Note: this document is only used as backup in case our html file isn't readable or there is any problem with it or any of the interactive elements. It was generated through `output: word_document`

Report Group 4

MBD S2 Group 4

Navidades 2020

Introduction

'As part of the course **"Programming in R" at IE - HST**, we foster our newly learned skills by participating in a Kaggle competition. The problem requires us to do Exploratory Data Analysis, Data Cleaning and Manipulation and implement some sort of Machine Learning in R.'

The data can be found here

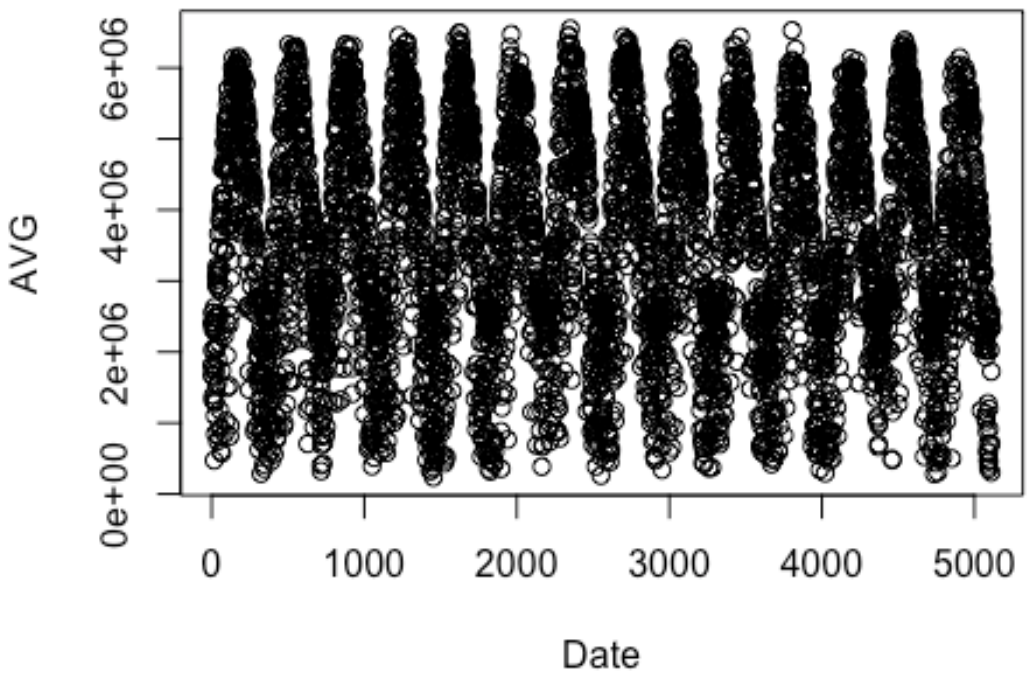
Exploratory Data Analysis Highlights

In order to carry out this task, we are first going to take a closer look at the data available, to determine which transformations should be taken place before we proceed to data processing stage and then make the predictive model.

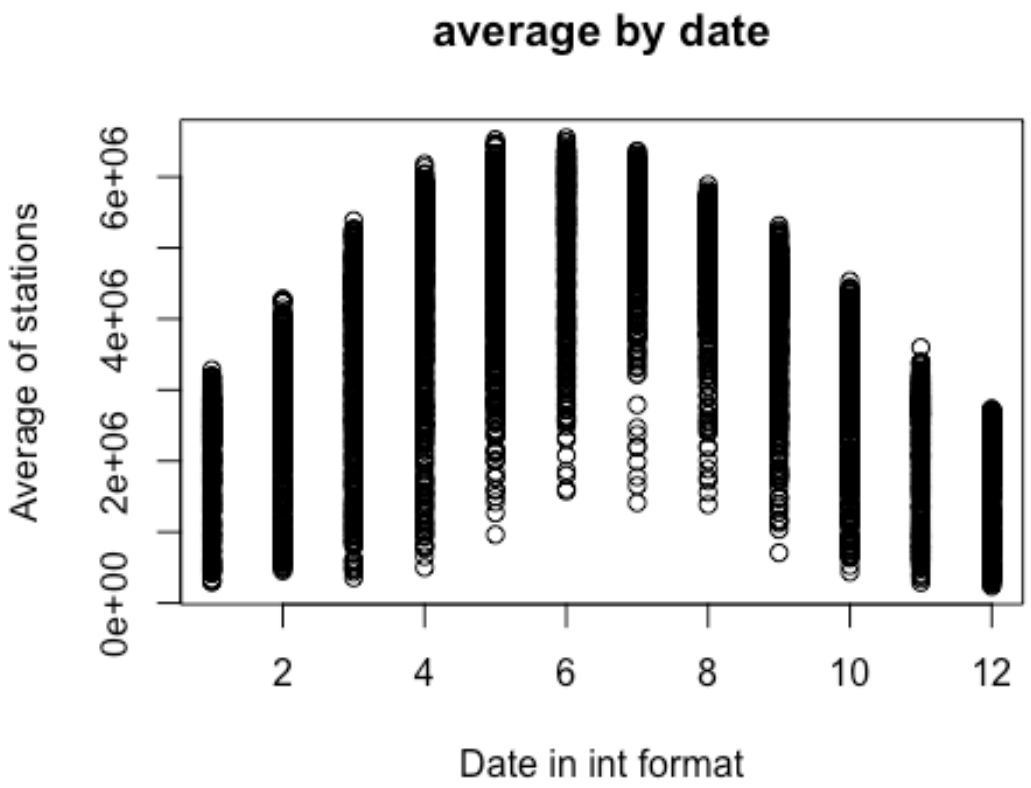
Including Plots: Dates dependent

We have designed several plots to understand and include the right variable in our prediction, first we took a closer look at the stations pattern depending on the dates:

Here an overall visualization



Here visualizing by month - averaging the stations



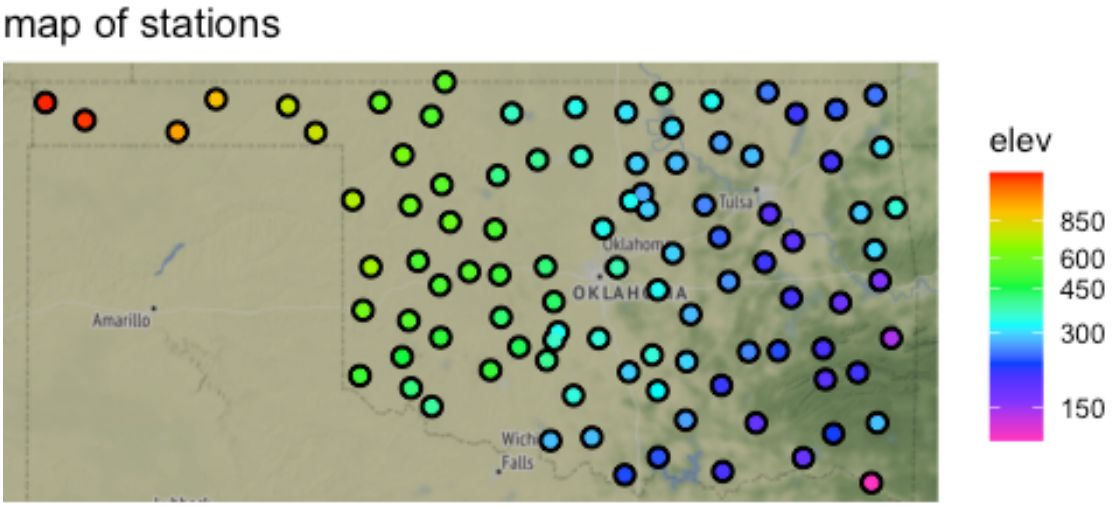
Including Plots: station coordinates dependent

We have merged the station additional information file with the summary of each stations as a data table to produce a comprehensible comparison between the average of each stations accross all date, its altitude and coordinate

Here we are plotting each station to its location on a map with a color code corresponding to it's altitude

```
## Loading required package: ggplot2

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.

## Using zoom = 7...

## Source : http://tile.stamen.com/terrain/7/27/49.png

## Source : http://tile.stamen.com/terrain/7/28/49.png

## Source : http://tile.stamen.com/terrain/7/29/49.png
```

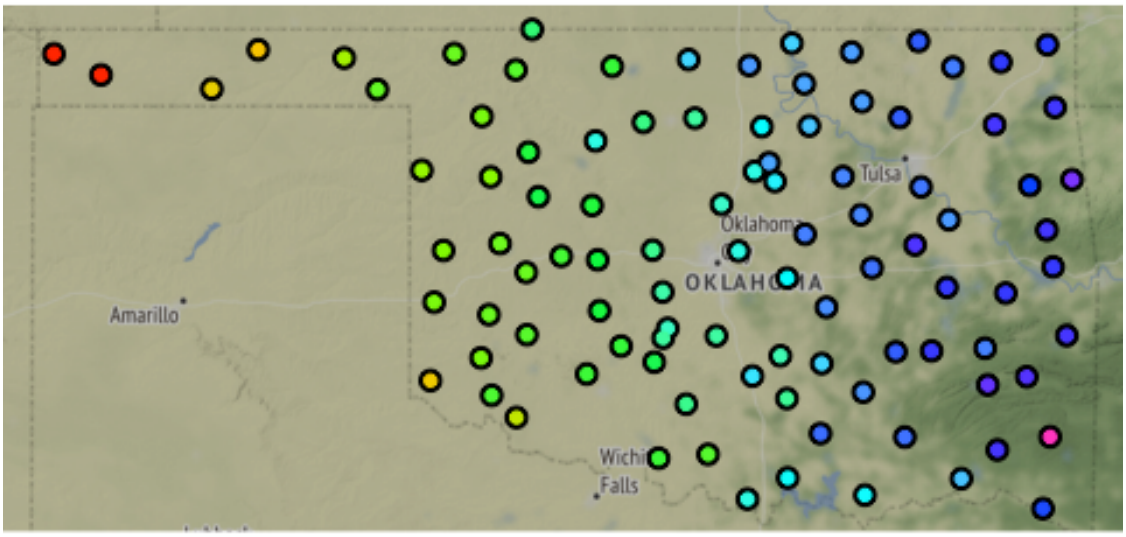
```
## Source : http://tile.stamen.com/terrain/7/30/49.png
## Source : http://tile.stamen.com/terrain/7/27/50.png
## Source : http://tile.stamen.com/terrain/7/28/50.png
## Source : http://tile.stamen.com/terrain/7/29/50.png
## Source : http://tile.stamen.com/terrain/7/30/50.png
## Source : http://tile.stamen.com/terrain/7/27/51.png
## Source : http://tile.stamen.com/terrain/7/28/51.png
## Source : http://tile.stamen.com/terrain/7/29/51.png
## Source : http://tile.stamen.com/terrain/7/30/51.png
```



And to compliment that, here we are plotting each station to its location on a map with a color code corresponding to it's average over time

```
## Using zoom = 7...
```

map of observations



Plot conclusion

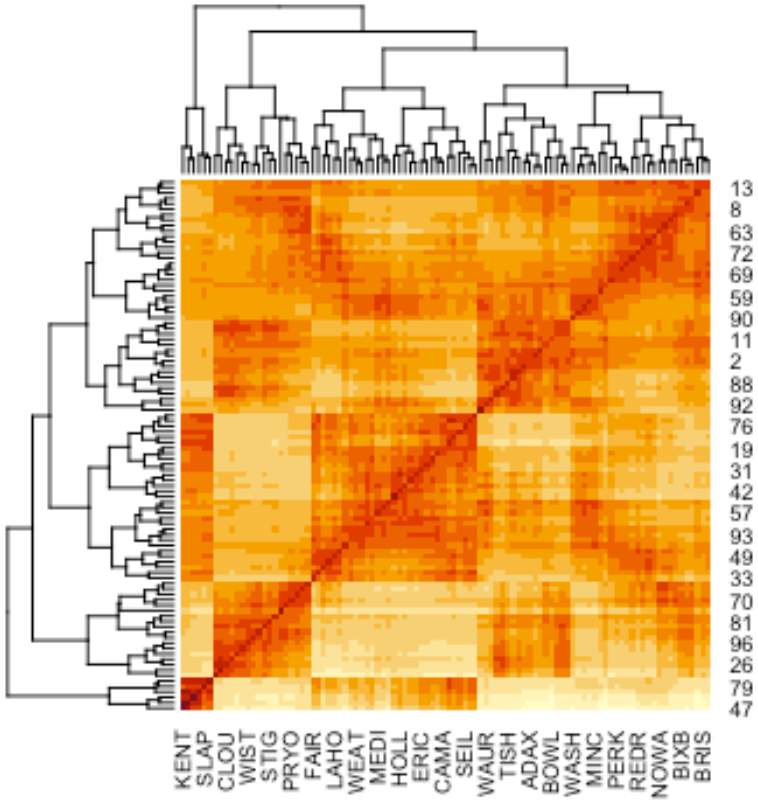
Using GGMAP function, we were able to find a strong correlation between altitude and average observation

Data is time variant with seasonnality pattern; higher osbervation are noticed in the summer month

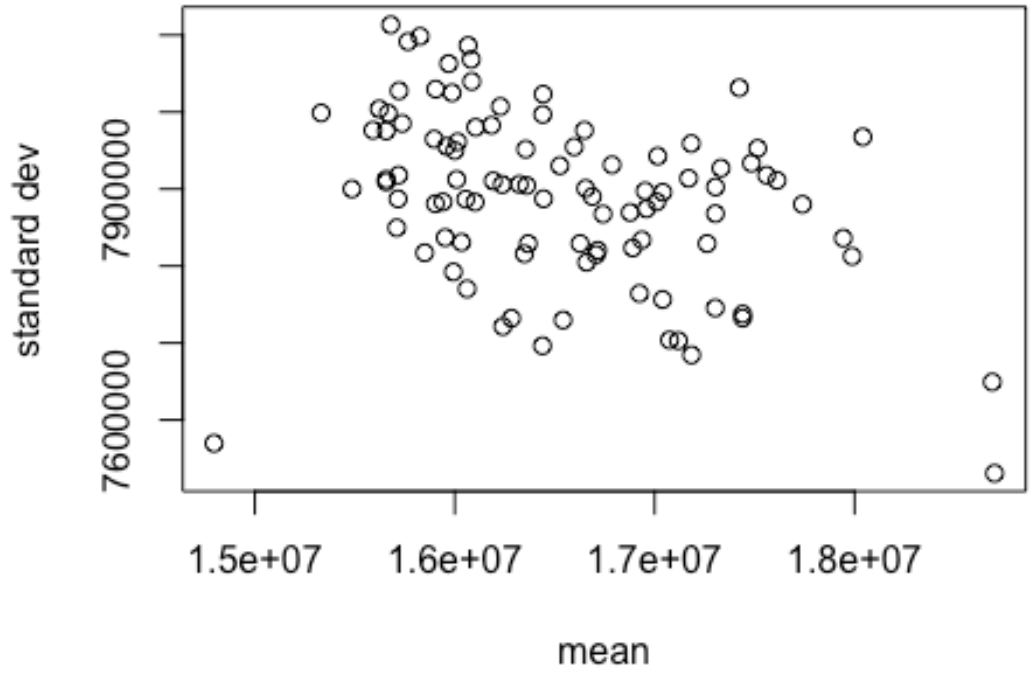
Data is dependent upon location and elevation with higher / west stations recording higher observation

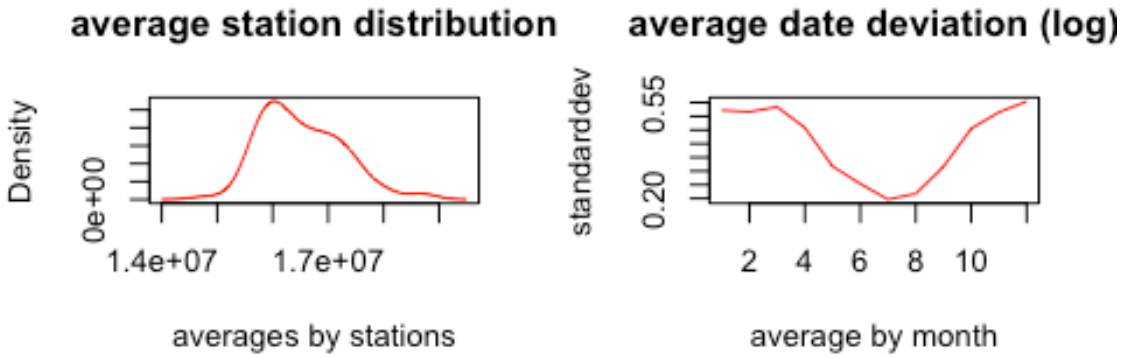
Correlation among stations

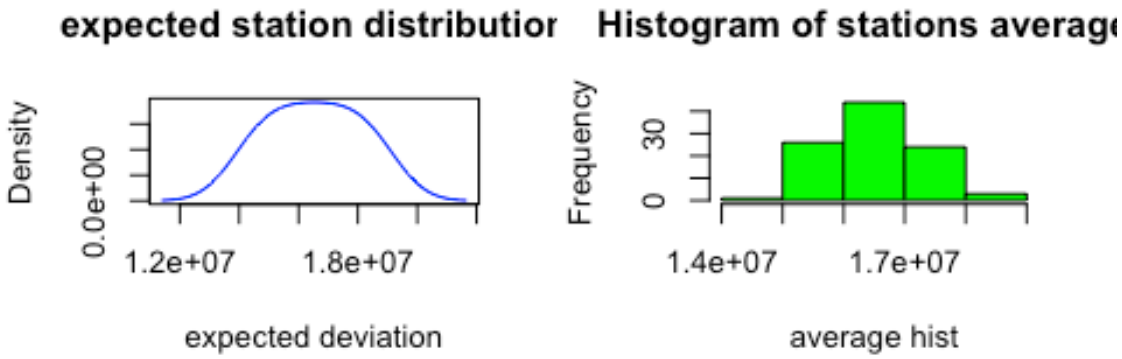
Following the analysis on their relative observation based on their location we thought we could establish some segmentation:



More on stations: distributions:







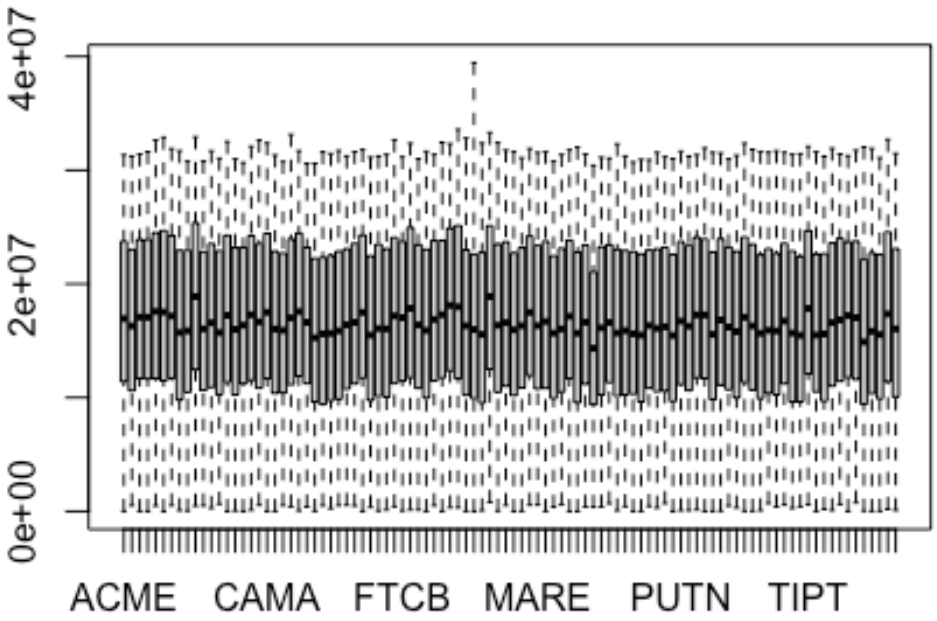
DATA PREPARATION

In order to obtain the results displayed above, we needed to prepare and the data:

- Finding NA ratio per column in order to see their potential influence the analysis
- We needed to find and map the extreme outliers

Outliers

Finding outliers



DATA PROCESSING

Once we had our outliers mapped, we replaced them with NA values in order to decrease the potential bias

Then once we have tackled the outliers, it was time to split the data and start training our model

Splitting into training and testing sets.

We will be training our model with 70% of the original data, valuating and then testing with 15% each.

```
train_index <- sample(1:nrow(solar_data), 0.7*nrow(solar_data))
val_index <- sample(setdiff(1:nrow(solar_data), train_index), 0.15*nrow(solar
_data));
test_index <- setdiff(1:nrow(solar_data), c(train_index, val_index))
```

BUILDING THE MODEL

While looking for our top performing model, we looked at the variety of different algorithms that could be used for this purpose, including **xgboost** and **svm**. After testing multiple models, we came to the conclusion that **auto.arima** was best suited for the purposes of this assigments, so we are going to showcase it below.

Although forecasting models might not have been the best choice for this kind of problem, we chose to go with auto.arima. The reason for this was, that we started with this part of the group-work before we had the classes on 'ML with R' and our intuition has guided us here.

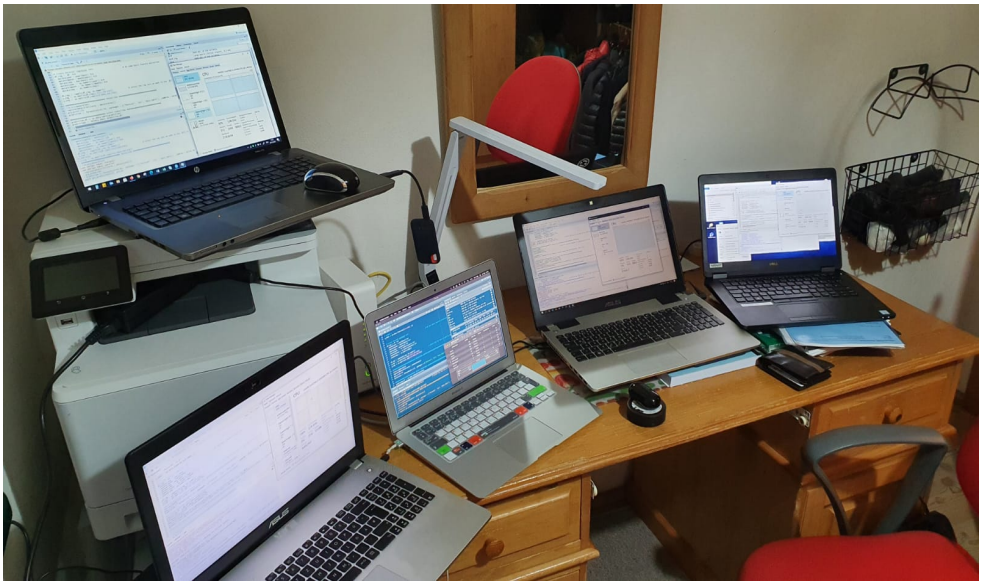
With this lack of knowledge, we went for a pipeline, that would allow for model validation via a system-call of the the Kaggle API from R. However, this means that our best model does not implement a proper train/test split, it does not make use of Hyper-Parameter Tuning or Cross Validation, and is probably far from efficient compared to other models.

After some experimentation with the models from the forecast package (incl. naive, ses, holt, arima, tbats, nnetar) we found that auto.arima resulted in the best scores. When we added additional external regressor variables, the score further improved. When we added the variables from the principal component analysis provided by the professor the score continued to improve, however the processing time also increased substantially.

To decrease processing time, we paralleled the prediction by rows, which resulted in an increase of performance.

The last run of this prediction happened on 8 cores and took about 40h. The scores achieved with auto.arima heavily depended on the amount of PCA-Variables used as external regressors.

What it took us to build this model



In case image doesn't work, see here

We are very excited for you to see our final model!