 – 2017
- EACH CITY WEATHER CONDITIONS ARE COLLECTED FOR ALL YEAR (365 DAYS)
- THIS DATA WAS CHOSEN IN A WAY TO UNDERSTAND THE WEATHER CONDITION IN CITIES

- AIM: DEVELOP APPROACHES FOR PREDICTING WHETHER THE EVENTS HAPPENING IS A (NORMAL SUNNY) DAY OR (RAIN/HAIL/SNOW/THUNDERSTORM) AND PREDICTING FOR YEAR 2018 ON THE BASIS OF CONSIDERING AVAILABLE TRAINING DATA.

DATA DESCRIPTION

DIMENTION OF THE DATA ⇔ 3655 ® & 25 ©

- CITY NAME
- DATE
- YEAR
- MONTH
- DAY
- TEMPERATURE – HIGH, AVERAGE, LOW
- DEW POINT – HIGH, AVERAGE, LOW
- HUMIDITY – HIGH, AVERAGE, LOW
- SEA LEVEL – HIGH, AVERAGE , LOW
- VISIBILITY – HIGH, AVERAGE, LOW
- WIND – HIGH, AVERAGE, LOW

Dataset includes,

- 2217 “NA’s” in the response variable
- Several “NA’s” in the variable in low wind
- Several “0” in the precipitation variable

city	year	month	date	day	high_tem	avg_temp	low_temp	high_dew	avg_dewp	low_dewp	high_hum	avg_humi	low_humi	high_hg	avg_hg	low_hg	high_vis	avg_vis	low_vis	high_winc	avg_wind	precip	events
Auckland	2016	1	1/1/2016	1	68	65	62	64	60	55	100	82	68	30.15	30.09	30.01	6	6	4	21	15	0	Rain/Fog/Snow/Thunderstorm
Auckland	2016	1	1/2/2016	2	68	66	64	64	63	61	100	94	88	30.04	29.9	29.8	6	5	1	33	21	0	Rain/Fog/Snow/Thunderstorm
Auckland	2016	1	1/3/2016	3	77	72	66	70	67	64	100	91	74	29.8	29.73	29.68	6	6	1	18	12	0	Rain/Fog/Snow/Thunderstorm

RMS

METADATA – ORIGINAL DATASET (DUBLIN CORE)

Digital Object Identifier (DOI) Number: The data was not published
Title / DOI Related Article: Evidence of weather conditions for different cities for the year 2016 – 2017
Variable info link: https://vincentarelbundock.github.io/Rdatasets/doc/mosaicData/Weather.html
Creator: The creator was not available
Publisher: Not published yet
Date Published: not given | **Date Downloaded in computer**: 17th Mar 2019
Source: Dataset https://vincentarelbundock.github.io/Rdatasets/datasets.html?fbclid=IwAR1a_0LfN4gf5mQjCyOacu3ucqQ1jZvlu7Tz1pd5atiLtslBQhc9QSCKyFE

Type: Metadata (.doc or .pdf), Dataset (.Xlsx, .CSV, .PPT)
Language: English
Dimension of the actual dataset: 3655 (Observations) & 25 (Variables)
List of Textual Variables: City – Total 1
List of Continuous variables: Total 20
List of categorical variables: Target (Y) Events – Total 1
List of Time series variables: Year, Month, Day – Total 3
Target: Variable Name – Events => Rainy/Fog/Thunderstorms/Snow/sunny/normal – 6 levels
Relation: Graduate student coursework project work for CS 579 – UIDAHO

PREPROCESSING

- STARTING WITH "NA'S" IT HAS BEEN REVALUED TO "0" EXCEPT FOR THE TARGET VARIABLE
- RESPONSE "EVENTS" HAD 2217 NA'S
  - LOGICAL ORGANIZATION, INSTEAD OF HAVING 12 LEVELS WE ARE CONVERTING INTO 2 LEVELS
  - NOT EVERYDAY THE CITY WILL HAVE RAIN OR SNOW OR THUNDERSTORMS AND HAIL
  - CONVERTED THE AVAILABLE OBSERVATIONS TO ONLY LEVEL JUST "RAIN/SNOW/HAIL/THUNDERSTORMS"
  - CONVERTED THE "NA'S" TO "NORMAL/SUNNY" DAYS

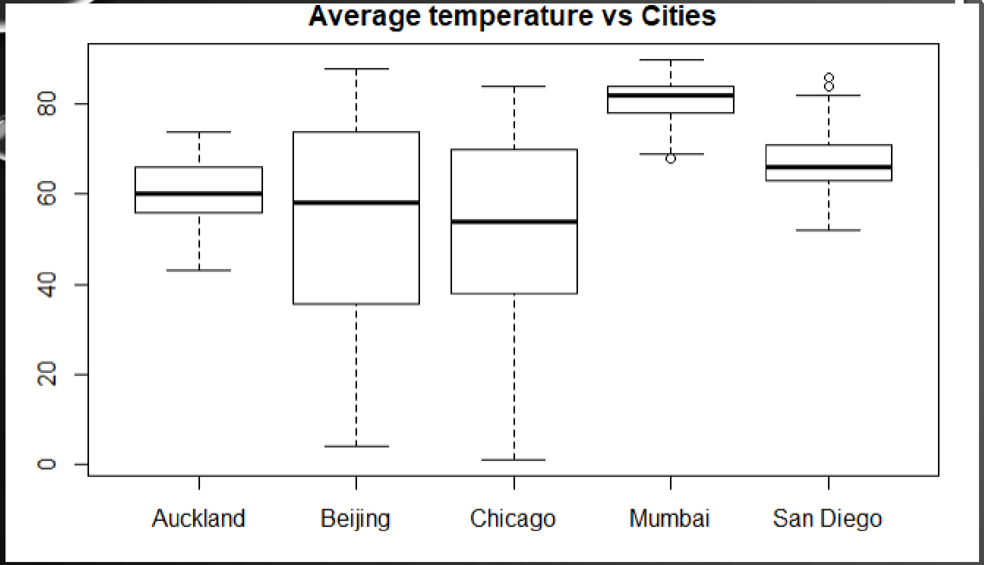
- **DIMENTION OF THE DATA AFTER PREPROCESSING ⬄ 3655 ® & 24 ©**

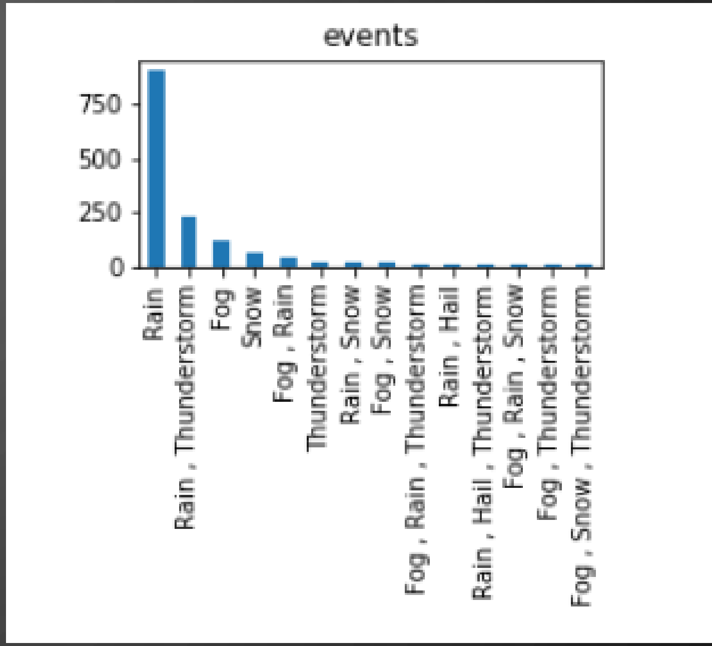
METADATA – AFTER PREPROCESSING

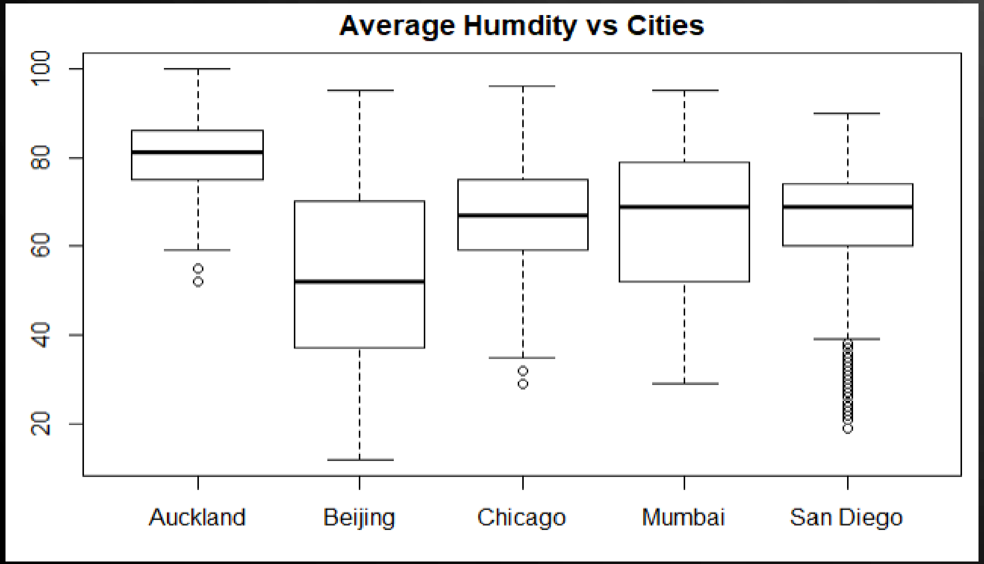
Digital Object Identifier (DOI) Number: The data was not published
Title / DOI Related Article: Evidence of weather conditions for different cities for the year 2016 – 2017
link: https://vincentarelbundock.github.io/Rdatasets/doc/mosaicData/Weather.html
Creator: The creator was not available
Publisher: Not published yet
Date Published: not given| **Date Downloaded in computer**: 28th Mar 2019
Source: Dataset <https://vincentarelbundock.github.io/Rdatasets/datasets.html?fbclid=IwAR1a_0LfN4gf5mQjCyOacu3ucqQ1jZvlu7Tz1pd5atiLtslBQhc9QSCKyFE

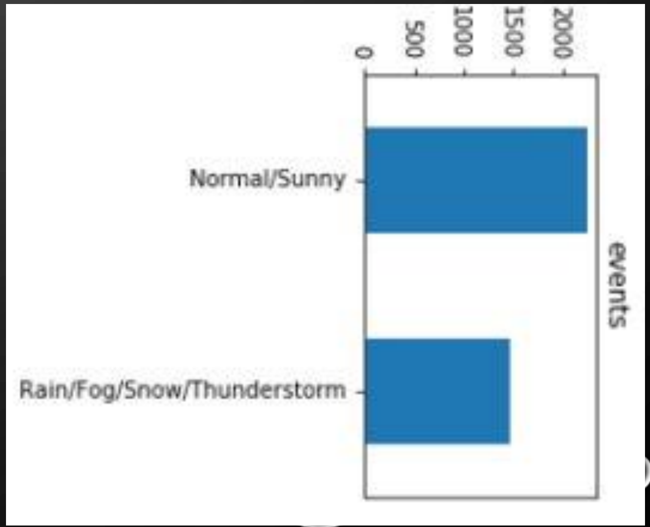
Type: Metadata (.doc or .pdf), Dataset (.Xlsx, .CSV, .PPT)
Language: English
Dimension of the actual dataset: 3655 (Observations) & 24 (Variables)
List of removed variables: SNO, date , low wind (because of more NA's)
List of Textual Variables: City – Total 1
List of Continuous variables: Total 18
List of categorical variables: Target (Y) Events – Total 1
List of Time series variables: Year, Month, Day – Total 3
Target: Variable Name – Events => Binary (0 – Rainy/Fog/Thunderstorms/Snow, 1 – Normal/Sunny days)

DATA VISUALIZATION – WITH PREPROCESSED RESPONSE

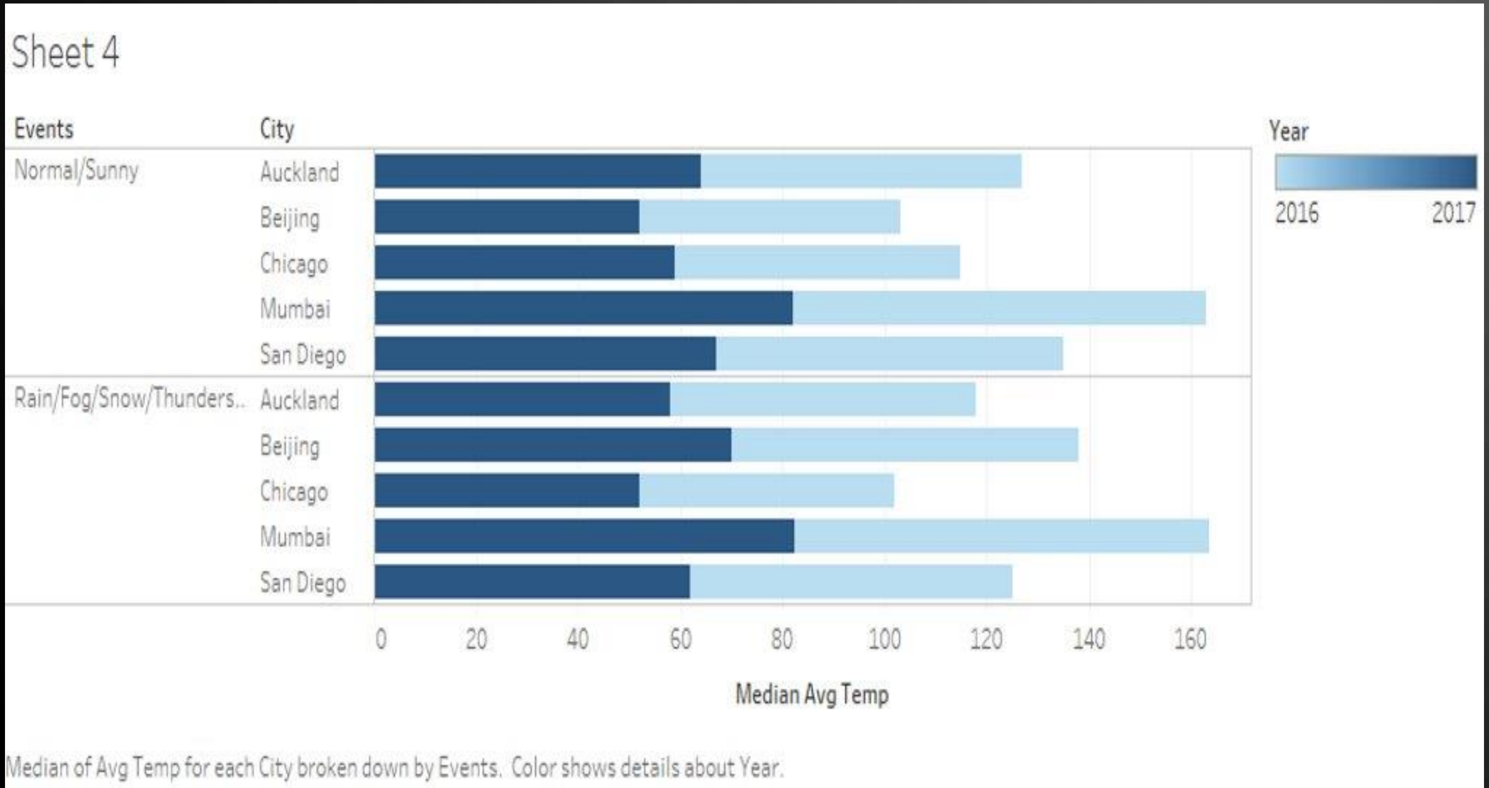


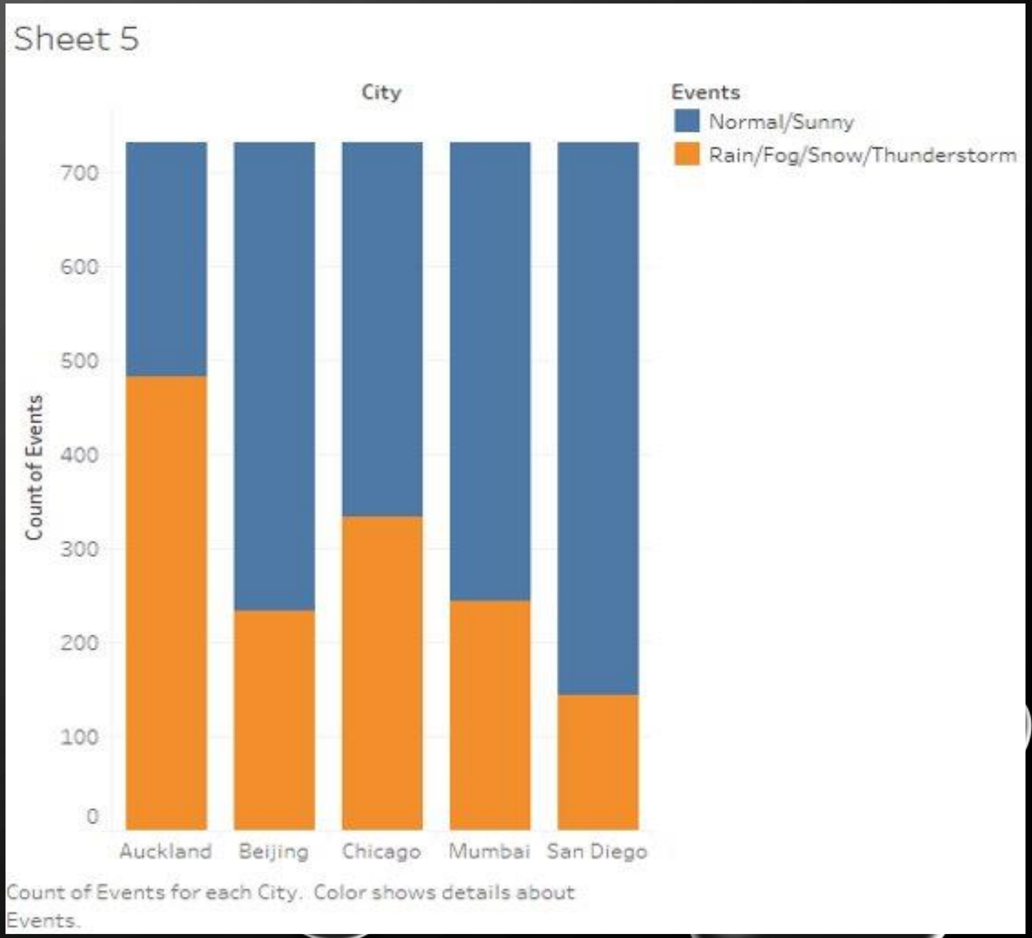




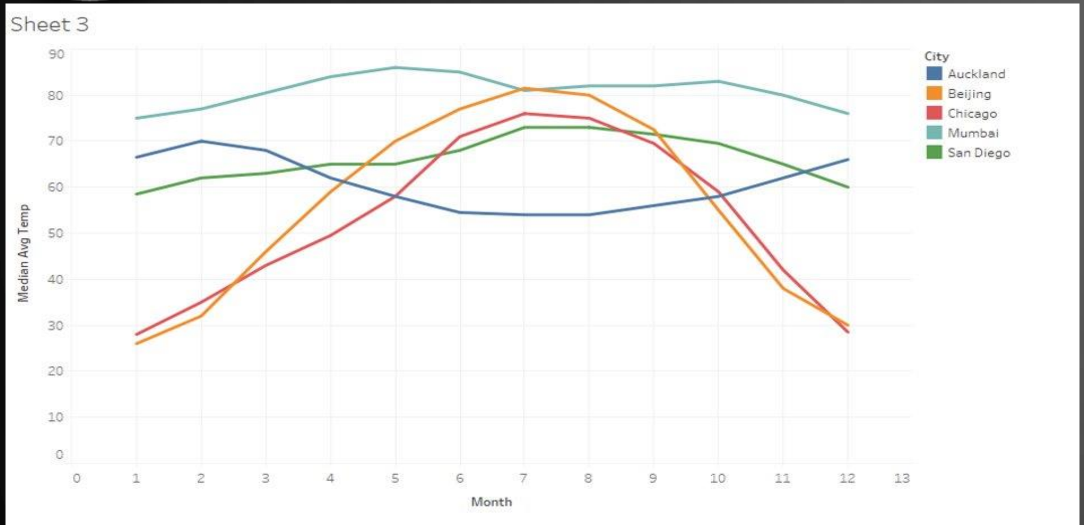


DATA VISUALIZATION “TABLEAU”

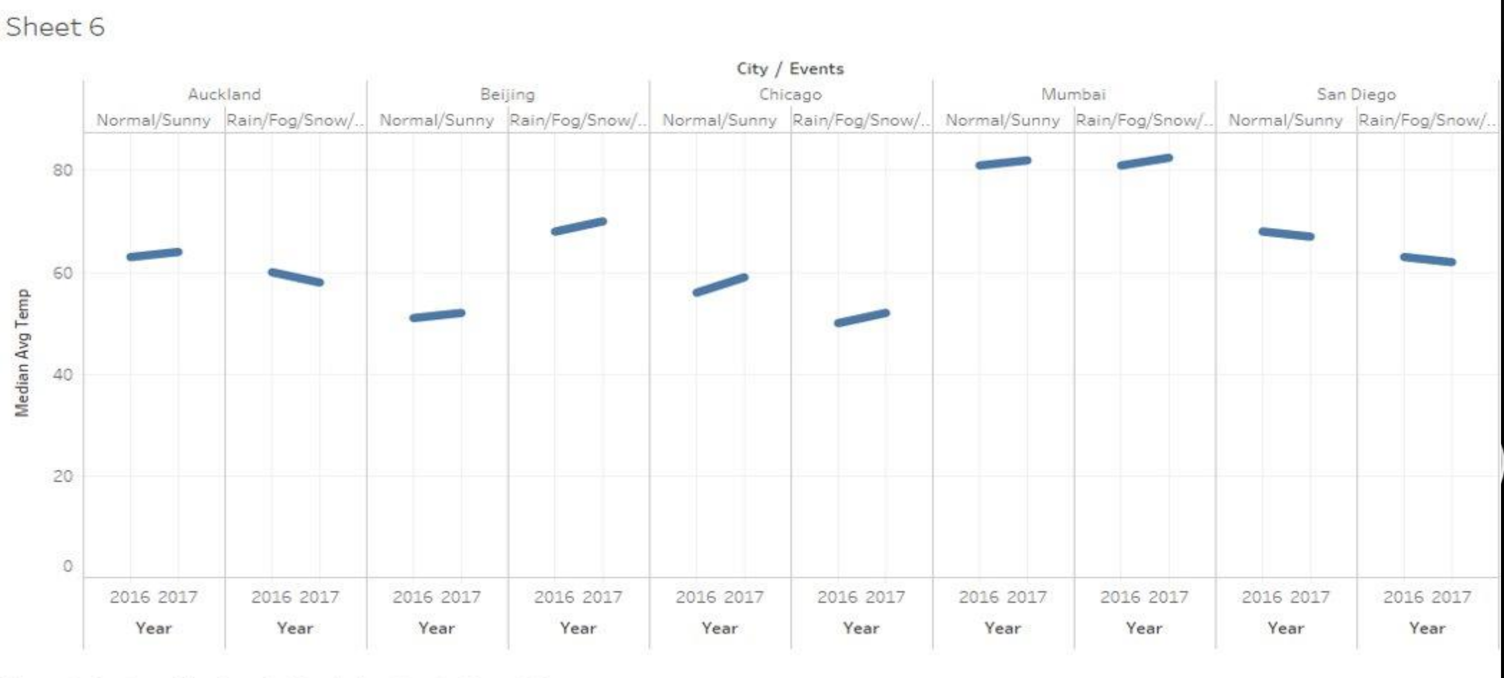




DATA VISUALIZATION “TABLEAU” (CONT…)

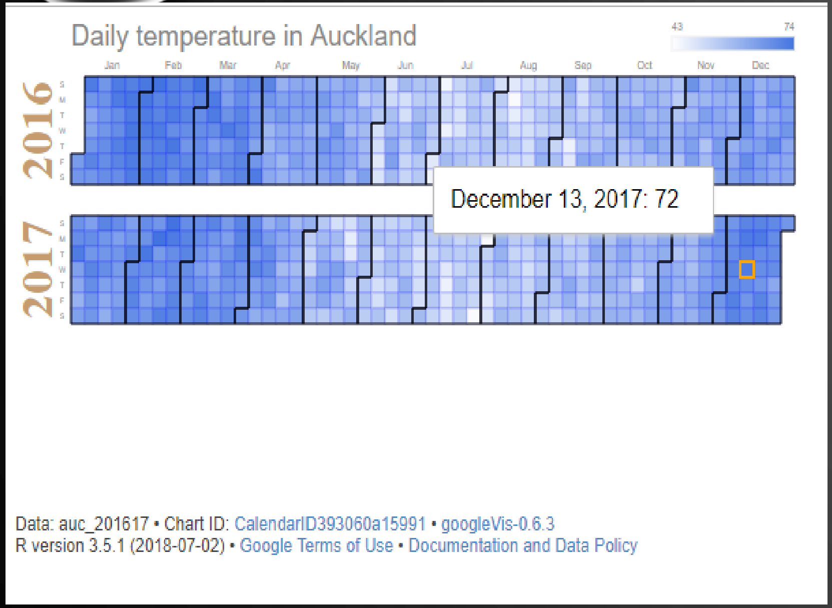


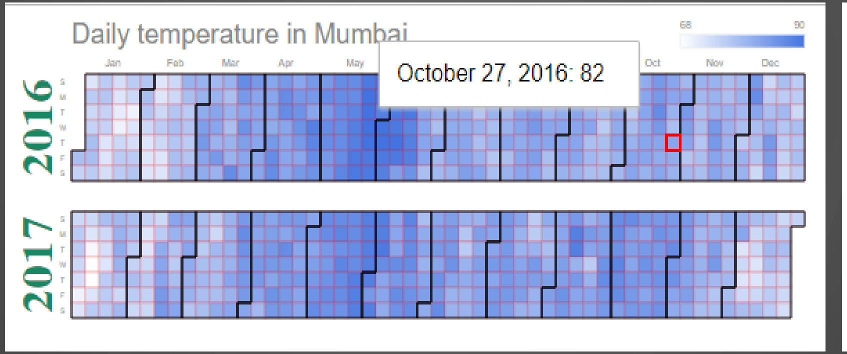
The trend of median of Avg Temp for Month. Color shows details about City.

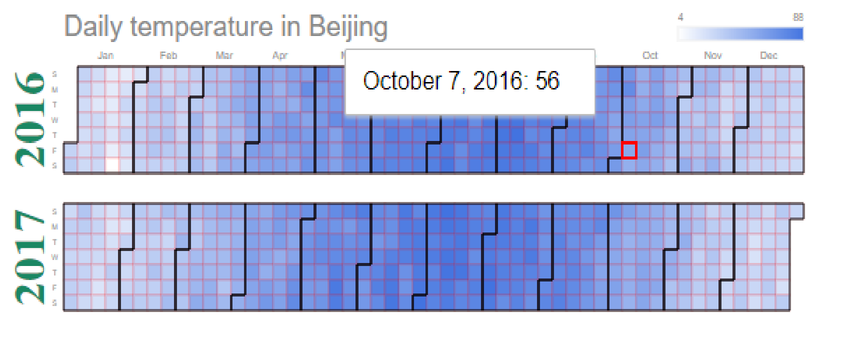


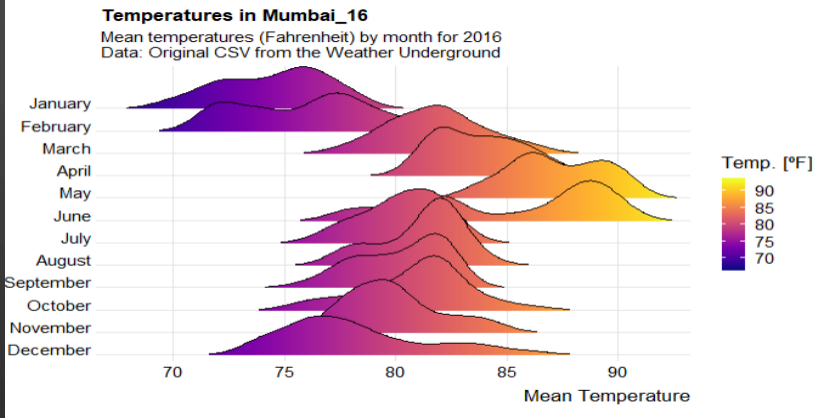
The trend of median of Avg Temp for Year broken down by City and Events.

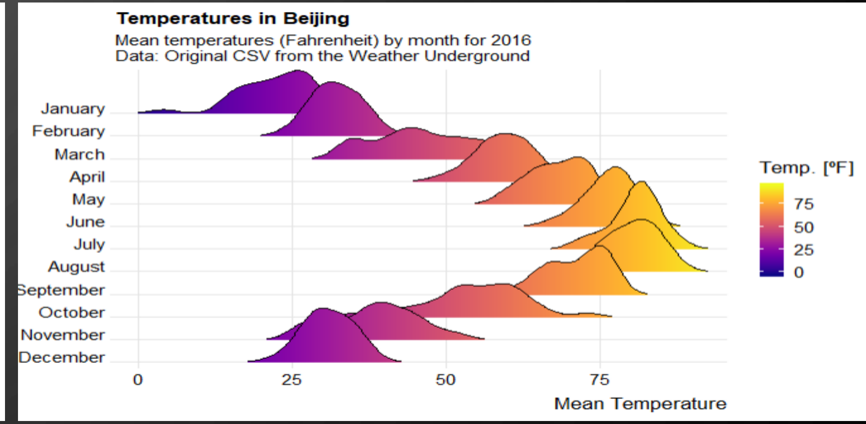
DATA VISUALIZATION IN "R" (CONT…)

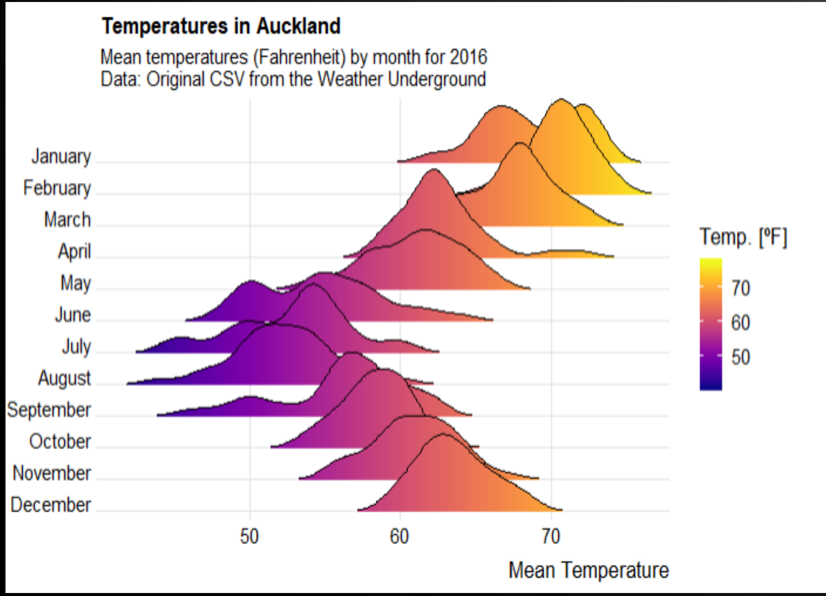


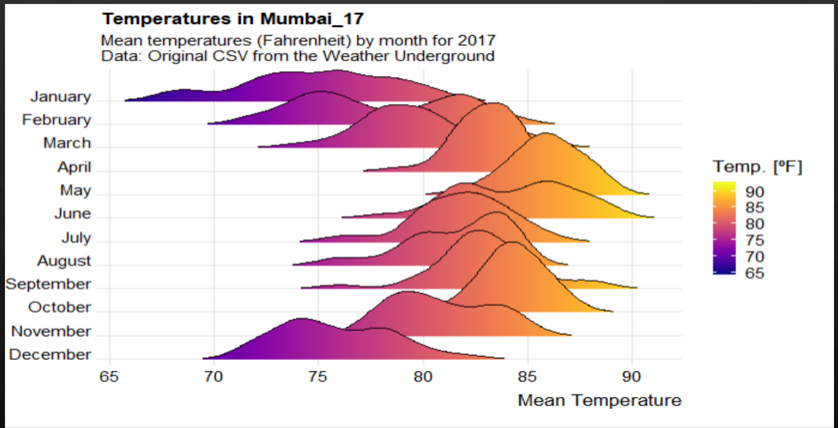


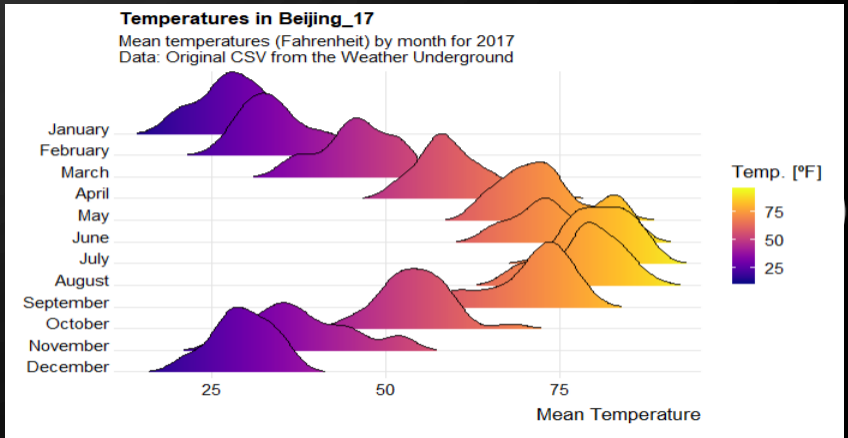




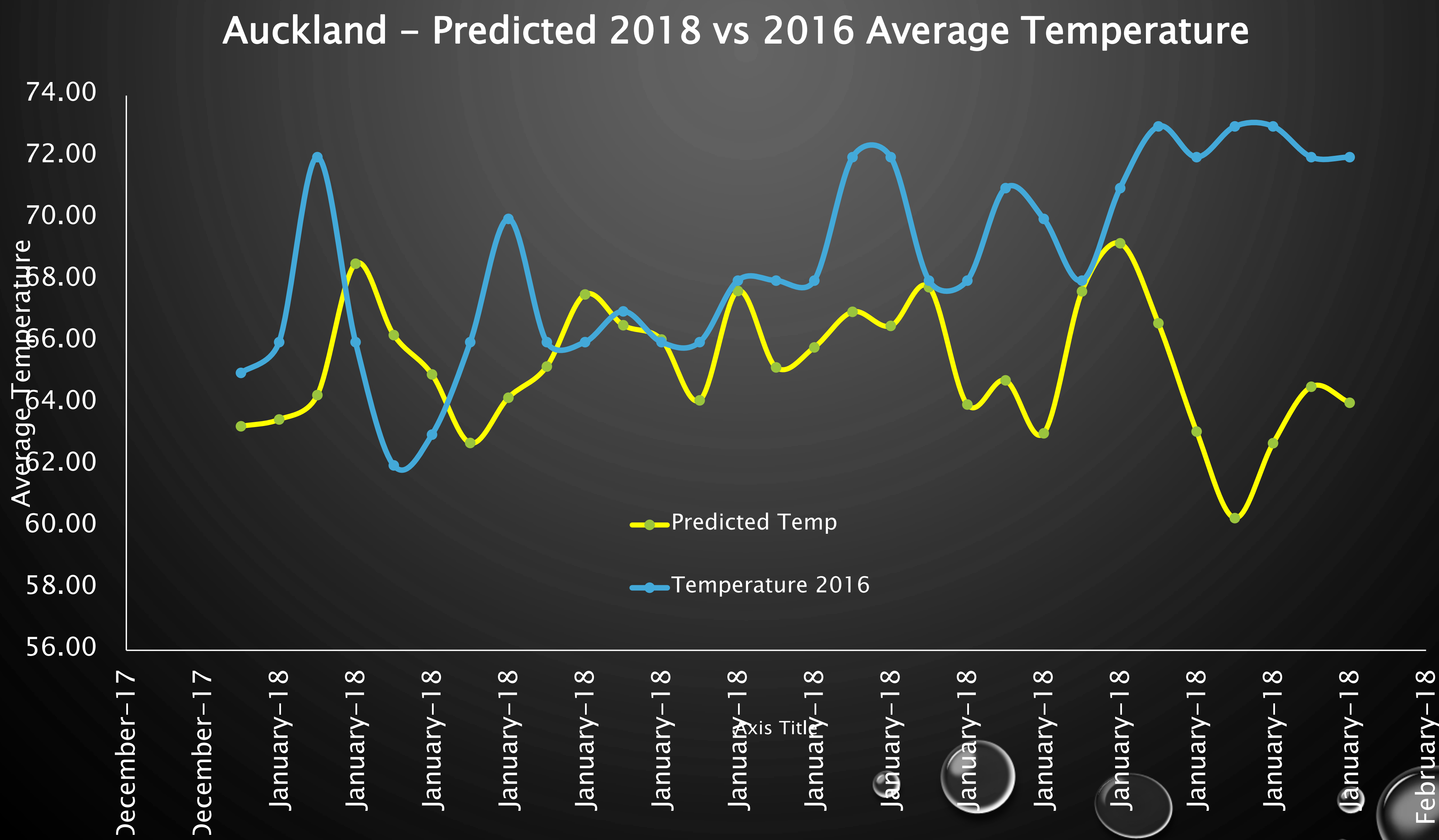




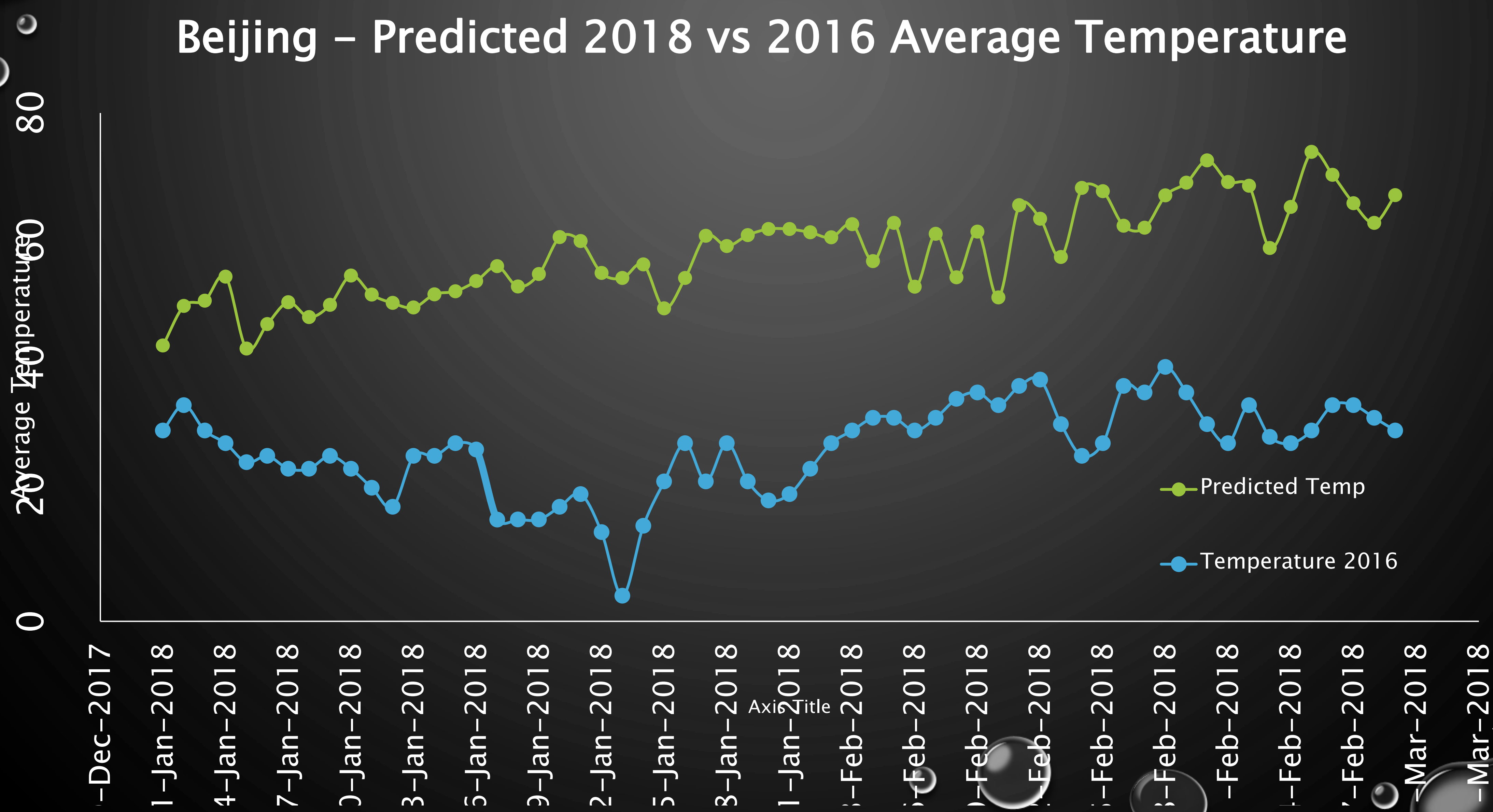




PREDICTED RESULTS



PREDICTED RESULTS



DATA MANAGEMENT (UPDATED)

1. **Creation of logical collections**: We analyzed required logical understandings and transformed using R and Excel by visualizing different attributes
2. **Physical Data Handling**: We have backed up data each time when we updated, imputed or transformed the data into one drive and Github as .CSV document.
3. **Interoperability support**: We made sure data could be downloaded as .CSV format and we all have access through cloud which will help and use in different platforms like R, Python, JAVA, Tableau.
4. **Data ownership**: This data is not published by the data creator. So, all members of the team will be reviewing any edited aspects to ensure the actual needs. The ownership is not planned to share as of now.
5. **Security Support**: Metadata is stored in Main project directory.
    1. All team members + Project watching member have access to one drive
    2. All edits made by members can be viewed by all 5 people

DATA MANAGEMENT (CONT..)

6. **Management & Access**: Metadata is included (Dublin Core standard) as .DOC file in main directory of one drive project

7. **Persistence**: Data updated as per March 28th2019, Old and future records were updated as and when needed in one drive and persistence has been maintained in all circumstances.

8. **Discovery**: The link of the data before preprocessing is provided in Metadata. Once published the link will be provided for public

9. **Data Dissemination and publications**: Edited versions of data has been stored to compare the original dataset. As of now we are using
    1. Data Visualization (R, Python, Tableau)
    2. Imputation & data frame editing (R & Python)
    3. Analysis (R, Python, JAVA)

    Changes will be recorded in metadata and shown if class students or professor is interested

ANALYSIS 1– TIME SERIES ALGORITHM

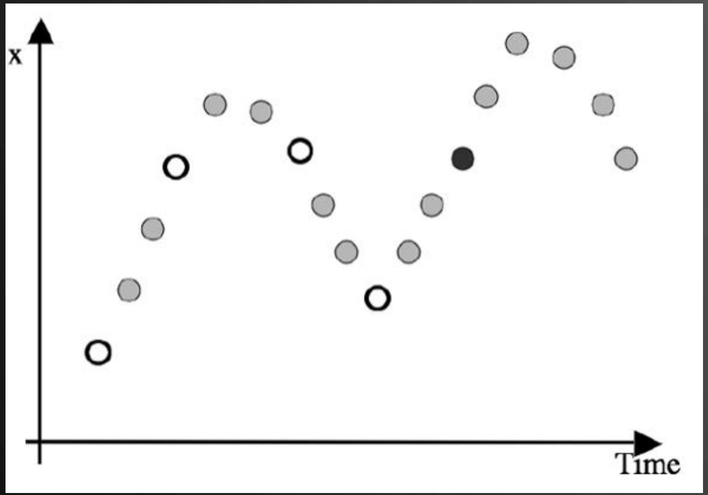
- **Goal**: To predict the future temperature.

- **Programming Language & Library**:
  - Java
  - Jblas: Linear Algebra for Java

- **Algorithm**:
  - Time series algorithm
  - Linear Regression (Transfer Function: Sigmoid Function)

- **Expected results**:
  - Comparison of new temperature and old temperature
    - Chart.js – Line Chart

TIME SERIES ALGORITHM

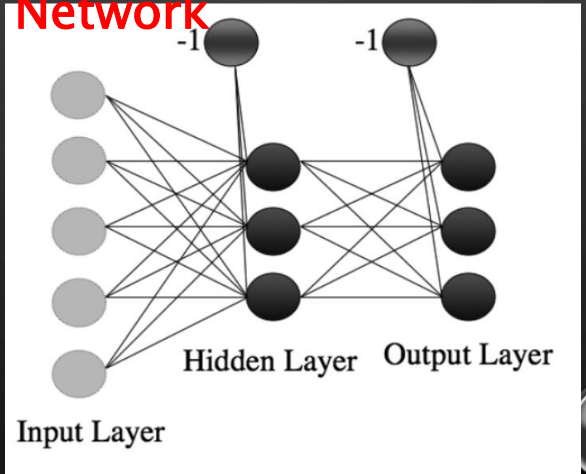


White Point	Steps
Black Point	Target
WhitePoint_1 – WhitePoint_2	Strides

Training Data

[Step, Step, …, Step]
[Step, Step, …, Step]
[Step, Step, …, Step]
[Step, Step, …, Step]
[Step, Step, …,

Multi-Layer Perceptron Network



Weight of Input Layer

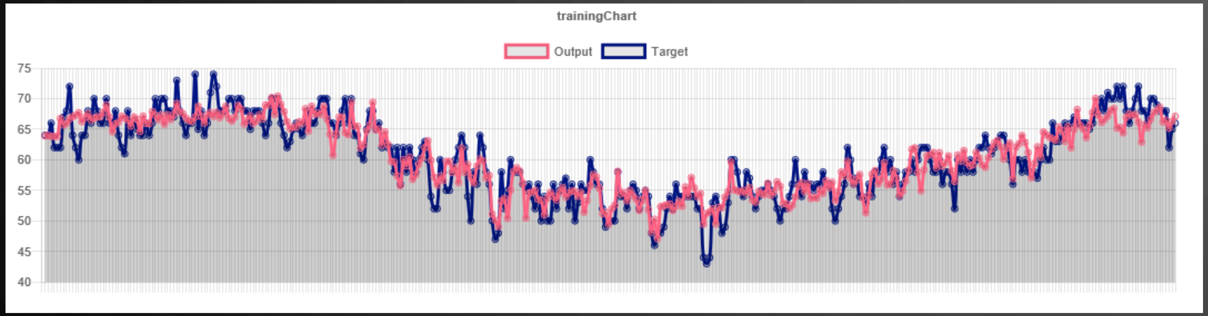
Weight of Hidden Layer

TIME SERIES ALGORITHM – RESULT

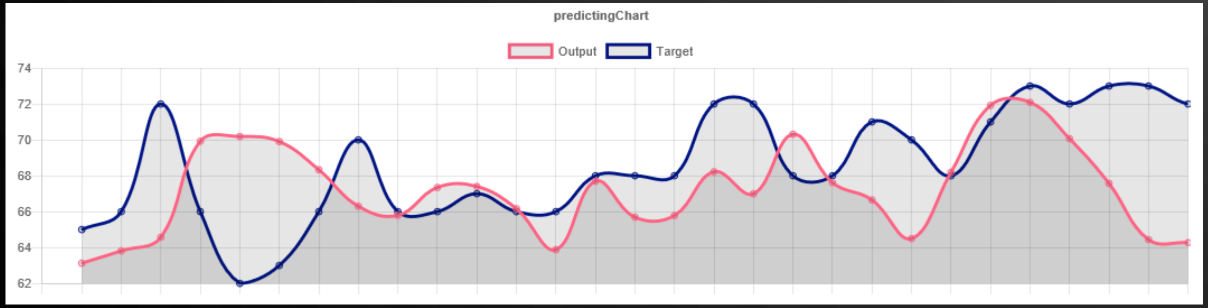
City– Auckland

JavaScript Library – Chart.js

Training Result



Prediction (1 Month)



Code

```
new Chart(ctx, {
    // The type of chart we want to create
    type: 'line',
    // The data for our dataset
    data: {
        labels: labels,
        datasets: [{
            label: 'Output',
            // backgroundColor: 'rgb(255, 99, 132)',
            borderColor: 'rgb(255, 99, 132)',
            data: output
        }, {
            label: 'Target',
            // backgroundColor: 'rgb(255, 99, 132)',
            borderColor: 'rgb(0, 20, 132)',
            data: target
        }]
    },
    // Configuration options go here
    options: {
        title: {
            display: true,
            text: id
        }
    }
});
```

TIME SERIES ALGORITHM – RESULT

- Accuracy of Prediction is not high
- Lack of expert knowledge
- Lack of Sample
- It might could be combined with other algorithms, such as
  - Using PCA
  - Using Validation Data for training
- Better arrangement of step and stride

ANALYSIS 2 : LOGISTIC REGRESSION

$$\log \left[\frac{Y}{1-Y} \right]$$

- Uses a logistic function to model a binary dependent variable from one or more independent variables
- **Target Variable (Binary):** Event => (Rain/No Rain) / (0/1)
- **Expected results**:
  - Training and Testing accuracy
  - ROC Curve

LOGISTIC REGRESSION

'precip' column was removed due to 'T' values in the given dataset

```
In [9]:  x = pd.DataFrame(wthr)
         x = wthr.drop(['events', 'precip'], axis = 1)
         y = pd.DataFrame([wthr.events]).T # Separating the response variable
         #y= y.astype('int')
         print (x.shape)
         print (y.shape)

(3655, 21)
(3655, 1)
```

The dataset was split into training and testing datasets

```
In [10]: xtrain, xtest, ytrain, ytest = train_test_split(x, y, random_state = 7, test_size = 0.25)
         print (xtrain.shape)
         print (xtest.shape)
         print (ytrain.shape)
         print (ytest.shape)

(2741, 21)
(914, 21)
(2741, 1)
(914, 1)
```

LOGISTIC REGRESSION: RESULTS

Training and testing accuracies

Code:

```
In [12]:  Log_Reg = LogisticRegression()
          Log_Reg.fit(xtrain, ytrain)
          y_mod = Log_Reg.predict(xtest)
          accuracy_Log1 = round(Log_Reg.score(xtrain, ytrain) * 100, 2)
          print ('Training accuracy = {}'.format(accuracy_Log1))
          accuracy_Log2 = round(accuracy_score(ytest, y_mod) * 100, 2)
          print ('Testing accuracy = {}'.format(accuracy_Log2))

          Training accuracy = 84.68
          Testing accuracy = 84.79
```

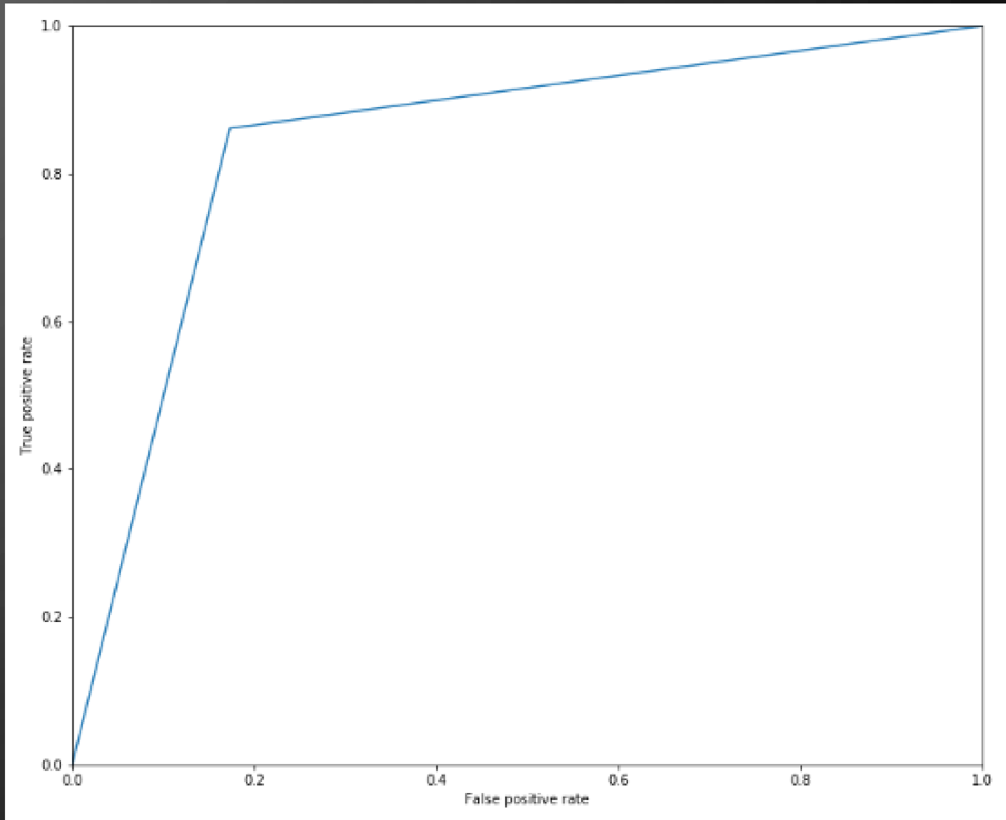
- We can see that the logistic regression model is 84% accurate in predicting the event

LOGISTIC REGRESSION: RESULTS

Receiver Operation Characteristics (ROC) Curve

Code:

```
In [10]:  from sklearn import preprocessing
          Label_encoder = preprocessing.LabelEncoder()
          y1 = Label_encoder.fit_transform(ytest)
          y2 = Label_encoder.fit_transform(y_mod)
          from sklearn import metrics
          plt.figure(figsize = (12, 10))
          fpr, tpr, thresholds = metrics.roc_curve(y1, y2)
          #thresholds[0] = 0.80
          plt.plot(fpr, tpr)
          plt.xlabel('False positive rate')
          plt.ylabel('True positive rate')
          plt.xlim([0.0, 1.0])
          plt.ylim([0.0, 1.0])
```



- True positive rate and false positive rates are plotted for all possible thresholds (Range: 0–1)

- We achieve true positive values (accurate predictions) 84% of the time

CONCLUSION

- Metadata has been created and updated for each change
- Data cleaning/Preprocessing has been carried out according to logical requirement
- Data management practices were successfully performed
- Useful information was observed through various data visualization methods
- With more weather data for years before 2016, time series algorithm could predict results with higher accuracy
- Logistic regression model was applied on the weather dataset and the events were predicted with significant accuracy

FUTURE WORK

By collecting more weather data for the years before 2016, better results can be produced such as:

- Predict the city based on climatic conditions during a specific time of the year
- Predict the weather data for 2018 and later
- Understand the effect of climate change over the years

THANK YOU