**Recruit Restaurant Visitor Forecasting**

*Author(s):*

Ayala Shvarzman

Arnon Kleinman

# Introduction

The running of a successful and profitable restaurant is not easy to say the least...

Even if you hired the right cooks and staff, the food is delicious, your location is prime, and your service level is excellent, you could still go under.

Many successful restaurants go belly up, it's a difficult to succeed in this business, and you need to save your money wherever you can.

We decided to make a predictive model that will help cut expenses and save money.

Our focus is the on number of customers a restaurant should expect each day, and by so to save a lot of money on buying excess raw materials that will be dumped at the end of the day, or by reducing the use of unnecessary labor, and by planning campaigns in advanced to attract customers on weaker days.

This paper examines the number of customers restaurants in japan have each day.

We were provided with data from two sites, "Hot Pepper Gourmet" (a restaurant review service) and "AirREGI" (a restaurant point of sales service).

"Hot Pepper Gourmet" (hpg): similar to Yelp, here users can search restaurants and also make a reservation online.

"AirREGI / Restaurant Board" (air): similar to Square, a reservation control and cash register system

# Methodology (Project design)

## Data

Here you have to describe how do you plan to manipulate the data. For this you have to answer to the following questions:

* Which data will be used?
  + Describe data sources
  + Describe possible external data sources that may enrich our data
  + Data for external validation?

The data includes number of visitors per restaurant per day, number of reservations throw the two sites, information about the restaurants, location, area, what kind of food it serves, we were also provided with information about the date, is it a holiday or not.

We added data from the web about weather and statistical data per prefecture (socioeconomic state, population density, etc...)

* On which time frames periods will your project will be based on?
  + Time-frame for training
  + Time-frame for test?
* How do you define your subjects?
  + Inclusion criteria?
  + Exclusion criteria?

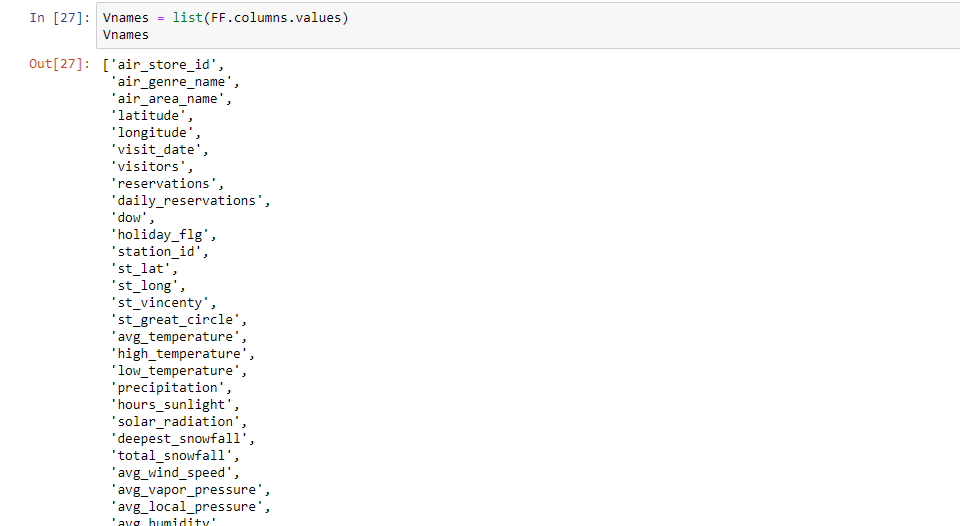
The project will be based on a time frame beginning of 2016 starching to middle 2017: The training data covers the dates from 2016 until April 2017. The test set covers the last week of April and May of 2017.

Inclusion criteria is a restaurant being included in the air\_stor\_info table

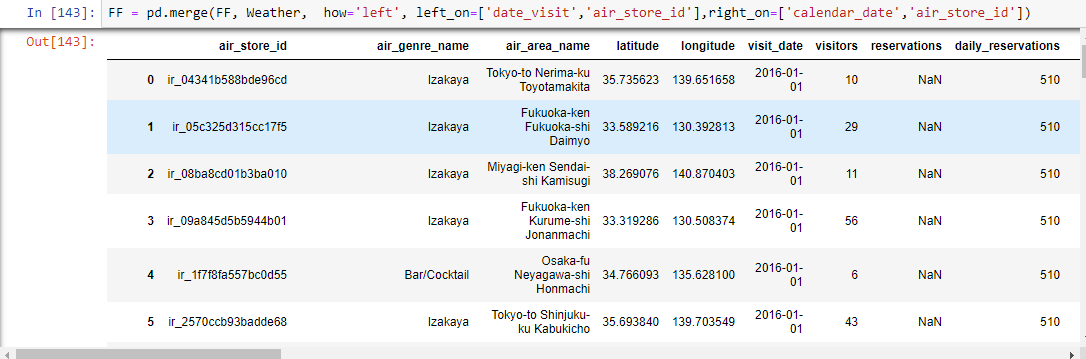
* Which would be your outcome variable?
* Are there confounder variables that may affect the outcome?
* Is there a possible source of bias in our data?
* Describe your data exploration strategy.
* Which techniques will be applied to enrich the data?
* How you will deal with outliers?
* How you will deal with missing values
* Add at the end of the protocol (appendix) the [Data retrieval protocol](https://docs.google.com/spreadsheets/d/1pYYjgwZ_8PS1Bcmc2kRNHTL0f_rk__GCJALLs1JHPUQ/edit#gid=0)

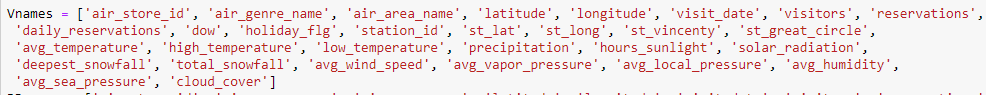
**Exploratory data analysis**

That is our flat file, before adding one hot encoding categorization:



Merging relevant columns we got:

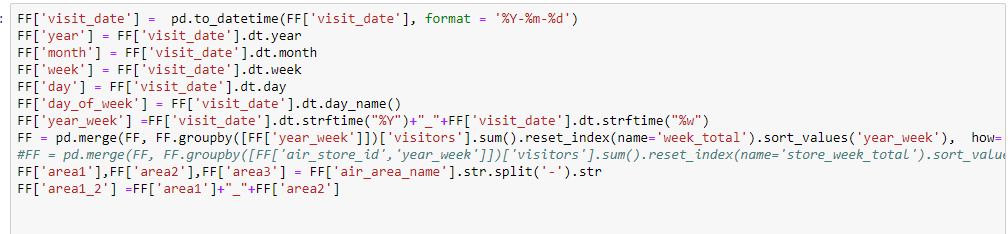




We had to change the type of our date column to date instead of datetime:

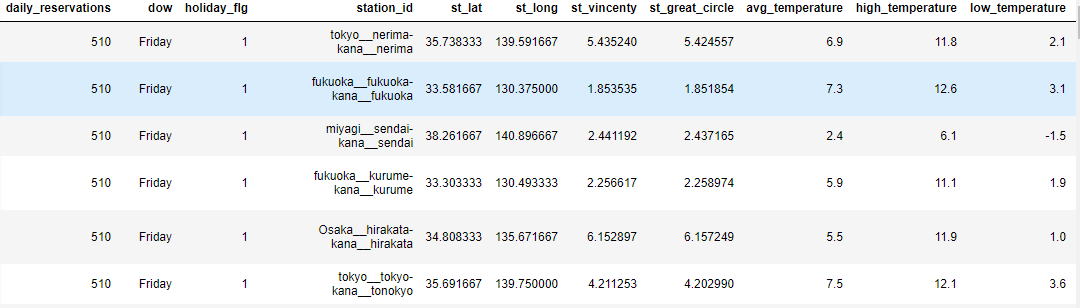


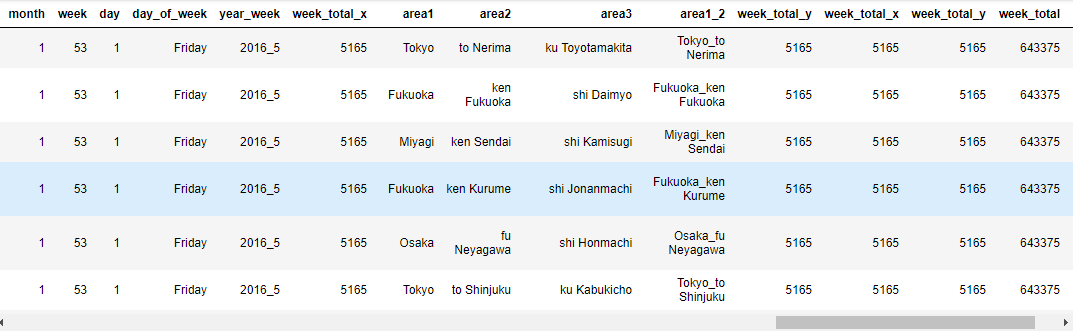
After dealing with datetime to date types issues in our Jupiter notebook, we switched its type:



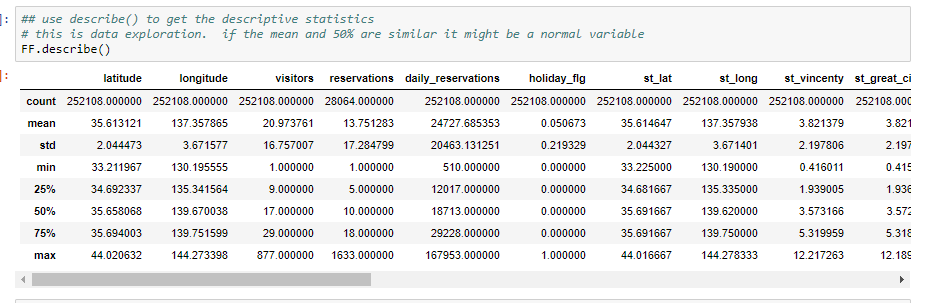
Finally, we got the following flat file table:

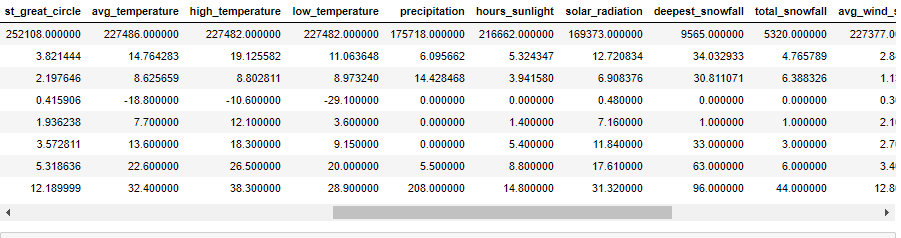


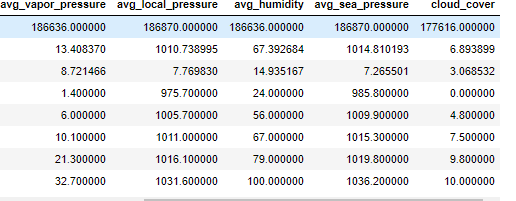




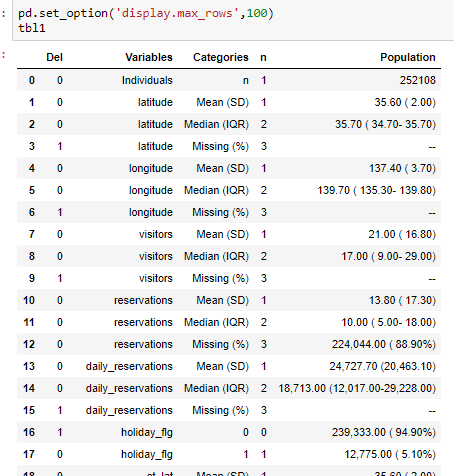
FF.describe() yiealds:



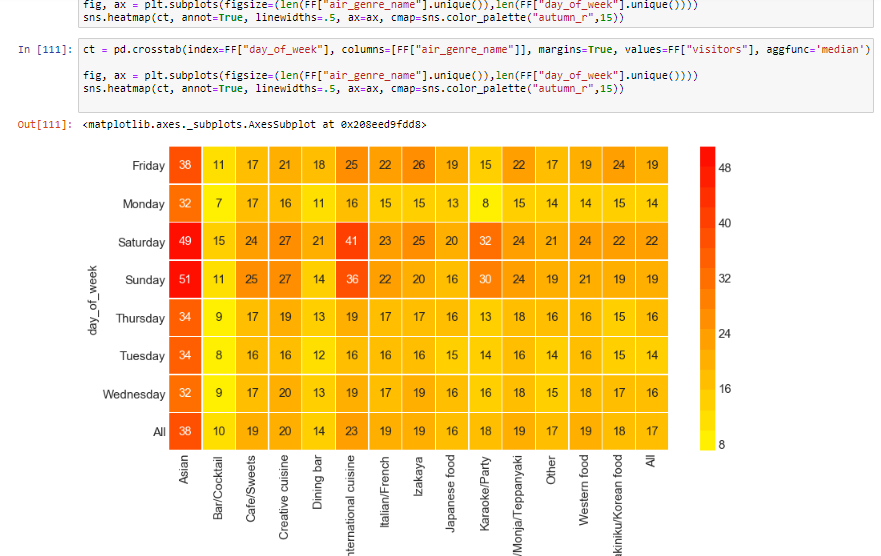




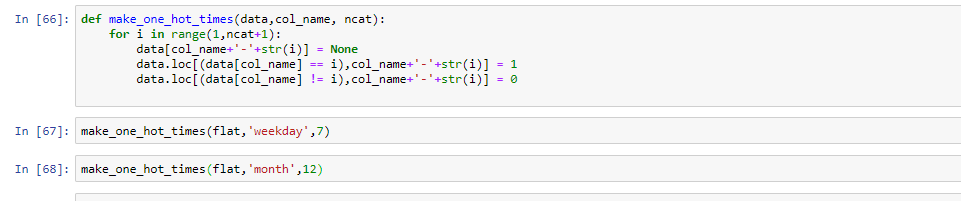
tbl1 = mechkar.pyMechkar().Table1(data=FF)



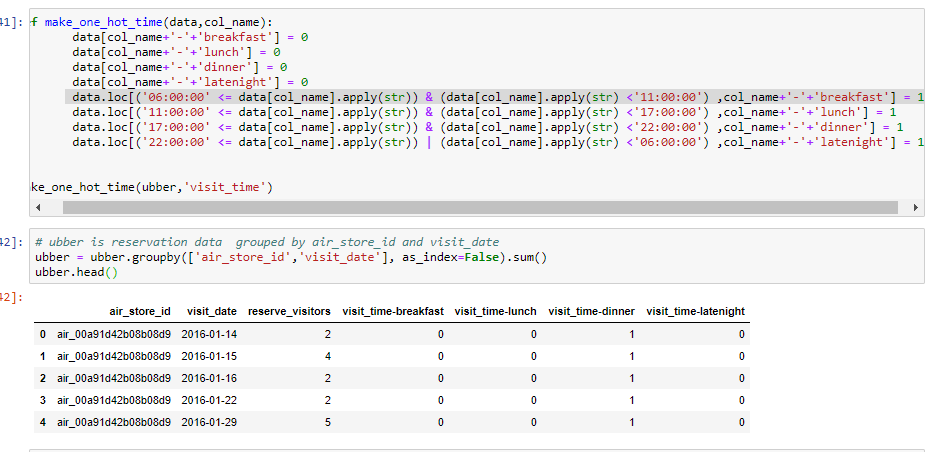
The following was a try we did to put in a table the median number of visitors as a function of genre and day of week, in order to study the traffic changes over these parameters:



After that, we used the following two one hot encoding functions to switch our categorical variables to binary values (specified above):

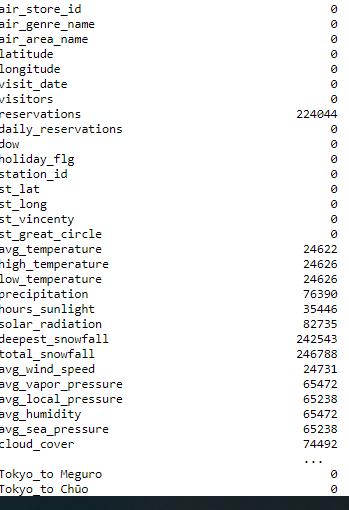


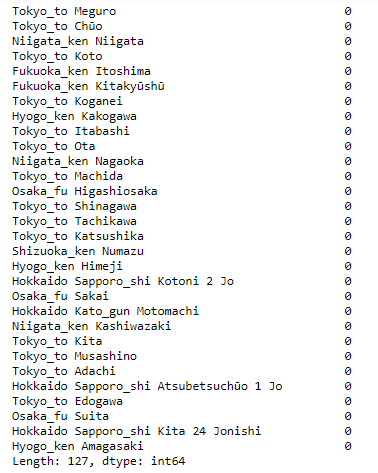
We also divided, using one hot encodig, the visit time to four categories, morning, lunch, dinner and latenight:



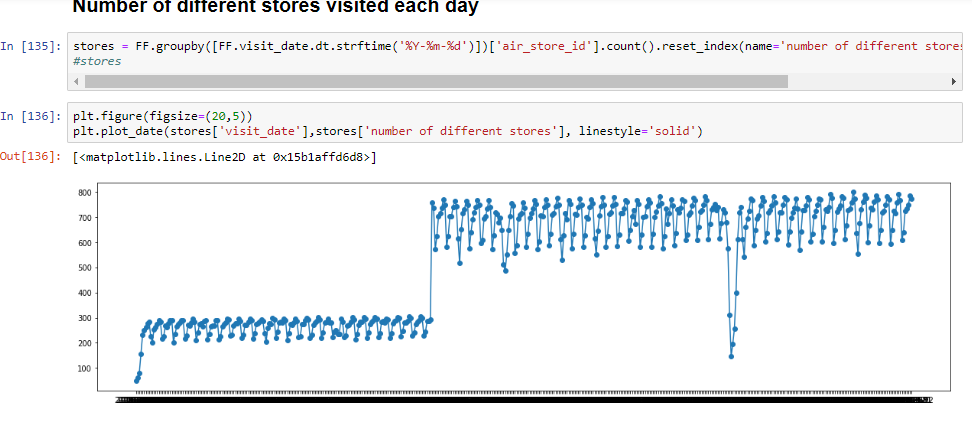
Now, we explored missing data in our dataset using isna(FF) function as follows:

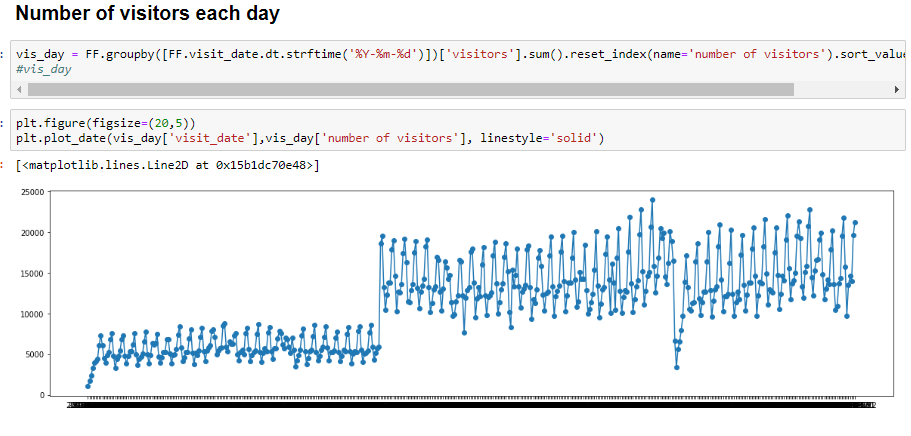


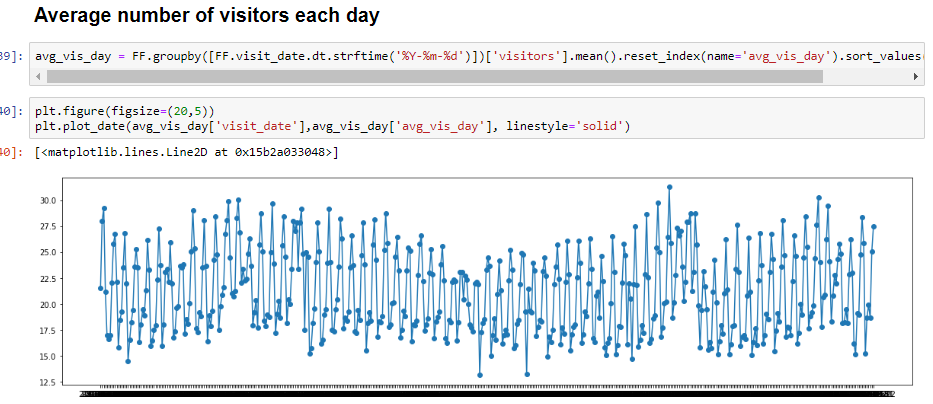




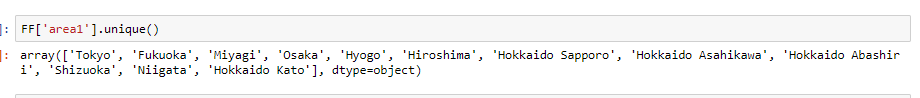
The following statistics were made on our flat table, FF, to study the data:

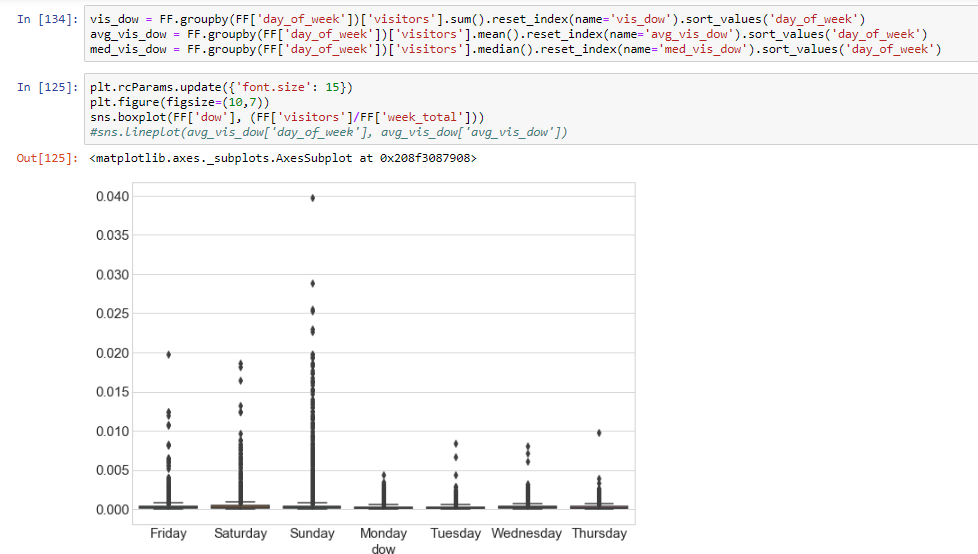


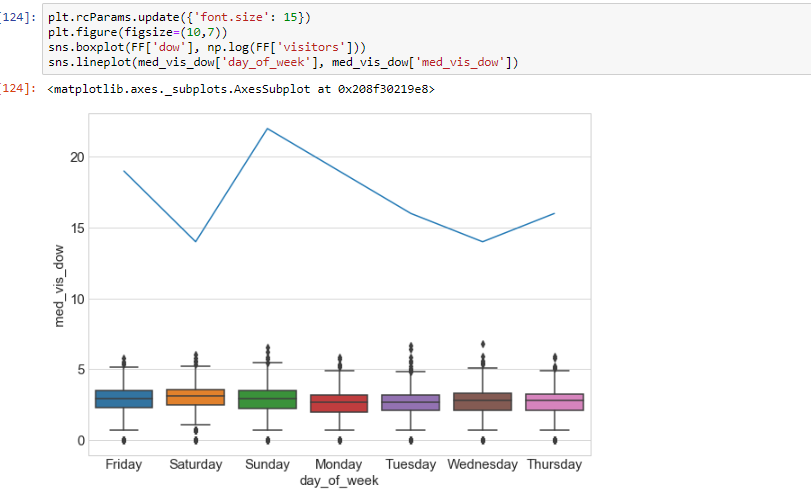




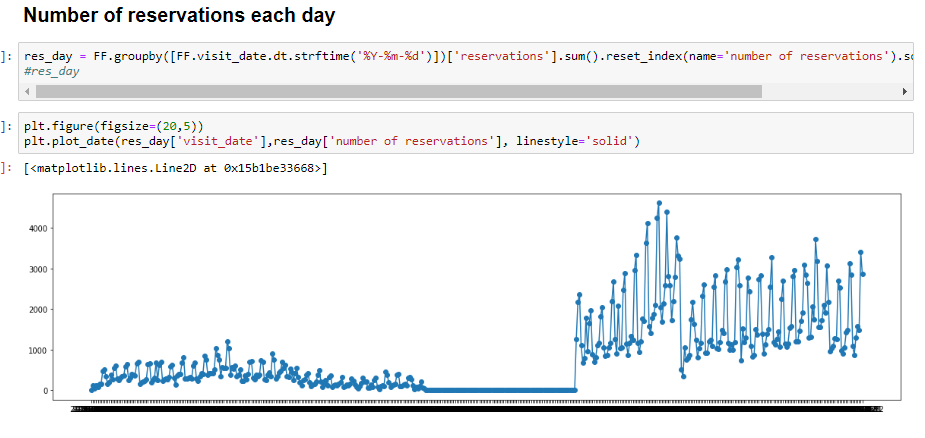
Dividing the region column and focusing on the first part, the county:

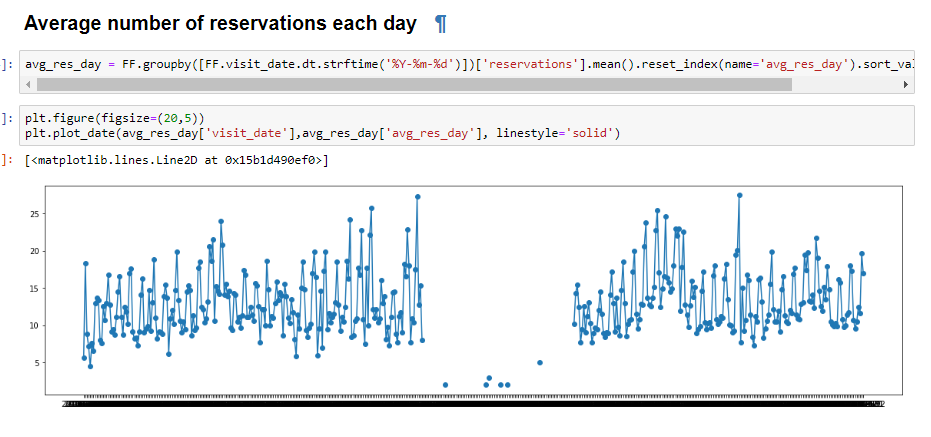




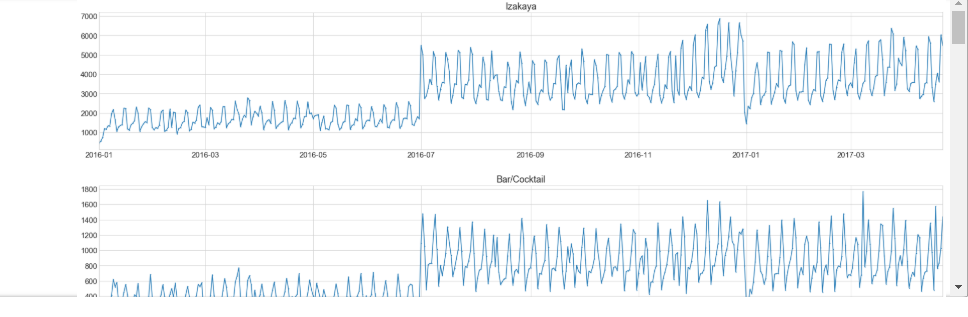


Studying daily statistics:

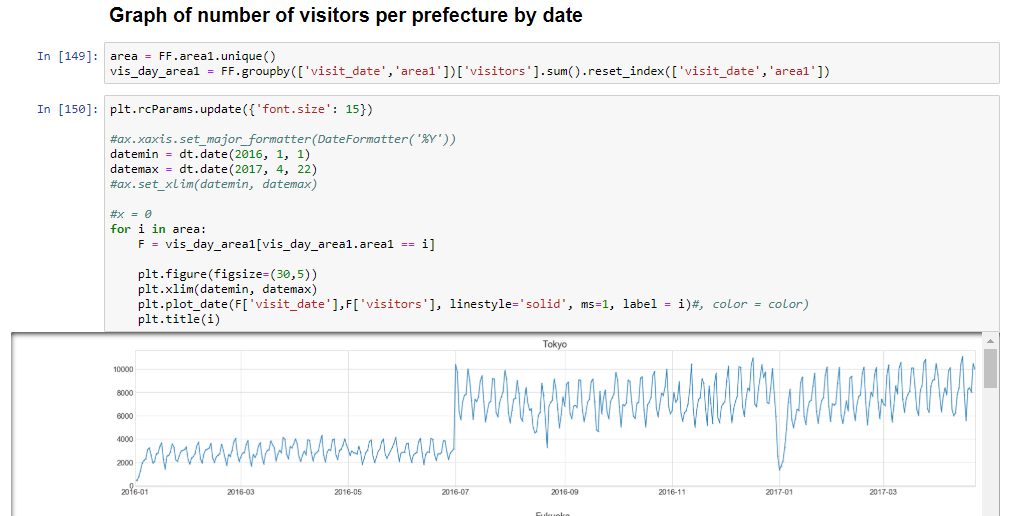








We have 14 of these, for each restaurant genre…



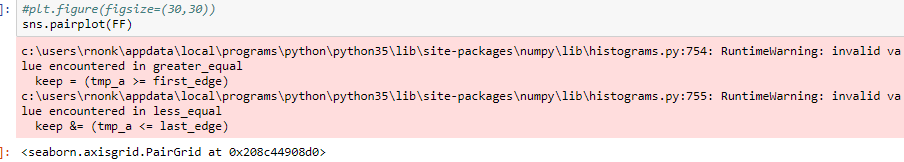
### We see that some fields have large precentages of missings So the weather fields with high precentage of missing ('deepest\_snowfall' 96% and 'total\_snowfall' 97%) Cannot be used, and will be removed! The reservations field that have 88% of missing, but is too important to be ignored.

Thus, after analyzing the data, we concluded the need to separate it to two models:

1st for all the stores, but with no reservations data, 2nd for only stores that have reservation data.

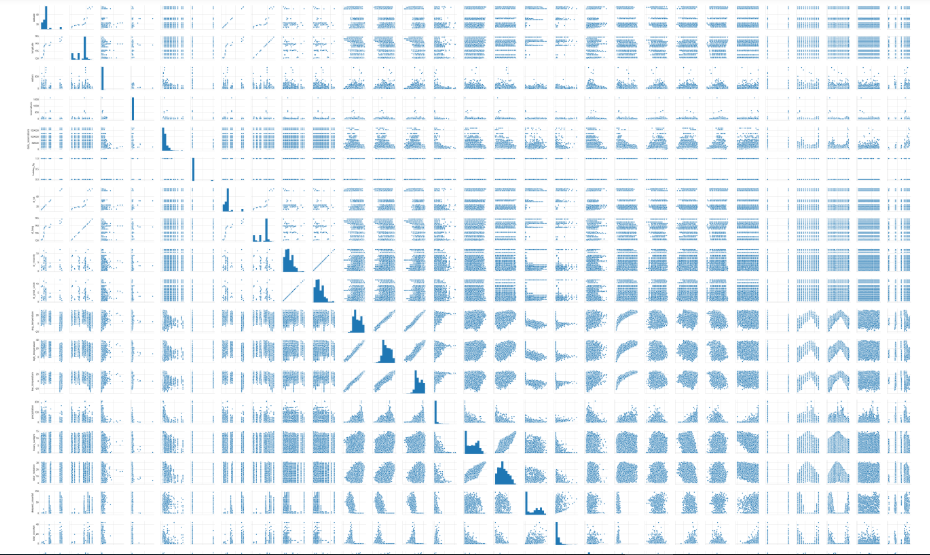
Thus, we added "has reservation data" flag column to our flatfile.

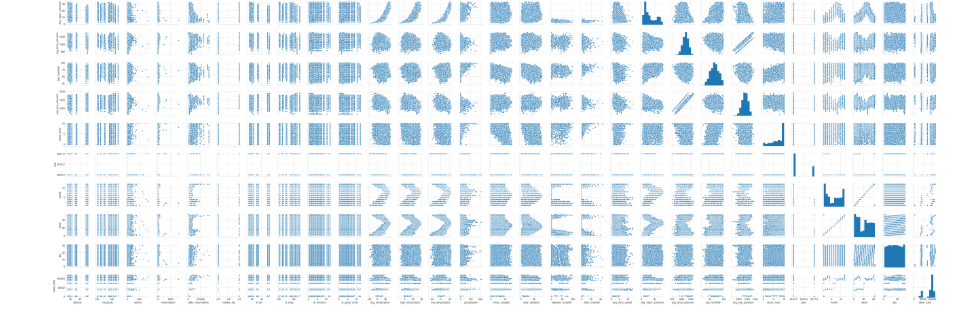
Pairplot analysis:





sns.pairplot(FF[Ordinal\_Fields])





**Clear outcome variable definition:**

the y variable, which we wish to predictt, is defined as the future restaurant visitors totals on a given date. We will provide, as a result, a table of three rows: restaurant id, date and number of visitors.

Several sources of bias exist in our data: the golden week, which is a national holiday in Japan and occurs at mid 2016. Holidays are a source of bias in general. That is why we have a special categorical variable tagging a date as a holiday or not.

**Variable Engineering:**

**Original data**

Our raw training data consists of about ~252,000 rows of data, and our key is a combination of date and restaurant id. About 20,000 rows included reservation data of some sort. Because so many reservation data was missing, we decided to divide our data to two different sets, meaning two different flat tables. One included reseration data , one not.

The columns of our flat file are the following: air\_store\_id, visit\_date, area, air\_genre\_name, several temperature and weather relatev columns (specified in our data retrieval protocol), holiday\_flag (yes or no ), reservations, year, weekday month.

We analyzed the data and several of our initial columns were divided using one hot encoding to categorical variables (will be discussed above at the variable engineering section).

**One hot encoding**

In addition, we also split categorical variables into binary indicator variables, as one of our models (GLMNet) could not handle categorical data. These variables include:

Weekday, split to 7 categories, day of month split to 12 categories, genre split to 14 types of restaurant genres, and finally the region (location of the restaurant) split to 14 different genres.

We will further explain in detail our one hot encoding process on the next chapter (feature and variable selection).

**Outliers determination and treatment:**

**FF\_NR** : data (flatfile) which doesn’t include reservations statistics

**FF\_OR**: data (flatfile) which includes reservations statistics information

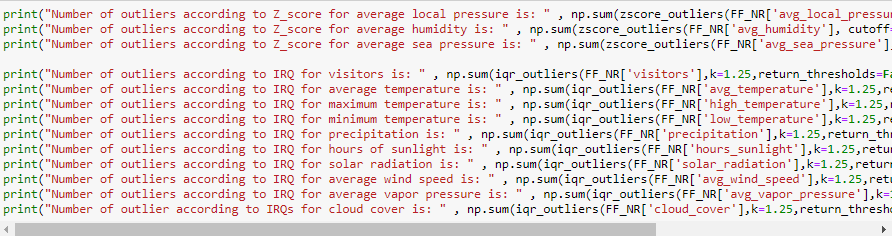
## **Univariate outlier detection**

We used the first function for variables of normal distribution and the second for non-normal distribution:

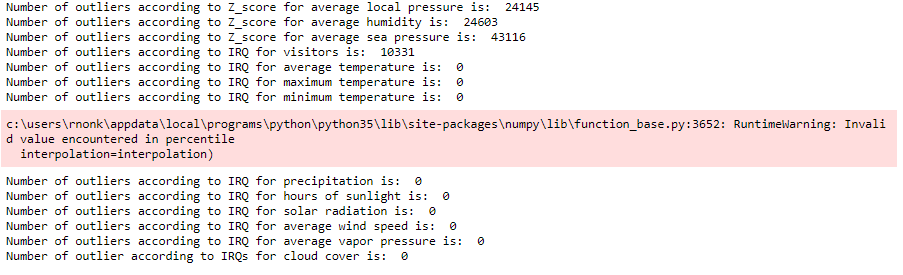
def zscore\_outliers(x, cutoff=3.0, return\_thresholds=False)

def iqr\_outliers(x, k=1.5, return\_thresholds=False)

### According to the pairplot observations, we made outliers tests. We found that the fields: 'avg\_local\_pressure', 'avg\_humidity', 'avg\_sea\_pressure' distribution is normal.



## 

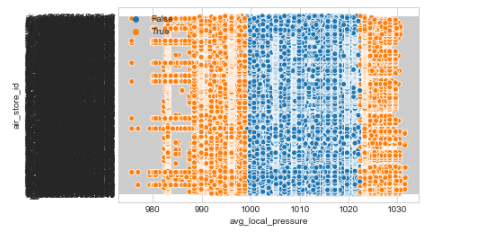


out1=zscore\_outliers(FF\_NR['avg\_local\_pressure'], cutoff=1.5, return\_thresholds=False)

sns.scatterplot(x=FF\_NR['avg\_local\_pressure'],y=FF\_NR['air\_store\_id'],hue=out1)

print("Number of outliers according to Z\_score for average local pressure is: " , np.sum(zscore\_outliers(FF\_NR['avg\_local\_pressure'], cutoff=1.5, return\_thresholds=False))

Number of outliers according to Z\_score for average local pressure is: 24145

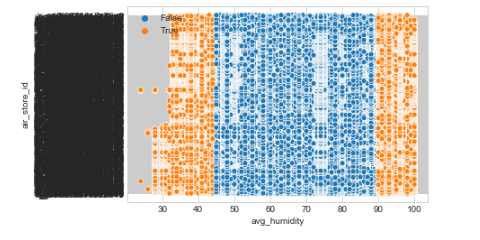


out2=zscore\_outliers(FF\_NR['avg\_humidity'], cutoff=1.5, return\_thresholds=False)

sns.scatterplot(x=FF\_NR['avg\_humidity'],y=FF\_NR['air\_store\_id'],hue=out2)

print("Number of outliers according to Z\_score for average humidity is: " , np.sum(zscore\_outliers(FF\_NR['avg\_humidity'], cutoff=1.5, return\_thresholds=False)))

Number of outliers according to Z\_score for average humidity is: 24603

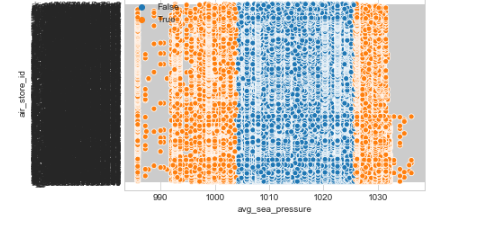


out3=zscore\_outliers(FF\_NR['avg\_sea\_pressure'], cutoff=1.5, return\_thresholds=False)

sns.scatterplot(x=FF\_NR['avg\_sea\_pressure'],y=FF\_NR['air\_store\_id'],hue=out3)

print("Number of outliers according to Z\_score for average sea pressure is: " , np.sum(zscore\_outliers(FF\_NR['avg\_sea\_pressure'], cutoff=1.5, return\_thresholds=False)))

Number of outliers according to Z\_score for average sea pressure is: 24415

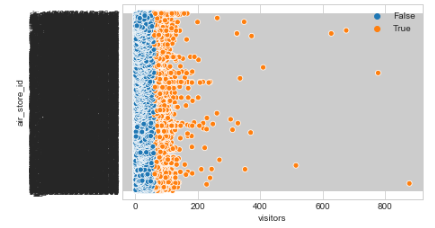


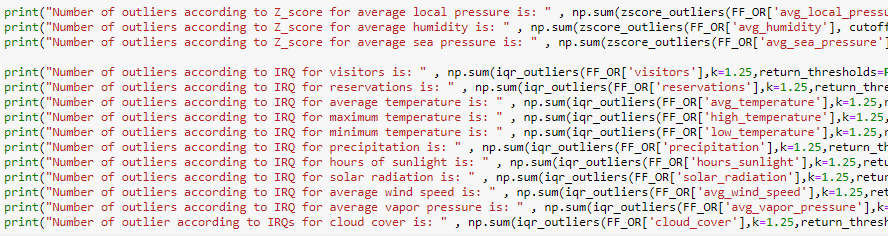
outNR\_visitors=iqr\_outliers(FF\_NR['visitors'],k=1.5,return\_thresholds=False)

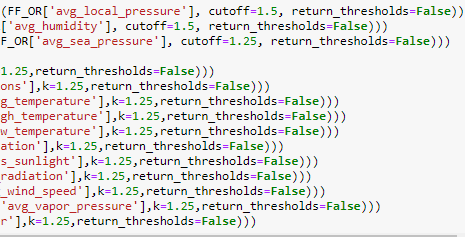
sns.scatterplot(x=FF\_NR['visitors'],y=FF\_NR['air\_store\_id'],hue=outNR\_visitors)

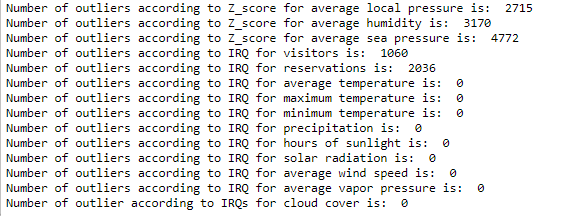
print("Number of outliers according to IRQ for visitors is: " , np.sum(iqr\_outliers(FF\_NR['visitors'],k=1.5,return\_thresholds=False)))

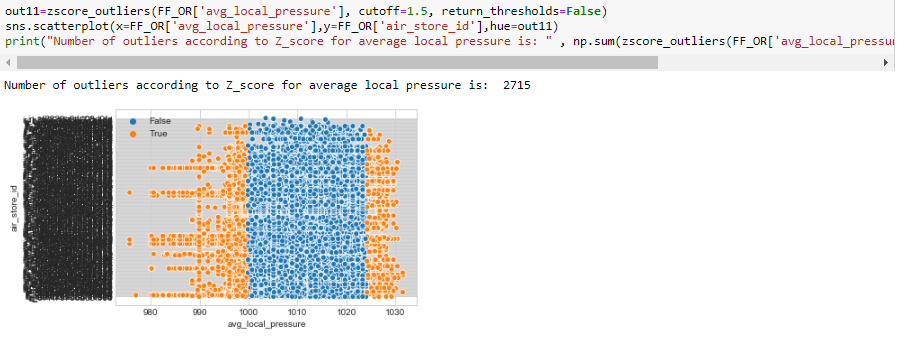
Number of outliers according to IRQ for visitors is: 6959

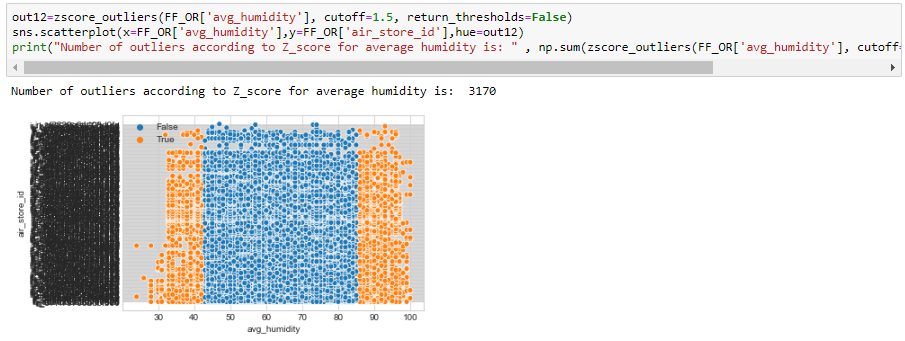


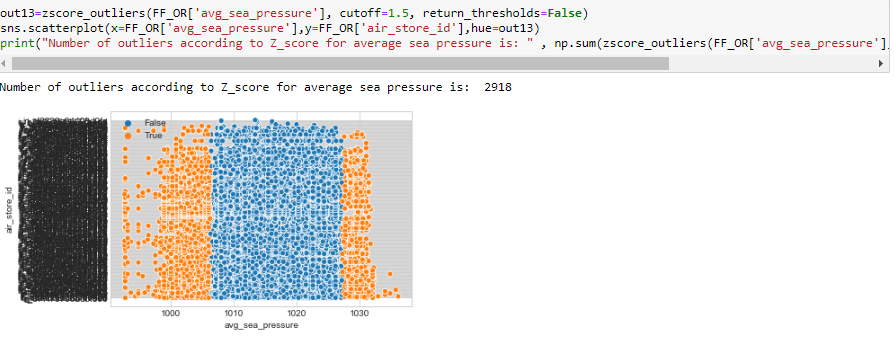










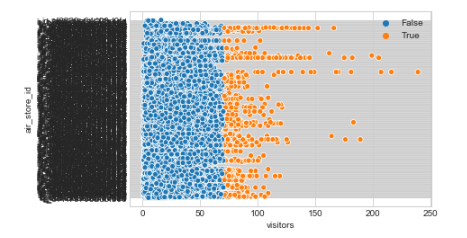


outOR\_visitors=iqr\_outliers(FF\_OR['visitors'],k=1.5,return\_thresholds=False)

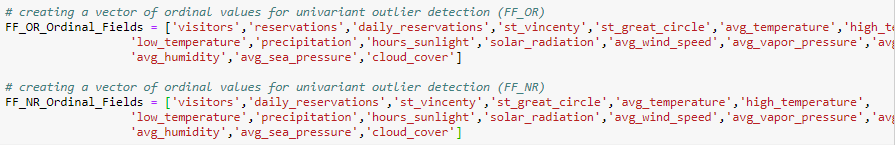
sns.scatterplot(x=FF\_OR['visitors'],y=FF\_OR['air\_store\_id'],hue=outOR\_visitors)

print("Number of outliers according to IRQ for visitors is: " , np.sum(iqr\_outliers(FF\_OR['visitors'],k=1.5,return\_thresholds=False)))

Number of outliers according to IRQ for visitors is: 673



## Statistical Tests



## Multivariate Test

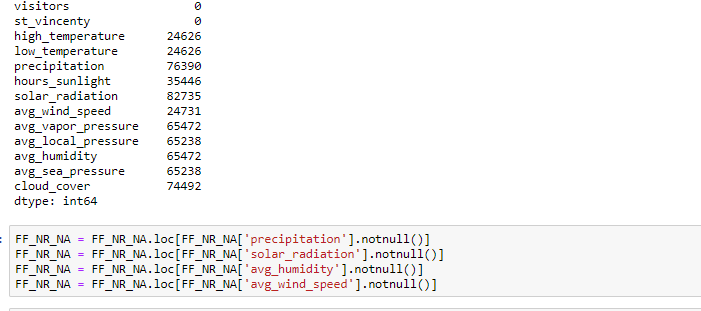
def dbscan\_mvoutliers(X):

from sklearn.cluster import DBSCAN

from sklearn.preprocessing import StandardScaler

#### **Running dbscan on no reservations faltfile model:**



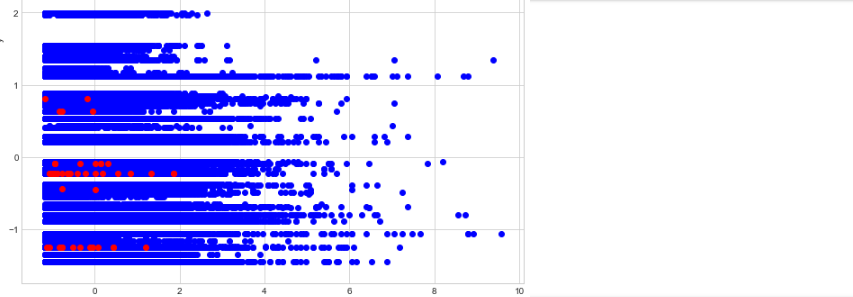


## 

## 

DBSCAN(algorithm='auto', eps=3.0, leaf\_size=30, metric='euclidean',

metric\_params=None, min\_samples=10, n\_jobs=None, p=None)

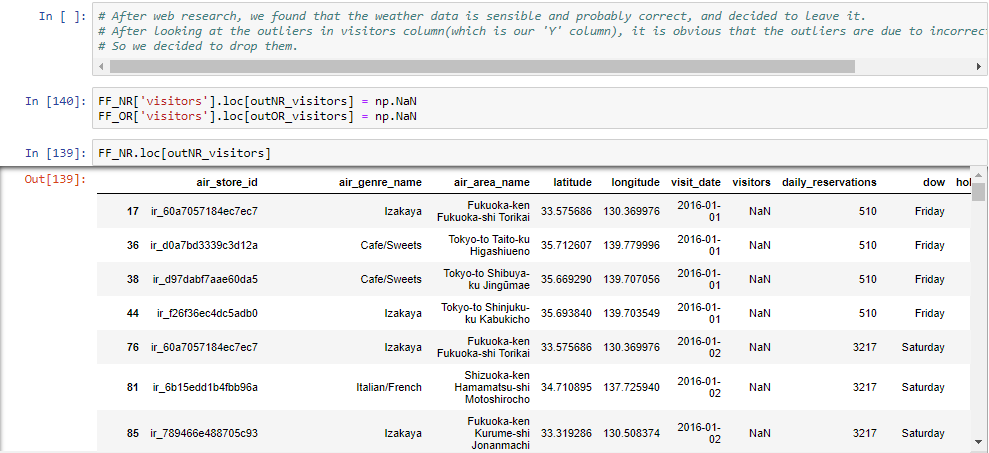


## 

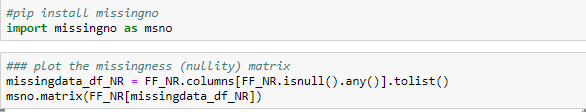
## 

## 

## Treating outliers



# Missing Values



## 

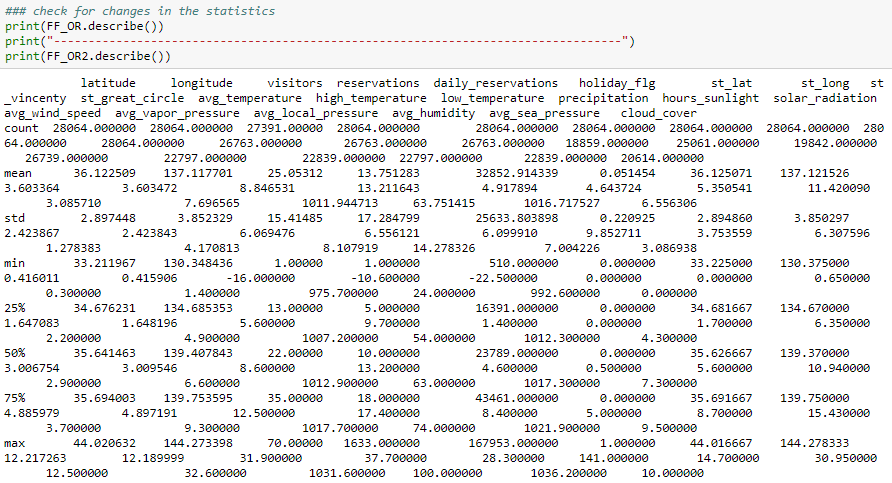
## 

## Missing Values Imputation

### Dropout

## 





## Models

Here you have to describe how do you plan to develop your models:

* How do you plan to divide your data
  + Training, validation, test - proportions, techniques
* Do you need to balance your data? How?
* Do you need to stratify/subsample your data? How?
* What techniques will you apply to model your outcome?
  + Unsupervised
  + Regression
  + Classification
* Will you use cross-validation and/or bootstrap?
* Which measures you will use to train and evaluate your models? Why?

Our target optimization metric is the Root Mean Squared Logarithmic Error.

The RMSLE is calculated as:

Where:

**n** - is the number of observations

**pi** - is our predicted count

**ai** - is the actual count

**log(x)** - is the natural logarithm of **x**

We seek to identify the models that result in predictions which minimize this error.

* Do you plan to use ensembling or will use your best model?

## Deployment of your model

* Who will make the QA of the project?
  + Which units will be assessed
  + Write a QA protocol for each step of the project
* Who is the final user of the predictions?
* How the prediction will be presented to the final user?
* How will the final user be trained to use and interpret the prediction?
* On which platform the predictions will be deployed?
* How frequently the model will be updated?
* What will happen in cases where the model return a null prediction (eg. incomplete data)?
* Which models were used and which were selected for the final prediction.
* Which measurements were used to evaluate the prediction.
* Which results we got from those models.

# Results

Here you will present the main results of all the process. We will describe:

* The final amount of data used (total, train, test, etc)
* The amount of outliers and the way of treating them,
* The amount of missing values and the methods used for imputing them,
* The distribution of the data (timeframes)
* The methods used to transform the data and to generate new features.

# Conclusion

Here you will write about how the project began, which were the most important challenges you had when developing the project, and how did you get the final prediction. You have to discuss also the limitations of the model, when it can be used and when not.