

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

10-8-2022

Recruit Restaurant Visitor Forecasting

Final Project

Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

ALEJANDRO C. PARRA GARCIA
PREDICTIVE MODELS

Table of Contents

Problem	2
Significance.....	2
Literature.....	2
Data Processing and Description.....	2
Types of Models	14
Models/Formulation	14
Performance / Accuracy.....	15
Limitations.....	18
Future Work	18
What was learned/Conclusions.....	19
References.....	19

Problem

Owning a restaurant can be difficult due to the uncertainty of not knowing the number of future clients that you will receive, and not being able to plan accordingly. This can result in having excess food and excess staff or not having enough.

The objective of this project is to build a model capable of predicting future visitors to the restaurant

Significance

If the model built are able to reliably predict future number of visitors with a certain degree of accuracy, that will mean that restaurant will be able to plan accordingly saving in resources, as they won't need to over-buy food that end up wasted. This will also increase the attention to the client as they can hire the exact number of staff needed.

Literature

From reading the book we can get some insight of the models that could be used in order to create the forecast, models such as ARIMA, linear regression, multiple linear regression, Dynamic regression, Neural network model... I will explore more about these models in the Types of models section.

For now we can search to look for other people and researchers approaches in solving this problem, we fist can take a look into the EDA performed by others through the Kaggle discussion section, since they did a wonderful work, we can find this in the discussions 'Be my guest - Recruit Restaurant EDA' (HEADS OR TAILS, 2020) and 'A Very Extensive Recruit Exploratory Analysis' (TROY WALTERS, 2017) we can see from their analysis that the genre of the restaurant has in some cases some impact on the number of visitors, the same can be said about the holiday flag, as it tends to increase the number of visitors if the Holliday was in middle of the week, while there isn't any impact if the holiday falls in the weekend. We can also see that there also a positive relation between the number of reservation and the visitors this is expected since the reserves translate into actual visitors. I will take a look at the data myself in the next section, 'Data processing and description'.

In relation with these discussions, we can see that the 'A Very Extensive Recruit Exploratory Analysis' was pre EDA and didn't talked about which model to use, while the other decided to use the ARIMA model to create predictions, this is reinforced by this scientific paper (Boomija, Anandaraj, Nandhini, & Lavanya, 2018) where the researchers used ARIMA in order to build the model to make the forecasting.

Data Processing and Description

The data given comes from two sources, AirREGI / Restaurant Board (air) and Hot Pepper Gourmet (hpg).

The data is presented in multiple datafiles:

sample_submission.csv: For submission, contains id + date and the visitors

date_info.csv: it doesn't contain any information about reservation or visitors. It contains the days as date format, the day in the week (Monday, Tuesday...) and a flag variable if it is a holiday.

air_visit_data.csv: Contains the actual number of visitors for a restaurant and date

air_store_info.csv: Contains information about the restaurant, like name, type, latitude, and longitude. For AIR data

hpg_store_info.csv: Contains information about the restaurant, like name, type, latitude, and longitude. For HPG data

air_reserve.csv: Contains the reservations done for a given day, the day the reservation was made, and the number of people. For AIR data

hpg_reserve.csv: Contains the reservations done for a given day, the day the reservation was made, and the number of people. For HPG data

store_id_relation.csv: Contains the relation between both sources of data

So, we need to join all the files into one dataframe in order to make predictions.

The reserve files include the hour of the reservation. So first we need to aggregate the data to daily.

Then we need to check the number of distinct restaurant id and distinct days each csv has.

file	Nº Restaurants	Nº of days	Start	End
date_info.csv		517	2016-01-01	2017-05-31
air_visit_data.csv	829	478	2016-01-01	2017-04-22
air_store_info.csv	829			
hpg_store_info.csv	4690			
air_reserve.csv	314	433	*	*
hpg_reserve.csv	13325	517	*	*
sample_submission.csv	821	39	2017-04-23	2017-05-31
store_id_relation.csv	150			

* Only have the date for which there is info, between 2016-01-01 and 2017-05-22

So, we only care for the 821 restaurants from the sample_submission.csv file. That means that we can ignore the rest. Since the time length is 517 days. That means that the resulting df is 424,457 rows. But we need to remember that we need to train it with only the first 478 days (392,438 rows), and the make the prediction for 39 days (32,019 rows)

So, now we need to merge all of this files.

First let's build one df with only the dates and the air_id of the submissions and the corresponding hpg_id. On that then add the rest of the files. The resulting dataframe has the following columns:

Columns	Unique elements	Nº of nulls	% Of missing value
date	517	0	0%
day_of_week	7	0	0%
holiday_flg	2	0	0%
air_store_id	821	0	0%
hpg_store_id	151*	346,907	81.17%
id	32,020*	392,438	92.45%
visitors_Pred	2*	392,438	92.45%
visitors	204*	173,989	40.99%
air_reserve_visitors	126*	394,627	92.97%
hpg_reserve_visitors	80*	405,837	95.61%
air_genre_name	14	0	0%
hpg_genre_name	17*	391,886	92.32%

* The function that count the unique elements, count the nulls as well, that means that the true number of unique, non-null, elements is the number given minus one.

Description:

- **Date:** goes from 2016-01-01 to 2017-05-31
- **Day_of_week:** goes from Monday to Sunday. (Origin: date_info.csv)
- **Holiday_flg:** 1 if holiday 0 otherwise. (Origin: date_info.csv)
- **air_store_id:** Id of the restaurant in the air format. Only used the id from the sample_submission.csv file, since those are the only ones that we need to make forecast on.
- **hpg_store_id:** Id of the restaurant in the hpg format. Only the ones that matched the air_store_id using the store_id_relation.csv file. Since that file only have 150 relations, that means that from the original 821 restaurant at most 150 are going to have relations. We can see, from the table, that the number of unique ids is 150. That means that 81.72% of the restaurant do not have this id. This means that all the data that comes from the hpg source is going to be missing in at least 81.72% of the cases.
- **id:** This is the id of for the sample_submission.csv, it combines the air_store_id and the date, separated by an '_'. This column has missing values since it only has values for the time between 2017-04-23 and 2017-05-31. It is this way since those are the days that needs to be predicted, this means 39 days. Since it is a combination of the id and the date, $39 \text{ (days)} * 821 \text{ (unique ids)} = 32,019$ different ids (Plus one from the nulls). There are $517 - 39 = 478$ days without this column, so, $478 * 821 = 392,438$ rows are missing the value
- **visitors_Pred:** This one follows the same logic as the id, as it comes also from the sample_submission.csv file. It has the same number of nulls. And the unique values are the 0 and the null. This column can be eliminated since it doesn't add any information for the model

- **visitors:** Contains the number of visitors for a given restaurant and a given day. (Origin: air_visit_data.csv). We can see that 40.99% of the data is missing, but we need to take into account that the last 39 days are the ones that we need to predict, this means that we are not going to have data for this time period. This means that $39 \times 821 = 32,019$ rows don't count. The actual missing values from 2016-01-01 to 2017-04-22 are: $[173,989 \text{ (Prev. \#null)} - 32,019 \text{ (forecast \#null)}] / [424,457 \text{ (\#rows)} - 32,019 \text{ (forecast interval)}] = 0.3617$ or 36.17% of missing values. This number is less, but it is still quite big for the prediction variable
- **air_reserve_visitors:** This are the reservation made for a given day and restaurant. (Origin: air_reserve.csv). I'm going to suppose that the nan values are because there weren't any reservations for that day.
- **hpg_reserve_visitors:** This are the reservation made for a given day and restaurant. (Origin: hpg_reserve.csv). I'm going to suppose that the nan values are because there weren't any reservations for that day.
- **air_genre_name:** Genre of the restaurant. (Origin: air_store_info.csv)
- **hpg_genre_name:** Genre of the restaurant. (Origin: hpg_store_info.csv). I'm going to drop this column since it has a lot of missing values and since we already have the air_genre_name.

Now I drop the hpg_store_id, visitors_Pred and hpg_genre_name. The reservations columns I change the Na to 0, since I supposed that if there is no data that means that they weren't any reservations. A new column was created with the sum of the air and hpg reservation. Also, since we need to build the model. Only the data from 2016-01-01 to 2017-04-22 will be used for now.

So this is the final data frame:

Columns	Unique elements	Nº of nulls	% Of missing value
date	478	0	0%
day_of_week	7	0	0%
holiday_flg	2	0	0%
air_store_id	821	0	0%
id	1**	392,438	100%
visitors	204*	141,970	36.17%
air_reserve_visitors	125	0	0%
hpg_reserve_visitors	80	0	0%
total_reserve	141	0	0%
air_genre_name	14	0	0%

* The function that count the unique elements, count the nulls as well, that means that the true number of unique, non-null, elements is the number given minus one.

** The id has now only 1 value because since we have eliminated the rows after 2017-04-22, the id column is now all Na. This column is not going to be used for training the model. Its only used in the extended dataset that features those dates, and that is going to be use for submission.

Now the data frame looks way more promising. Since the number of nulls have been greatly decrease, in the phase of building the model the id column can be ignore, the only problem remaining is the Visitors column. That is because that column is the one that we want to make forecast for, and we are missing around 36 % of the data.

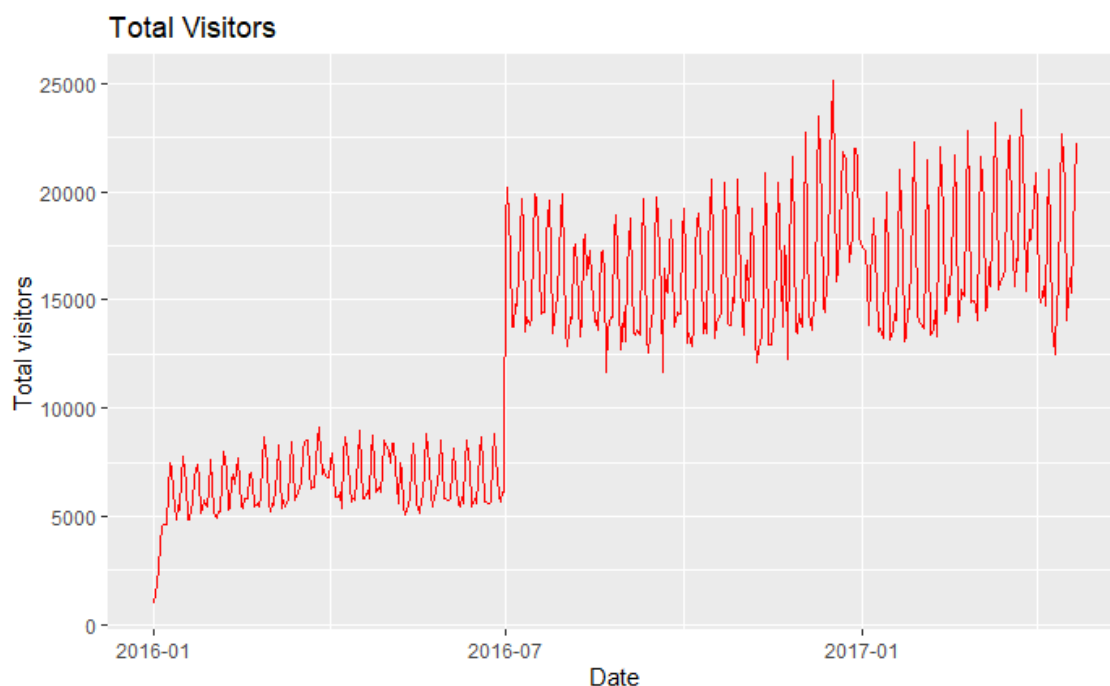
In order to reduce the amount of missing datapoints we eliminate the NA from the start of the series.

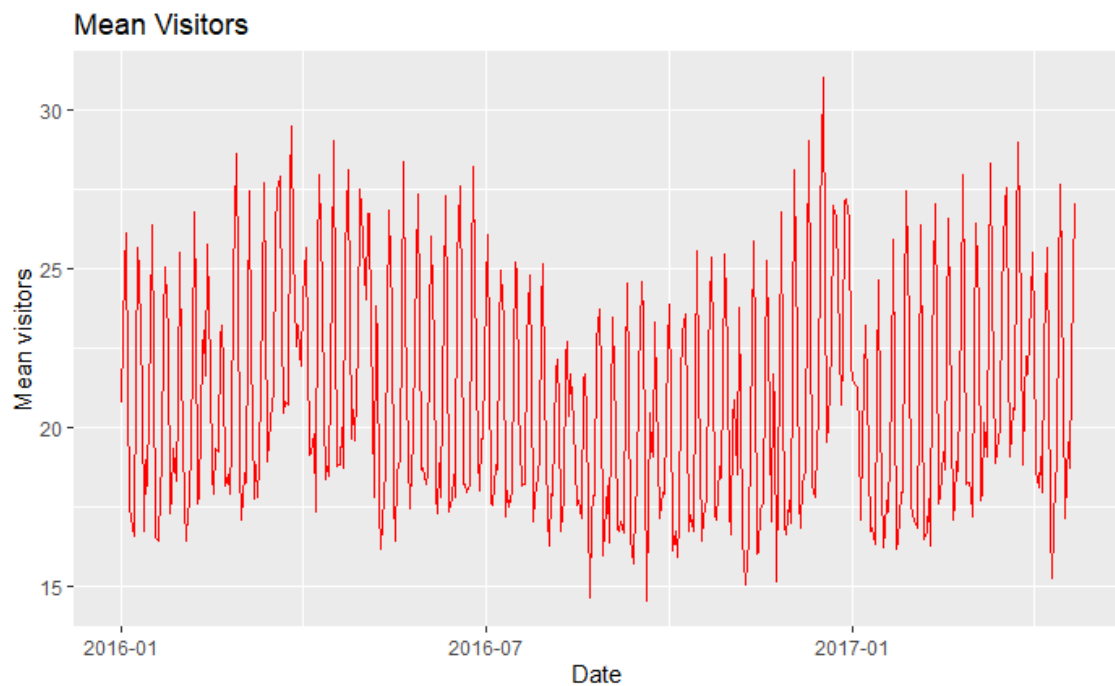
The missing values now is 75,531 from 325,999 rows that means 23.17% if missing values. Just by deleting the initials NA from the time series we have decrease the missing values from 40.99% to 23.17%. This is for the entire dataset, if we look only for the dates between 2016-01-01 to 2017-05-31, we can see that the missing values are 43,512 from 293,980 rows that means: 14.8% of missing values for the dates 2016-01-01 to 2017-04-22.

But we still have missing values. In order to impute the missing values, we are going to do it by using the adjacent values. But before doing this imputation, we are going to create a new column indicating which values were NA and which ones weren't.

Now there aren't any NA values, only on the id column (used only for the submission file), we can start to analyze the datasets and explore the data.

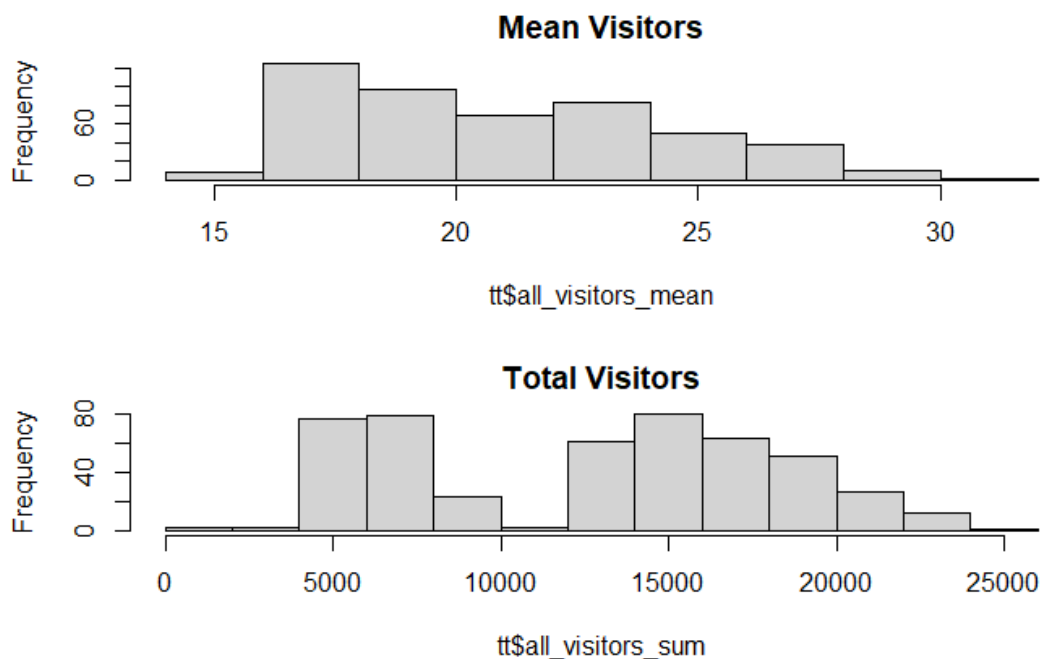
Plot of Total and Mean visitors





Here we can see the total number of visitors and the average visitors by day. We can see that if we look at the total number, we see a huge jump around 2016-7, because new restaurants were added to the air register. But if we look at the average visitors of the restaurant, we don't see that huge jump in visitors. This means that by adding new restaurants to the register we don't experience any substantial change in the visitors by restaurant. We can also see that there is some seasonality to the model.

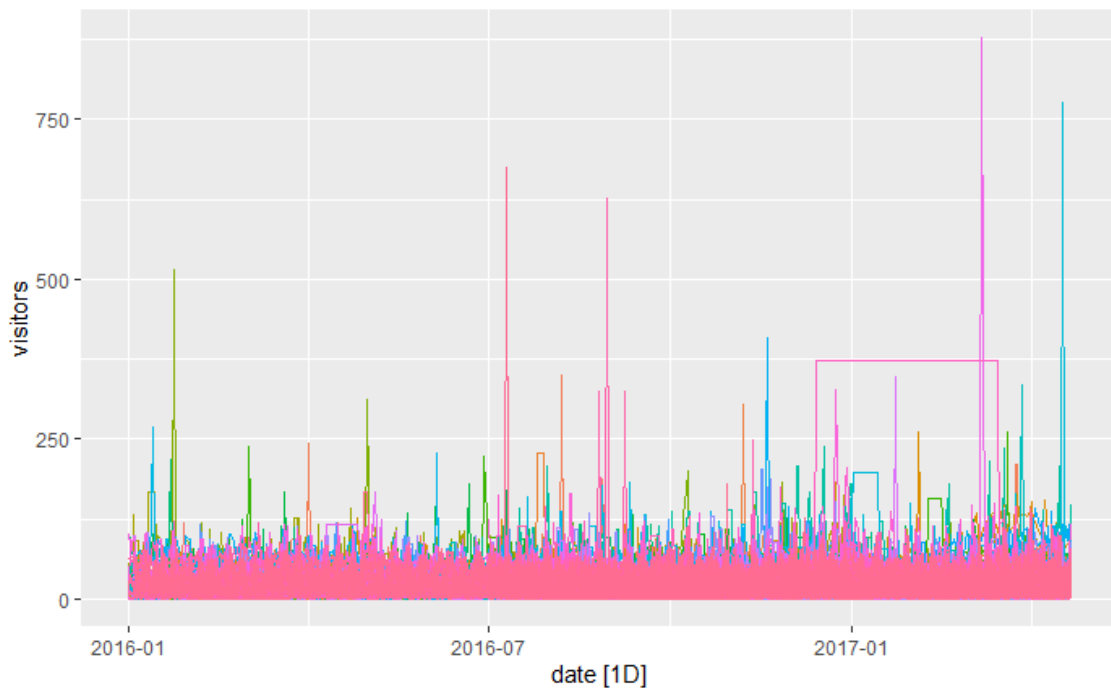
Histogram of Total and Mean visitor



By looking at the histogram of the total and mean visitors we can see a similar pattern. The mean visitors peak is between 15-20, and it has a long tail forwards 30 visitors.

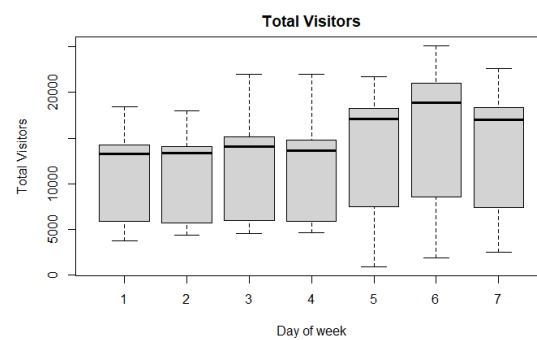
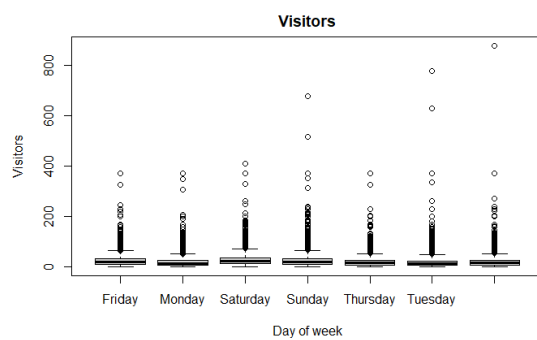
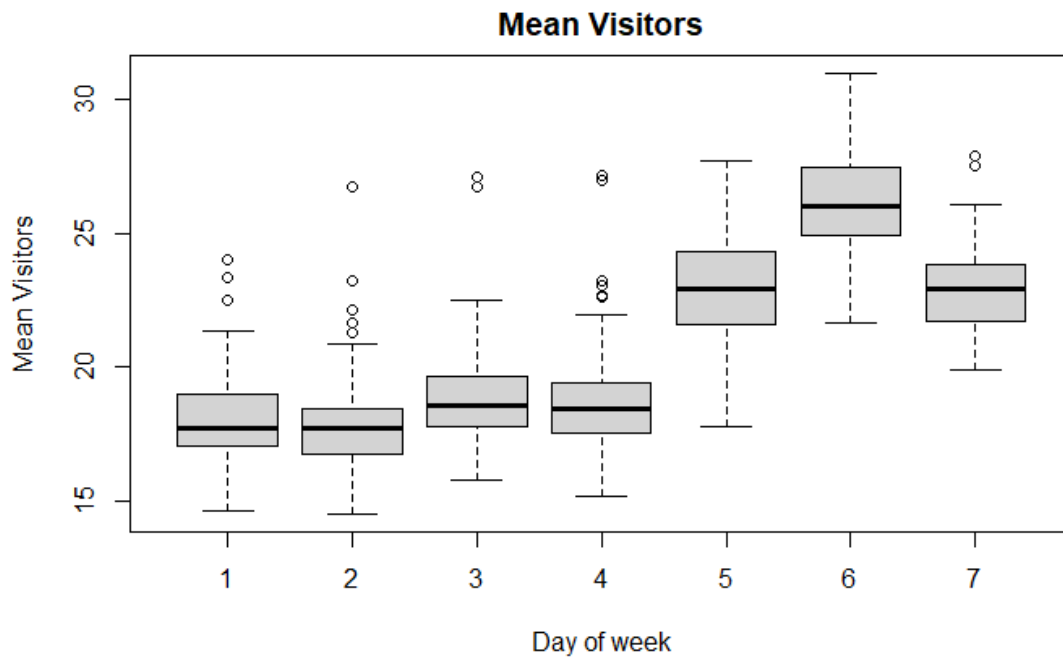
While for the total visitors we can see two distributions together, one around the 6,000 visitors and the other around the 15,000 visitors. This distribution is the same we saw before, where at the beginning visitors were around 6,000, and after 2016-7 when new restaurants were added to the pool, the total number of visitors increased.

Plot of visitors by restaurant

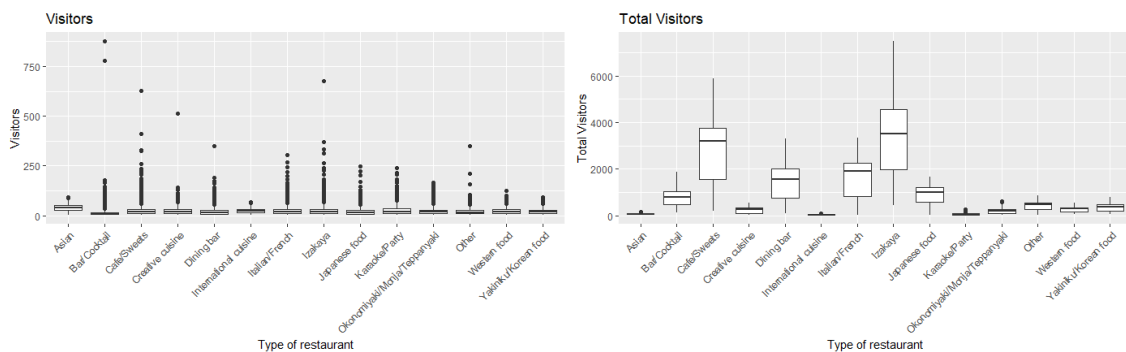
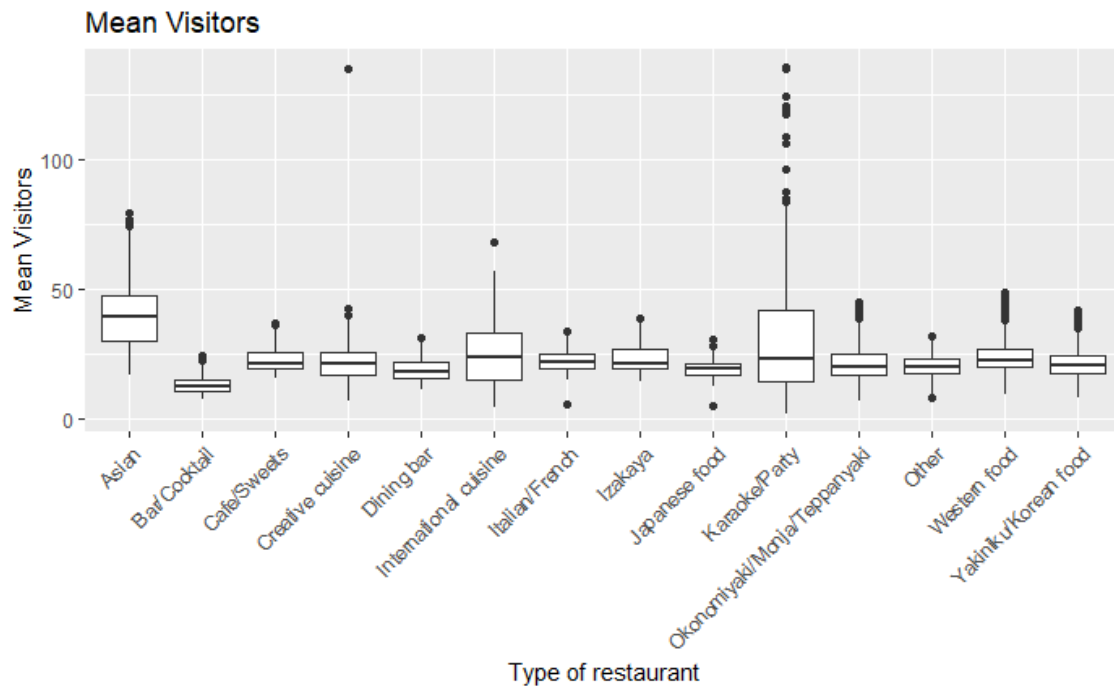


Here we can see the visitors by restaurant, we can see that most values are low, and there are some peaks that increase the number of visitors, this can be because of a big event or and error in the data.

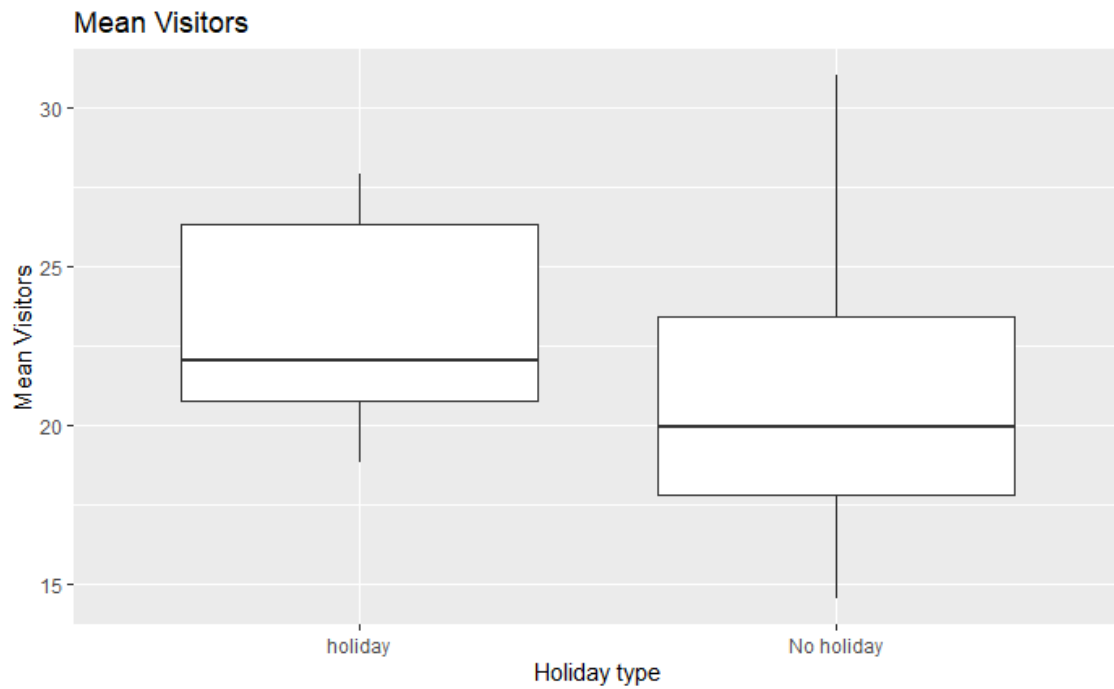
Boxplots of visitors by, Day of Week, Restaurant Type and Holiday flag



Here we can see the visitors by the day of the week, The upper image is the mean visitors by day of week, the bottom left is the visitors by day of week of all restaurants, and bottom right is the total visitors by day of week. We can see that the most visits occur on a Saturday follow by Friday and then by Sunday, meaning that restaurant received on average more visitors in the weekend that over the week.

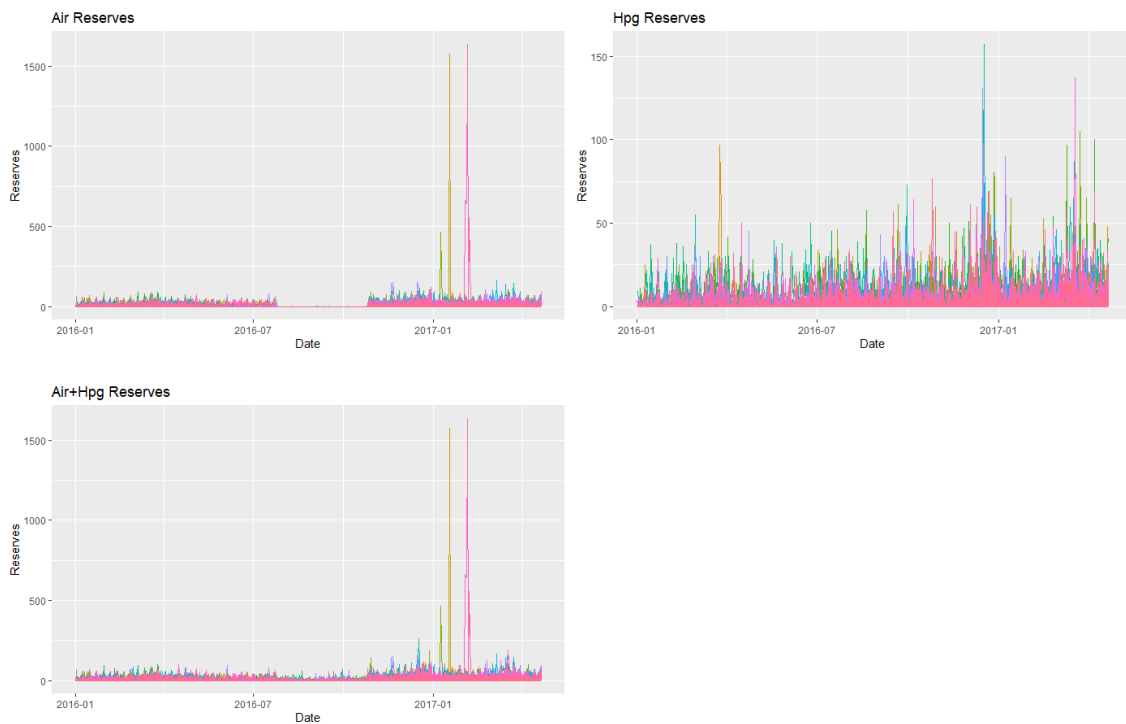


This graph is similar to the previous one, but in this case the visitors are separated by the genre of the restaurant. We can see that overall most types received similar number of visitors, there are a couple of exceptions, Asian seems to be the one that received on average more visitors than the rest of restaurants, International cuisine seems to be a little more spread than the rest achieving higher average than the others type, and lastly we have Karaoke/Party, in this case we can see that most of their values lies among the rest of the types of restaurants, but it also has many outliers that spread the average over a higher bracket, according to (HEADS OR TAILS, 2020) this huge difference in the karaoke is due to having a really high impact on weekends but an overall low impact on the rest of the week.

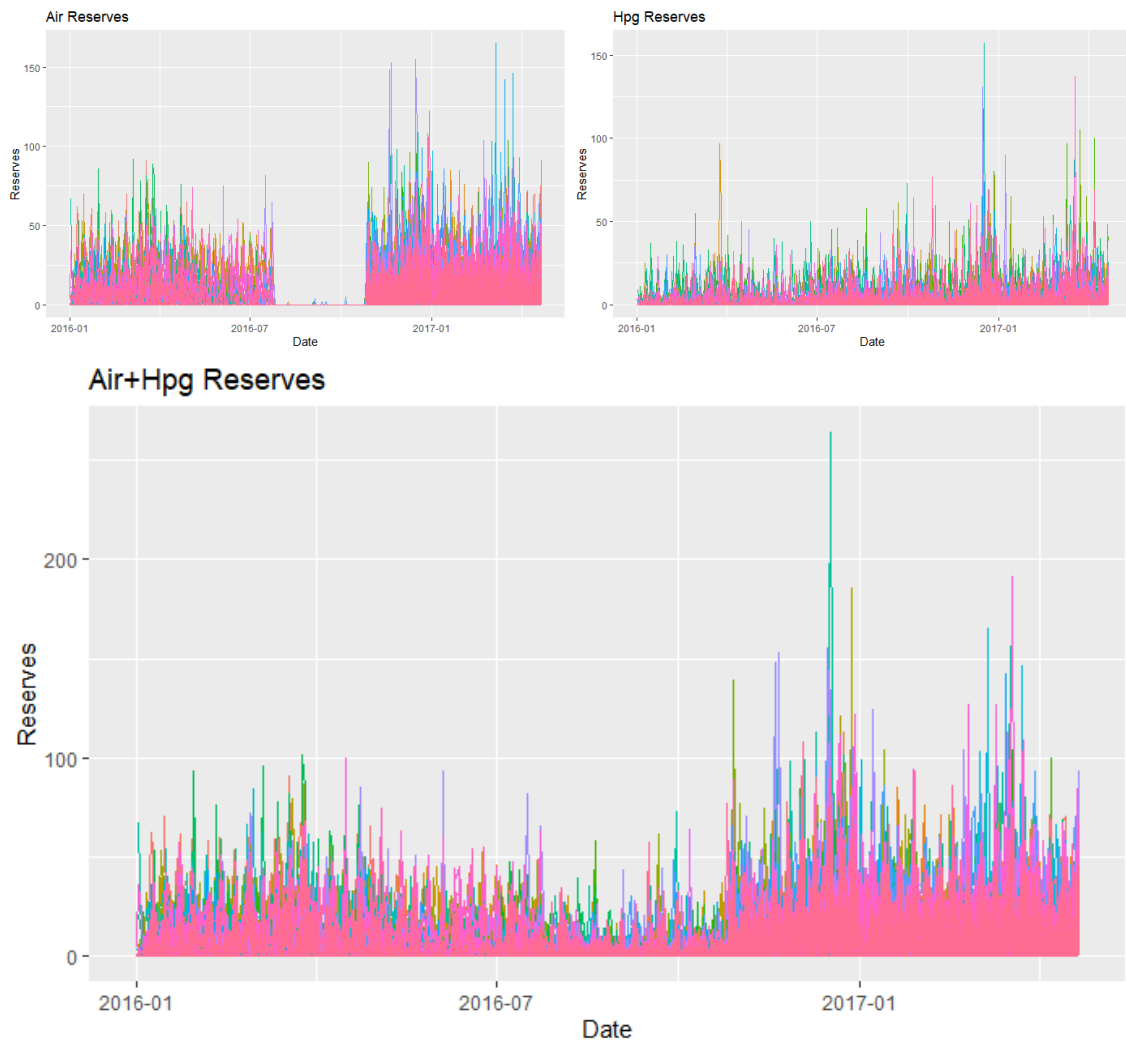


Here we can see the distribution of the Visitors by Holiday, we can see that overall, the average of holidays is way higher than on non-holidays. But we also must address that the No-Holidays has a higher distribution, meaning that the day with the highest mean visitors was on a non-holiday.

Plot of reserves, Air Hpg and Total

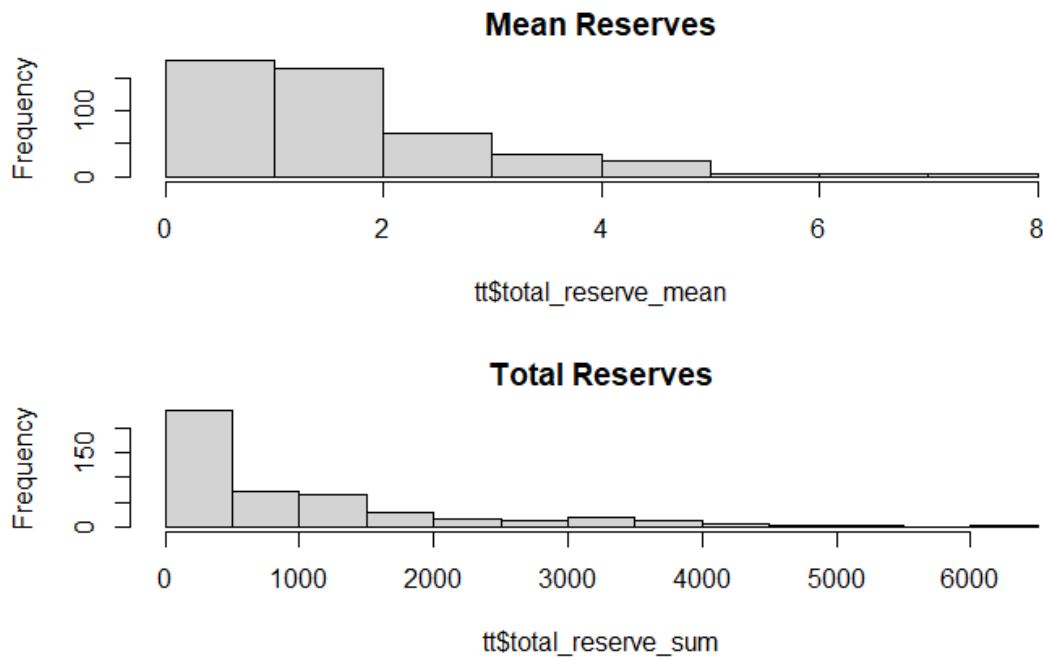


We can see that there are some outliers in the reservations, lets remove them and impute using the closest points. Once that is done, we get the next graphs



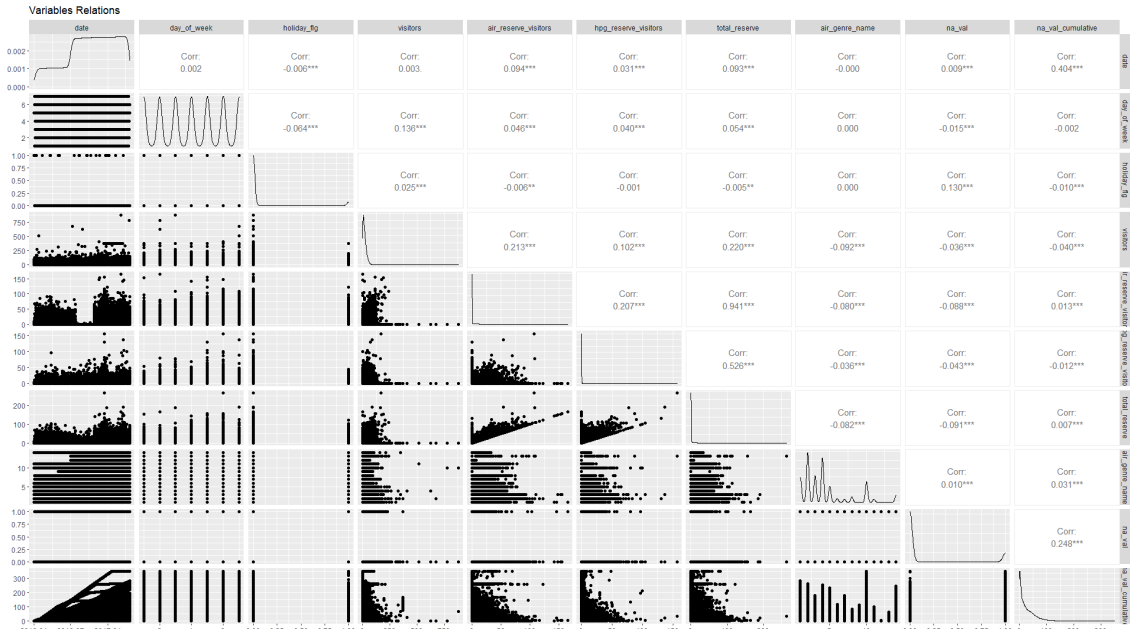
Here we can see that the air reserves experience a huge drop on a couple of months in between, but this can be compensated by the Hpg reservations, leaving a total reservation with no clear gap.

Histogram of Total and Mean Reserves



Here we can see the mean reserves (Air+Hpg) and Total reserves (Air+Hpg), we can see that most restaurants receive reserves for 0-2 people, and there is a long tail to 8 persons reservation.

Scatterplot Matrix of relations between variables



We can see that the visitor is pretty much uncorrelated with the rest of the variables, it has the highest correlation, 0.136 with the day of the week, meaning that there is some relationship between these two variables. The highest correlation between two variables can be find in by total_reserve and air_reserve_visitors, with a correlation of 0.941, the total_reserve also has a 0.526 correlation with the hpg_reserve_visitors, this is expected because the total reserve variable is built by combining this other two

variables into one. We can also see that the type of restaurant (`air_genre_name`) variable is uncorrelated with the rest of the variables, this is because in order to generate this graph this variable has been converted to integers, but we need to take into account that by doing this we are creating order in the types, where there weren't any other previously, for example I have made 'Italian/French' as a 1, 'Izakaya' as a 2, 'Cafe/Sweets' as a 4... But in reality, it doesn't make any sense that 'Cafe/Sweets' is 4 times bigger than 'Italian/French', or twice as big as 'Izakaya'. That is why this variable is uncorrelated. In order to use this variable to build the model we need to do one-hot encoding, the same goes for the day of the week.

Types of Models

There are a couple of different models and strategies that we can follow in order to build a good model.

The Linear Regression is the most basic model that we could use in order to predict the future visitors, but it will be hard to capture the true evolution of the target variable since, as we have seen before. We can increase the complexity of this model and use Multiple linear regression, and use other variables as inputs, such as the holiday flag, the day of the week, the genre of the restaurant, the reserves made, or the others variables that we created, the flag that marks if the value of visitors was missing for a particular date and restaurant, or the cumulative `na_flag`, that just adds the previous explained flag over time for each restaurant.

Another useful model will be the ARIMA model (Boomija, Anandaraj, Nandhini, & Lavanya, 2018) (HEADS OR TAILS, 2020), This model is composed of three different parts: the AR -auto regression-, the I -integration- and MA -moving average. This is most likely the best model, but for that we need to build it and test it.

Some other models that can be tested are dynamic regression using harmonic Fourier terms, or Neural Networks.

Models/Formulation

Now we need to split the data into train and test part, for that in going to use 48 days as test (~10%). This means that the training dataset spans from 2016-01-01 to 2017-03-05, and the test span from 2017-03-06 to 2017-04-22. This is going to be used to test the performance of the model.

Models' names:

- $t \rightarrow$ trend
- $s \rightarrow$ season
- $d \rightarrow$ day_of_week
- $h \rightarrow$ holiday_flg

- AR → air_reserve_visitors
- HR → hpg_reserve_visitors
- TR → total_reserve
- g → air_genre_name
- n → na_val
- NC → na_val_cumulative

I'm going to build the following models:

First, I built 7 different TSLM models (with different combination of parameters):

```
# Normal model Trend + season
ts = visitors ~ trend() + season()
# Include Trend season day of week and holiday flag
tsdh = visitors ~ trend() + season() + day_of_week + holiday_flg
# Include reservations
tsTR = visitors ~ trend() + season() + total_reserve
# Include the genre
tsg = visitors ~ trend() + season() + air_genre_name
# Include Null count
tsn = visitors ~ trend() + season() + na_val
tsNC = visitors ~ trend() + season() + na_val_cumulative
# Combine All predictors
tsdhARHTRnNC = visitors ~ trend() + season() + day_of_week + holiday_flg +
air_reserve_visitors + hpg_reserve_visitors + total_reserve + na_val +
na_val_cumulative
```

Second, I built 4 different ARIMA models:

```
arima = visitors
arima_TR = visitors ~ total_reserve
arima_dh = visitors ~ day_of_week + holiday_flg
arima_dhTR = visitors ~ day_of_week + holiday_flg + total_reserve
```

Thirdly, I built 3 different Dynamic Harmonic regression models:

```
'K = 1' = visitors ~ fourier(K=1)
'K = 2' = visitors ~ fourier(K=2)
'K = 3' = visitors ~ fourier(K=3)
```

Performance / Accuracy

Now let's evaluate the performance of all the models. For that let's look at the Accuracy of each model, according to a couple of metrics like RMSE, BIC, AICc...

The multiple linear regression model performance is:

	.model	avg_ME	avg_RMSE	avg_MAE	avg_ACF1	avg_MPE	avg_MAPE	avg_MASE	avg_RMSSE
1	tsTR	0.22021378	11.11120	8.557964	0.1295796	-44.88430	78.32824	NaN	NaN
2	tsdh	0.23467804	11.38330	8.823246	0.1324299	-47.86765	81.70607	NaN	NaN
3	ts	0.07110510	11.41842	8.862017	0.1371841	-49.49764	82.69096	NaN	NaN
4	tsn	-0.06164152	11.53534	8.961295	0.1338726	-51.22742	83.45211	NaN	NaN
5	tsdhARHRTnNC	-0.03917201	11.81831	9.180183	0.1264046	-45.48017	79.83732	NaN	NaN
6	tsNC	-0.46760078	12.38936	9.718778	0.1421485	-51.37179	85.74530	NaN	NaN
7	tsg	NaN	NaN	NaN	NA	NaN	NaN	NaN	NaN

	.model	avg_BIC	avg_AICc	avg_adj_r_squared	avg_CV	avg_AIC
1	ts	-Inf	-Inf	NaN	NaN	-Inf
2	tsdh	-Inf	-Inf	NaN	NaN	-Inf
3	tsdhARHRTnNC	-Inf	-Inf	NaN	NaN	-Inf
4	tsn	-Inf	-Inf	NaN	NaN	-Inf
5	tsNC	-Inf	-Inf	NaN	NaN	-Inf
6	tsTR	-Inf	-Inf	NaN	NaN	-Inf

The ARIMA model performance:

	.model	avg_ME	avg_RMSE	avg_MAE	avg_ACF1	avg_MPE	avg_MAPE	avg_MASE	avg_RMSSE
1	arima_dh	0.9163948	10.45674	7.980348	0.1289290	-50.59266	78.62813	NaN	NaN
2	arima	0.6167927	11.33515	8.811350	0.1751549	-62.80125	89.85773	NaN	NaN
3	arima_dhTR	NaN	NaN	NaN	NA	NaN	NaN	NaN	NaN
4	arima_TR	NaN	NaN	NaN	NA	NaN	NaN	NaN	NaN

	.model	avg_BIC	avg_AICc	avg_AIC	avg_sigma2
1	arima_dhTR	2233.175	2190.837	2189.638	108.8259
2	arima_dh	2235.450	2194.508	2193.493	117.4522
3	arima	2248.403	2229.628	2229.369	123.5923
4	arima_TR	2305.807	2283.686	2283.435	131.7538

The Dynamic Harmonic regression models performance:

	.model	avg_ME	avg_RMSE	avg_MAE	avg_ACF1	avg_MPE	avg_MAPE	avg_MASE	avg_RMSSE
1	K = 1	0.724428	10.75301	8.273992	0.1080718	-55.26454	83.1153	NaN	NaN
2	K = 2	NaN	NaN	NaN	NA	NaN	NaN	NaN	NaN
3	K = 3	NaN	NaN	NaN	NA	NaN	NaN	NaN	NaN

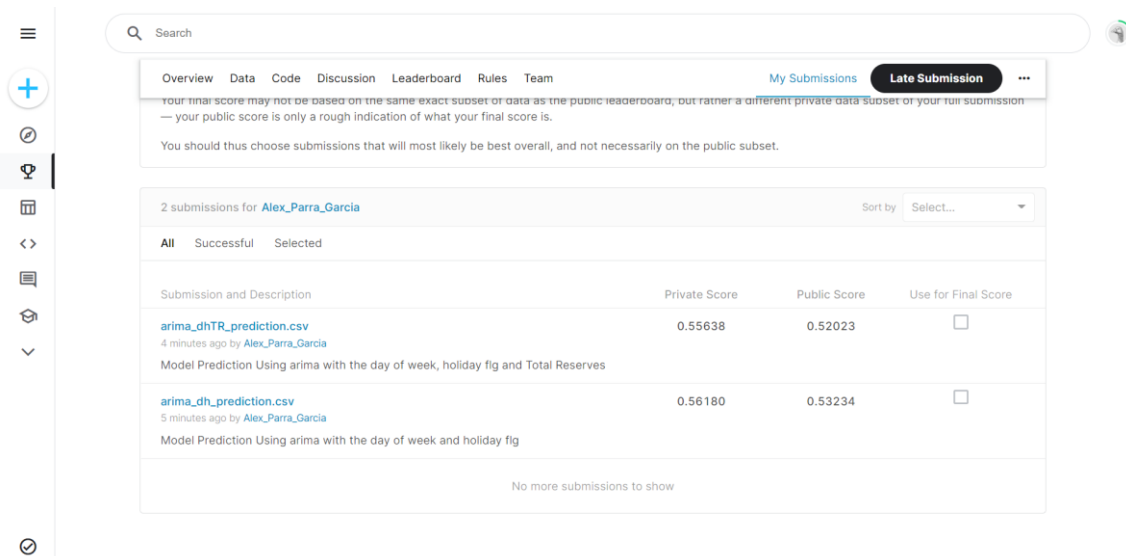
	.model	avg_BIC	avg_AICc	avg_AIC	avg_sigma2
1	K = 2	2239.701	2208.471	2207.824	110.6143
2	K = 1	2240.050	2215.036	2214.715	115.6214
3	K = 3	2241.920	2204.336	2203.396	108.4865

We can see that the model with the lowest RMSE is the Arima model with the day of the week and Holiday flag as predictors. If we look at the BIC or AICc the best model is the ARIMA with day of the week, Holiday flag and total reserves as predictors, follow by the ARIMA with the day of the week and Holiday flag.

There were some NaN problems in the predictions of some models, that I don't really understand by they are happening. The dataset doesn't contain any Na values, so no idea on that regard, because when I run the model with just a couple of restaurants it works fine, but when I run all of them it gives those errors, and since in order to train the models on the hole restaurant dataset take so long to run, I'm unable to reiterate to find the error. As I said when I run it with fewer restaurant in order to increase the seep there are no errors.

So now that we have the best model, being the ARIMA with the day of the week and Holiday flag, and also the ARIMA day of the week, Holiday flag and Total Reserves. And submit those to the Kaggle in order to find the one with the lowest error.













Once the submission is done, we can check the scores that each model achieved:



2 submissions for Alex_Parra_Garcia			
Submission and Description		Private Score	Public Score
arima_dhTR_prediction.csv	4 minutes ago by Alex_Parra_Garcia Model Prediction Using arima with the day of week, holiday fig and Total Reserves	0.55638	0.52023
arima_dh_prediction.csv	5 minutes ago by Alex_Parra_Garcia Model Prediction Using arima with the day of week and holiday fig	0.56180	0.53234

We can see that the model that included the day of week, holiday flag and total reserves, outperformed the one using only the day of week and holiday flag.

We can see that the Score for the best model is 0.55638. This will mean that this model falls between the positions 1503 and 1504.

1498	▼ 20	cqychen		0.55598	1	5Y
1499	▼ 20	meicher		0.55598	16	5Y
1500	▼ 63	Daebang Kim		0.55598	5	5Y
1501	▼ 43	Kaige Yang		0.55602	9	5Y
1502	▲ 4	Vadi		0.55605	6	5Y
1503	▼ 259	Dmitry P		0.55629	18	5Y
1504	▲ 110	kymhorskell		0.55662	5	5Y
1505	▼ 3	lotusky		0.55668	11	5Y
1506	▼ 40	Konstantin Izmaylov (DMIA)		0.55684	39	5Y
1507	▲ 26	jocelyn		0.55711	9	5Y
1508	▲ 247	Cobham Coders		0.55718	14	5Y
1509	▼ 17	Michael Levinson		0.55728	1	5Y

Limitations

There is a really big limitation regarding the input files that I had access to, since there were a lot of missing values, in some columns more than 95%, this in return means that we need to drop the variable entirely. The visitor variable had more than 40% missing values, meaning that a lot of imputations was made in order to eliminate the null values. All this imputation, that was around 15% of the data, can have a negative impact on the model performance.

Other limitations are in regard to the number of combinations of variable used, due to the limitations of my computer I couldn't try all the combinations of parameters that I had in mind, meaning that there could be an even better model with a different set of combinations.

Future Work

Increase the reliability of the input file, by increasing the amount of information contained in it, decreasing the number of Null values, this will decrease the number of imputations that need to be performed. There were more than 40% of missing value for the variable visitors, and this variable is the one that we want to forecast.

Trying multiple more combinations of parameters in order to find a better model. We could also have created new predictors variables, such as taking into account the population density that probably affects the number of restaurants visitors. Using the Lat and Lon in order to calculate the number of restaurants in a given radius around each one, since this will have an effect on it. Also, we could use other factors such as google trends with the different genre of restaurants in order to track the popularity of each one as times goes on.

What was learned/Conclusions

I have learned how to manage, merge, explore, big datasets on R. How to build robust models capable of making predictions into the future in order to get a glimpse into future results, this will enable us to plan accordingly.

I have also learned that the ARIMA models tend to perform better than some more basic models such as the normal linear regression or multiple linear regression. But combining into the ARIMA model the use of other variables, can enhance the performance.

References

- Boomija, G., Anandaraj, A., Nandhini, S., & Lavanya, S. (may de 2018). Restaurant Visitor Time Series Forecasting Using Autoregressive Integrated Moving Average. *Journal of Computational and Theoretical Nanoscience*, 5. doi:10.1166/jctn.2018.7345
- HEADS OR TAILS. (2020). *Kaggle*. Retrieved from Be my guest - Recruit Restaurant EDA: <https://www.kaggle.com/code/headsortails/be-my-guest-recruit-restaurant-eda/report>
- TROY WALTERS. (2017). *Kaggle*. Obtenido de A Very Extensive Recruit Exploratory Analysis: <https://www.kaggle.com/code/captcalculator/a-very-extensive-recruit-exploratory-analysis/report>