Forecasting Sales in a Fast-Food Chain

Group 1: Luiz Carvalho, Manoj Soman Nair, Nafiseh Davari, Stanislav Taov

16/02/2020

Abstract

Based on many research papers we know that the weather has a strong influence on food sales as weather impacts the emotional state of people and can affect their purchase decisions. As a result, many retail food chains started using weather data to forecast short-term sales predictions to minimize stocked or expired products and avoid missing sales. In this study, we built a system to predict food sales for a fast-food in São Paulo, Brazil. We took into consideration, the temporal granularity of sales data, the input variables to use for predicting sales and the representation of the sales output variable. We were able to decide which machine learning algorithms suites better food sales prediction and used appropriate measures for evaluating their accuracy and showed success in predicting sales of some weather-sensitive products such as beverages.

Business problem

In today's highly competitive and constantly changing business environment, the accurate and timely estimation of future sales, also known as sales prediction or sales forecasting, can offer critical knowledge to companies involved in the manufacturing, wholesale or retail of products. Short-term predictions mainly help in production planning and stock management, while long-term predictions can help in business development decision making. In our specific case, we have a fast-food chain in Brazil with 400 stores with difficulty in predicting its short term production.

Sales prediction is particularly important for this particular company due to the short shelf-life of many of its products, which leads to loss of income in both shortage and surplus situations. Producing too many leads to waste of products, while producing too few leads to opportunity loss. Therefore we have a situation where predicting correctly how much you have to produce each item each day is important.

Moreover, food consumer demand is constantly fluctuating due to factors such as price, promotions, changing consumer preferences or weather changes. Sales prediction is typically done arbitrarily by managers. However, skilled managers are hard to find and they are not always available. In our specific case, this forecast is done based on experience however it remains far from accurate. Average loss (too many or too few) spins around 10%. Here it is important to put the management perspective: It is their view today they rely too much on the managers and would like to have a computer system that can play the role of a skilled manager. Over time the expectation was to have some tool that would free the company from human dependence. In addition, they believe that the level of error is high and could be reduced.

Therefore, from the management perspective, a system capable of predicting the sales would be worth having even if at the beginning it doesn´t perform better than the current process. Equal performance would be acceptable. In addition to that, there was an understanding that the system would be able to improve its performance over time as more and more historical data is added to its reference database.

Analytical problem

The problem is how to build a model that effectively predicts the demand with a level of assertiveness equal or superior to the current one and improves over time. One way is to build such a system would be to model the expert knowledge of skilled managers within a computer system. Alternatively, we could exploit the wealth of sales data and related information to automatically construct accurate sales prediction models via machine learning techniques. The latter is a much simpler process, it is not biased from the particularities of a specific sales manager and it is dynamic, meaning it can adapt to changes in the data. Furthermore, it has the potential to outweigh the prediction accuracy of a human expert, who typically is imperfect.

Nevertheless, we listened to the thoughts of the people currently in charge of making this forecast and we were informed that they believe that the demand is correlated to the following factors:

1) Day of the month (payment days usually have bigger demand)
2) Day of the week (Fridays, Saturdays, Sundays and holidays usually have a big demand)
3) Month (Holidays months usually have bigger sales – In Brazil Dec-Jan-Jun-Jul)
4) Weather (temperature, rain and sun have an impact on what people eat)

This particular company sells its food through several channels: 1) directly from its stores, 2) through a web deliver service and 3) through a call-center. In our study, we are not going to differentiate these channels just counting the total volume sold of each item each day. Here it is worth mentioning that the insigns provided by the people in charge of the process today should be seen with a grain of salt given the fact that they know that a system like that would be built to replace them. Therefore all these assumptions must be checked against the hard data.

Datasets

To be able to build our model we decided to use as a sales sample in the city of São Paulo which alone responds for almost 40% of the total sales. This is a simplifying strategy given the fact that if the process works for this city we can easily deploy it in the others.

Getting the Data

Initially we have managed to get sales data by type of item for thirteen months (Jan – 2018 and Jan-2019) – 396 registers. Secondly we managed to get the weather stations measurements in São Paulo for the whole year of 2018 and January of 2019. It is public information available at the website:

http://www.inmet.gov.br/projetos/rede/pesquisa/ In sequence we prepared this data crossing these two files unifying them by date.

Data Evaluating

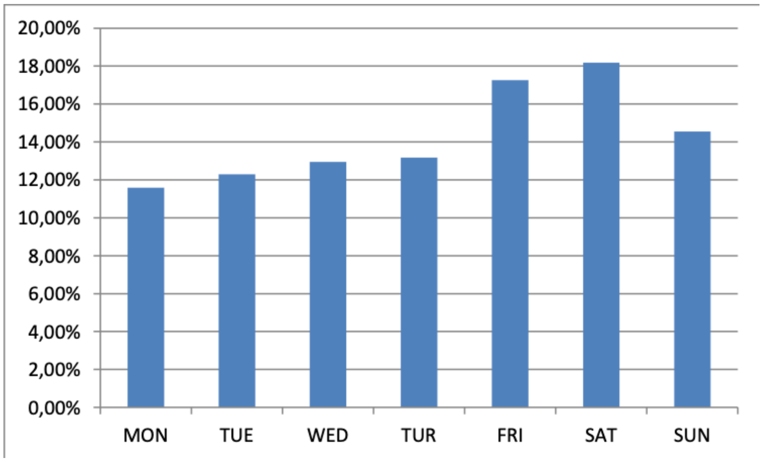
Accuracy and completeness: A visual inspection showed that the sales data per day was basically correct, although some values seem to be too high or too low. The weather measurements had some problems of completeness. There were several days without the insolation, temperature and humidity recorded. In addition of that there are several days where the level of rain is zero, this is a problem because we don´t know it happens because it wasn´t recorded or because in fact didn´t rain in these days.

Data Dictionary

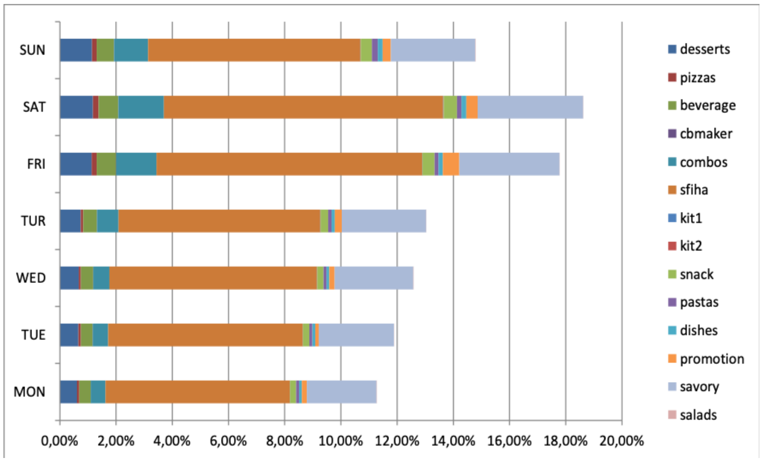
Data – Day of the weather measurements and sales Precipitac – Volume of rain in millimetres per square meter during the day Tempmax – Max temperature during the day Tempmin – Minimum temperature during the day Tempmed – Average temperature during the day Umidade – Level of humidity in the air Insolacao – Level of sun Diasemana – day of the week abreviation Diasem – weight of the day of the week considering the average volumes sold Mes – Weight of the month considering the average sales volumes Desserts – number of desserts sold in the day Pizzas – Number of pizzas sold in the day Beverage – Number of Beverage sold in the day Cbmaker – Number of a special dish sold in the day Combos – Number of combos sold in the day Sfiha – Number of sfihas sold in the day kit1 – Number of kits 01 sold in the day (Is a dish with a gift) kit2 - Number of kits 02 sold in the day (Is a dish with a gift) snack – Number of snacks sold in the day pastas – Number of pastas sold in the day dishes – Number of lunches sold in the day promotion – Number of promotions sold in the day (This is episodic and may not be counted) savory – Number of savory sold in the day salads – Number of salads sold in the day total – Total number of items sold in the day

Data Exploration and Cleaning

The meaning of the numbers in the field diasem is: 1) Monday, Tuesday, Thursday 2) Friday 3) Saturday and Sunday 4) Holidays The number represents the weight of the day regards sales. This weight reflects the view of the current planners regards sales and we need to check if the assumption holds. Analyzing the sales per day of the week we have the following graphic:



The current assumption of separating the days of week into four categories seems to be a bit wrong , the graphic shows three ranges as follows: 1) Monday – Tuesday – Wednesday-Thursday – Range 11% - 13% of the sales 2) Friday and Saturday – Range 17% and 19% of the sales 3) Sunday – Range 14% a 15% of the sales The holidays match the volumes of FRI and SAT falling into the Range 2. The distribution by type of item goes as shown:



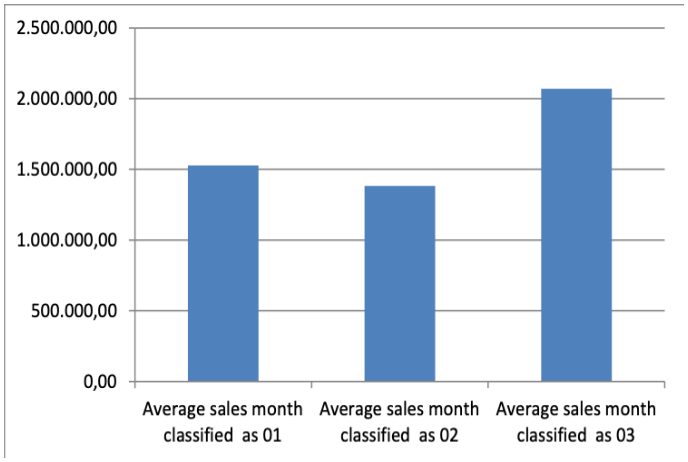
Here we can see that the division of training data by type of the day would be better if done conjugate with the type of product, That means the variation in sales along the week is not uniform for all products. We didn´t do that and surely it is an aspect of this model which can be improved. Our analysis suggested that we group the days by type as follows:

1) Monday – Tuesday – Wednesday-Thursday – Range 11% - 13% of the sales
2) Sunday – Range 14% a 15% of the sales
3) Friday and Saturday – Range 17% and 19% of the sales

The same process of grouping the days by sales profile was done for the months, the current method was as follows:

1) March, April, May, August, September, October, November
2) February and July
3) January ,June and December

The number represents the weight of the month regards sales, this weight reflects the view of the current planners regards sales and we checked if the assumption holds. Analyzing the actual sales monthly by month classification we saw the following:



That suggests that there is a differentiation among the months as follows: 1) 1.500.000 items sold monthly (8,61% above the baseline) 2) 1.380.000 items sold monthly (Baseline) 3) 2.079.000 items sold (50% above the baseline) That suggests the change 1 with 2 as a classification for the month to keep it aligned with the sales volumes: 1) 1.380.000 items sold monthly (Baseline) 2) 1.500.000 items sold monthly (8,61% above the baseline) 3) 2.079.000 items sold (50% above the baseline)

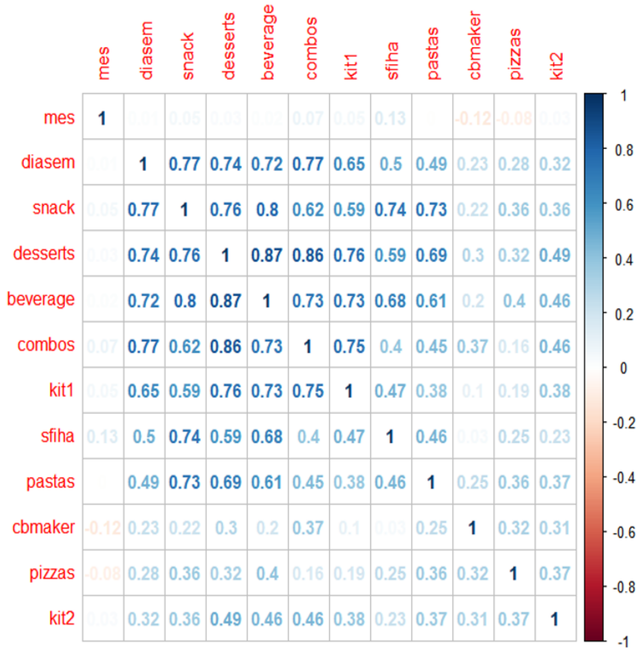
As we can see, although the weather do have some influence regards the amount sold of each type of product the main aspects influencing the demand are the days of week and month of the year. Note that the objective is first to identify if the listed factors are in fact defining the demand and identify which ones are the most relevant. Establishing a correlation using the Kendall method between the demand for specific items. Here we are going to limit ourselves to the following items: Desserts – number of desserts sold in the day Pizzas – Number of pizzas sold in the day Beverage – Number of Beverage sold in the day Sfiha – Number of sfihas sold in the day snack – Number of snacks sold in the day pastas – Number of pastas sold in the day dishes – Number of lunches sold in the day savory – Number of savory sold in the day salads – Number of salads sold in the day

The reason for that is the fact that the others are not regular items but some sort of promotion only made available for limited span of time. Items expurgated from the analysis: Cbmaker – Number of a special dish sold in the day Combos – Number of combos sold in the day kit1 – Number of kits 01 sold in the day (includes a gift) kit2 - Number of kits 02 sold in the day (Includes a gift) promotion – Number of promotions sold in the day (This is episodic and may not be counted)

Seasonal parameters:

Diasem – weight of the day of the week considering the average volumes sold (1,2 or 3)
Mes – Weight of the month considering the average sales volumes (1,2 or 3)

We managed to see the correlation between the day of the week and month and the consumption of the several items:



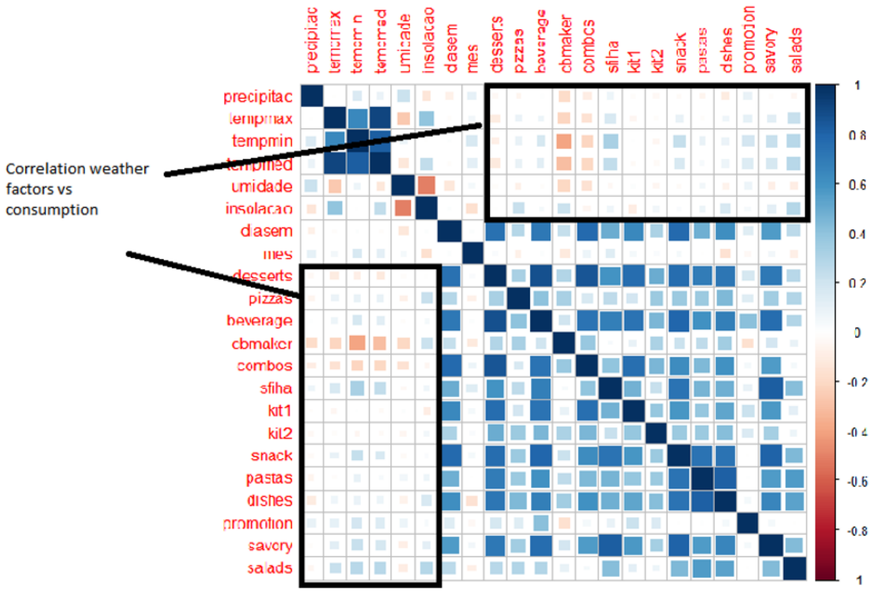
	mes	diasem	snack	desserts	beverage	combos	kit1	sfiha	pastas	cbmaker	pizzas	kit2
mes	1	0.01	0.05	0.03	0.02	0.07	0.05	0.13	0	-0.12	-0.08	0.03
diasem	0.01	1	0.77	0.74	0.72	0.77	0.65	0.5	0.49	0.23	0.28	0.32
snack	0.05	0.77	1	0.76	0.8	0.62	0.59	0.74	0.73	0.22	0.36	0.36
desserts	0.03	0.74	0.76	1	0.87	0.86	0.76	0.59	0.69	0.3	0.32	0.49
beverage	0.02	0.72	0.8	0.87	1	0.73	0.73	0.68	0.61	0.2	0.4	0.46
combos	0.07	0.77	0.62	0.86	0.73	1	0.75	0.4	0.45	0.37	0.16	0.46
kit1	0.05	0.65	0.59	0.76	0.73	0.75	1	0.47	0.38	0.1	0.19	0.38
sfiha	0.13	0.5	0.74	0.59	0.68	0.4	0.47	1	0.46	0.03	0.25	0.23
pastas	0	0.49	0.73	0.69	0.61	0.45	0.38	0.46	1	0.25	0.36	0.37
cbmaker	-0.12	0.23	0.22	0.3	0.2	0.37	0.1	0.03	0.25	1	0.32	0.31
pizzas	-0.08	0.28	0.36	0.32	0.4	0.16	0.19	0.25	0.36	0.32	1	0.37
kit2	0.03	0.32	0.36	0.49	0.46	0.46	0.38	0.23	0.37	0.31	0.37	1

As we can see the consumption of some items varies more them others as the days of the week change.

Now we have to see if there is a correlation between the weather factors and the consumption by item:

Precipitac – Volume of rain in millimetres per square meter during the day Tempmax – Max temperature during the day Tempmin – Minimum temperature during the day Tempmed – Average temperature during the day Umidade – Level of humidity in the air Insolacao – Level of sun

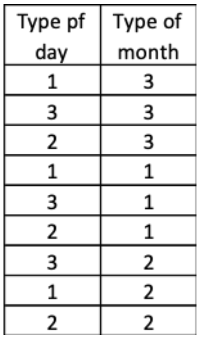
Remembering that doing the correlation we are going to identify the variable R which can vary from +1 to -1 indicating that a relationship exists between the two variables from absolute direct correlation (+1), no correlation (0) to inverse correlation (-1). Using R studio and the command “corrplot” we managed to see the following:



As we can see in the graphics there is some correlation between weather parameters and consumption. However, the weather conditions are not so determining as the day of the week and the month of the year.

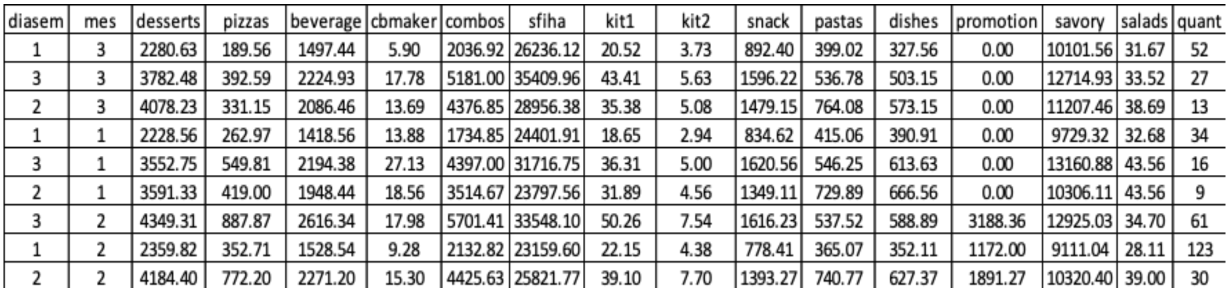
Therefore building a model to guess the demand will imply the definition of the average demand of the day of the week in a given month and in sequence adjust this demand by the weather conditions.

To archive that we grouped the demand by the combination of the type of day vs type of month with nine categories:



Type pf day	Type of month
1	3
3	3
2	3
1	1
3	1
2	1
3	2
1	2
2	2

This analysis grouped the data as follows:



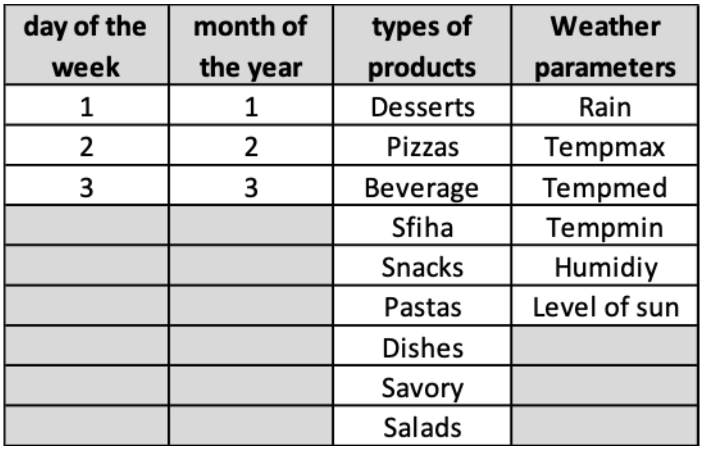
diasem	mes	desserts	pizzas	beverage	cbmaker	combos	sfiha	kit1	kit2	snack	pastas	dishes	promotion	savory	salads	quant
1	3	2280.63	189.56	1497.44	5.90	2036.92	26236.12	20.52	3.73	892.40	399.02	327.56	0.00	10101.56	31.67	52
3	3	3782.48	392.59	2224.93	17.78	5181.00	35409.96	43.41	5.63	1596.22	536.78	503.15	0.00	12714.93	33.52	27
2	3	4078.23	331.15	2086.46	13.69	4376.85	28956.38	35.38	5.08	1479.15	764.08	573.15	0.00	11207.46	38.69	13
1	1	2228.56	262.97	1418.56	13.88	1734.85	24401.91	18.65	2.94	834.62	415.06	390.91	0.00	9729.32	32.68	34
3	1	3552.75	549.81	2194.38	27.13	4397.00	31716.75	36.31	5.00	1620.56	546.25	613.63	0.00	13160.88	43.56	16
2	1	3591.33	419.00	1948.44	18.56	3514.67	23797.56	31.89	4.56	1349.11	729.89	666.56	0.00	10306.11	43.56	9
3	2	4349.31	887.87	2616.34	17.98	5701.41	33548.10	50.26	7.54	1616.23	537.52	588.89	3188.36	12925.03	34.70	61
1	2	2359.82	352.71	1528.54	9.28	2132.82	23159.60	22.15	4.38	778.41	365.07	352.11	1172.00	9111.04	28.11	123
2	2	4184.40	772.20	2271.20	15.30	4425.63	25821.77	39.10	7.70	1393.27	740.77	627.37	1891.27	10320.40	39.00	30

This would be the demand if only these two factors were influencing the actual sales, therefore the challenge is to identify how the weather conditions make the average demand deviate (up and down) from these values baseline factors.

Identifying the demand

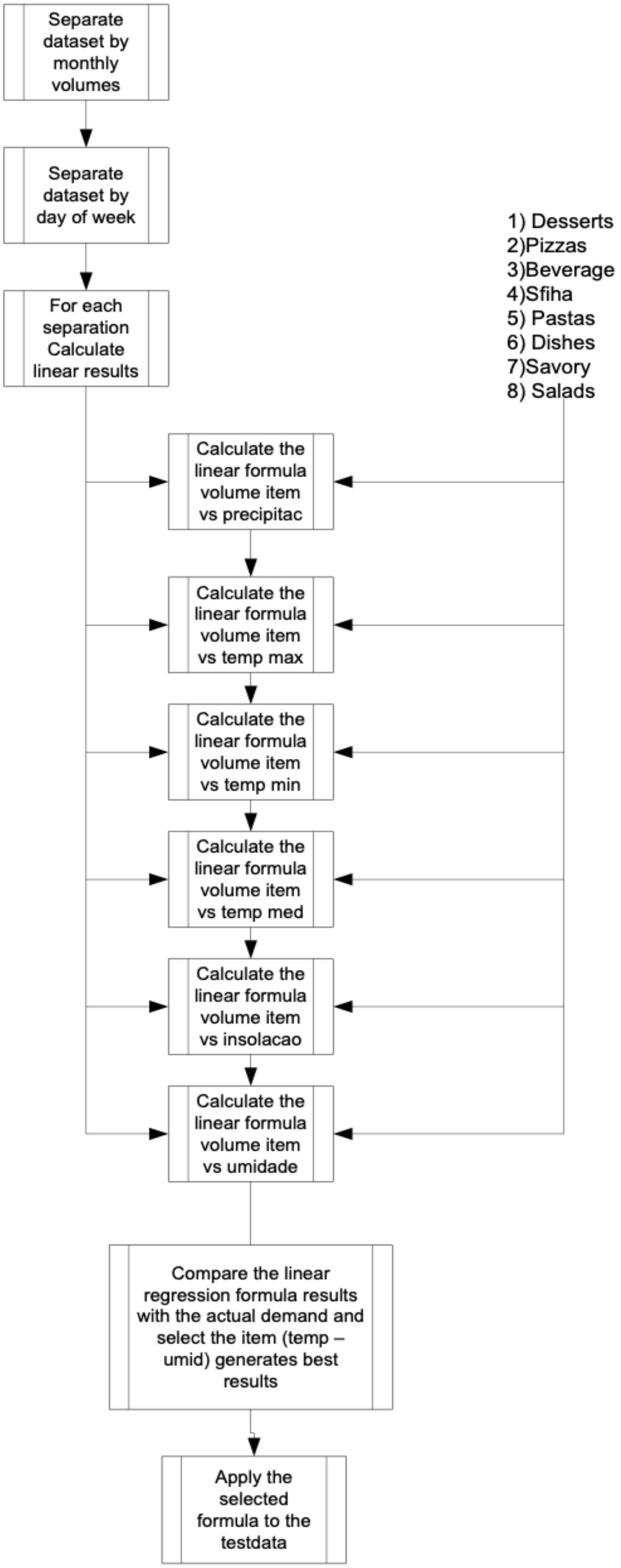
Once we sliced the dataset by type of week and type of month we have to identify which one of the six weather parameters is more influential in adjusting the demand for each item (nine types).

To be able to do that we have to identify the correlation factor R^2 between each weather parameter and the volume sold. The factor with bigger R^2 is the ones to be used as "predictor" of the demand. We are going to use linear regression to predict the demand, identifying the parameters A and B in the formula $Ax + B = Y$ where x is the chosen weather parameter and Y de expected demand for the product. This calculation will be done for each one of the nine types of products. For each row in the data_test we do the following combinations of tests:



day of the week	month of the year	types of products	Weather parameters
1	1	Desserts	Rain
2	2	Pizzas	Tempmax
3	3	Beverage	Tempmed
		Sfiha	Tempmin
		Snacks	Humidiy
		Pastas	Level of sun
		Dishes	
		Savory	
		Salads	

Analytical model


Separate dataset by monthly volumes
Separate dataset by day of week
For each separation Calculate linear results
1) Desserts
2)Pizzas
3)Beverage
4)Sfiha
5) Pastas
6) Dishes
7)Savory
8) Salads
Calculate the linear formula volume item vs precipitac
Calculate the linear formula volume item vs temp max
Calculate the linear formula volume item vs temp min
Calculate the linear formula volume item vs temp med
Calculate the linear formula volume item vs insolacao
Calculate the linear formula volume item vs umidade
Compare the linear regression formula results with the actual demand and select the item (temp – umid) generates best results
Apply the selected formula to the testdata

```
glimpse(df)

## Observations: 396
## Variables: 61
## $ data       <fct> 09/01/2018, 13/01/2018, 14/01/2018, 12/01/2018,
10/01/2018…
## $ praca      <fct> SAO PAULO, SAO PAULO, SAO PAULO, SAO PAULO, SAO PAULO,
SAO…
## $ classe     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA…
## $ web        <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ ifood      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ call       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ android    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ iphone     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ total      <dbl> 45909, 68321, 56313, 57792, 49405, 50214, 44148, 46017,
50…
## $ precipitac <dbl> 9.70, 3.75, 3.75, 14.80, 0.80, 18.50, 12.10, 3.75,
2.20, 3…
## $ tempmax    <dbl> 25.5, 26.8, 30.4, 24.4, 29.0, 30.2, 28.5, 28.1, 25.9,
29.4…
## $ tempmin    <dbl> 18.3, 19.0, 19.2, 19.3, 19.0, 20.0, 19.7, 21.6, 19.8,
17.5…
## $ tempmed    <dbl> 21.16, 23.08, 23.20, 22.22, 22.84, 23.68, 22.80, 23.94,
21…
## $ umidade    <dbl> 83.00, 64.75, 80.00, 75.00, 78.75, 72.50, 52.25, 78.50,
87…
## $ insolacao  <dbl> 3.97, 2.40, 5.10, 0.50, 5.30, 3.70, 11.20, 5.00, 1.30,
10.…
## $ diasemana  <fct> TUE, SAT, SUN, FRI, WED, TUR, TUR, TUE, WED, FRI, MON,
SAT…
## $ diasem     <int> 1, 3, 3, 2, 1, 1, 1, 1, 1, 2, 4, 3, 3, 1, 1, 2, 1, 1,
1, 3…
## $ mes        <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3…
## $ desserts   <dbl> 1938, 3062, 3760, 3311, 1972, 2065, 1616, 1851, 2325,
2725…
## $ pizzas     <dbl> 157, 407, 360, 398, 163, 200, 178, 218, 200, 370, 541,
447…
## $ beverage   <dbl> 1153, 2083, 1988, 1924, 1270, 1446, 1265, 1357, 1367,
1710…
## $ cbmaker    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ combos     <dbl> 479, 2080, 1334, 1722, 515, 521, 506, 550, 516, 1793,
733,…
```

```
## $ sfiha      <dbl> 31772, 43783, 33867, 35623, 34514, 34451, 29874, 29862,
33…
## $ kit1       <dbl> 12, 33, 19, 32, 13, 8, 17, 16, 15, 28, 10, 14, 18, 24,
9, …
## $ kit2       <dbl> 3, 3, 2, 4, 0, 6, 3, 2, 0, 2, 0, 2, 4, 4, 1, 2, 2, 2,
0, 1…
## $ snack      <dbl> 948, 1891, 1793, 1762, 959, 1103, 929, 1124, 1041,
1637, 1…
## $ pastas     <dbl> 356, 589, 856, 529, 373, 403, 463, 575, 556, 407, 494,
540…
## $ dishes     <dbl> 165, 229, 308, 201, 133, 154, 198, 282, 235, 203, 189,
251…
## $ promotion  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ savory     <dbl> 8893, 14120, 11964, 12237, 9442, 9821, 9053, 10137,
10845,…
## $ salads     <dbl> 33, 41, 62, 49, 51, 36, 46, 43, 45, 34, 22, 44, 56, 40,
45…
## $ range      <fct> Range-09, Range-20, Range-18, Range-18, Range-10,
Range-12…
## $ dessert1   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ desserte   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ pizzas1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ pizzase    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ beverage1  <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ beveragee  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ cbmaker1   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ cbmakere   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ combos1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ combose    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ sfiha1     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ sfihae     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ kit11      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ kit1e      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ kit21      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
```

```
## $ kit2e      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ snack1     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ snacke     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ pastas1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ pastase    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ dishes1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ dishese    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ promotion1 <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ promotione <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ savory1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ savorye    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ salads1    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
## $ saladse    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0…
```

There are many columns with zero values. Dropping all columns with zero values

```
df$cbmaker   <- NULL
df$combos    <- NULL
df$kit2      <- NULL
df $kit1      <- NULL
df $promotion <- NULL
df $dessert1 <- NULL
df $desserte <- NULL
df $pizzas1 <- NULL
df $pizzase <- NULL
df $beverage1 <- NULL
df $beveragee <- NULL
df $cbmaker1 <- NULL
df $cbmakere <- NULL
df $combos1 <- NULL
df $sfiha1 <- NULL
df $sfihae <- NULL
df $kit21 <- NULL
df $kit11 <- NULL
df $kit1e <- NULL
df $combose <- NULL
df $kit2e <- NULL
df $snack1 <- NULL
df $snacke <- NULL
df $pastas1 <- NULL
df $pastase <- NULL
df $dishes1 <- NULL
df $dishese <- NULL
df $promotion1 <- NULL
df $promotione <- NULL
df $savory1 <- NULL
df $savorye <- NULL
df $salads1 <- NULL
df $saladse <- NULL
df $web <- NULL
df $ifood <- NULL
df $call <- NULL
df $android <- NULL
df $iphone <- NULL
df $classe <- NULL
```

```
head(df)
```

```
##         data     praca total precipitac tempmax tempmin tempmed umidade
## 1 09/01/2018 SAO PAULO 45909       9.70    25.5    18.3   21.16   83.00
## 2 13/01/2018 SAO PAULO 68321       3.75    26.8    19.0   23.08   64.75
## 3 14/01/2018 SAO PAULO 56313       3.75    30.4    19.2   23.20   80.00
## 4 12/01/2018 SAO PAULO 57792      14.80    24.4    19.3   22.22   75.00
## 5 10/01/2018 SAO PAULO 49405       0.80    29.0    19.0   22.84   78.75
## 6 11/01/2018 SAO PAULO 50214      18.50    30.2    20.0   23.68   72.50
##   insolacao diasemana diasem mes desserts pizzas beverage sfiha snack
pastas
## 1      3.97       TUE      1   3     1938    157     1153 31772   948
356
## 2      2.40       SAT      3   3     3062    407     2083 43783  1891
589
## 3      5.10       SUN      3   3     3760    360     1988 33867  1793
856
## 4      0.50       FRI      2   3     3311    398     1924 35623  1762
529
## 5      5.30       WED      1   3     1972    163     1270 34514   959
373
## 6      3.70       TUR      1   3     2065    200     1446 34451  1103
403
##   dishes savory salads    range
## 1    165   8893     33 Range-09
## 2    229  14120     41 Range-20
## 3    308  11964     62 Range-18
## 4    201  12237     49 Range-18
## 5    133   9442     51 Range-10
## 6    154   9821     36 Range-12
```

To deal with the inconsistencies we defined three strategies:

1) Regards the missing values in precipitation we managed to see the average rain in each month (public information) and check it against the sum of each day/month in the database. Through this process, we managed to identify that in fact, the zero represented days without rain (the data was right).

2) The registers with temperature min, max or medium and humidity equal zero were filled with the mean of these parameters (just two samples fall into this scenario).

3) In the case of the sun intensity, we had a situation where 196 out of 396 samples were equal zero. Considering that there is no possibility that the sun didn´t appear for so many days it was assumed we had a problem with the data. We managed to check the average sun intensity per month in the city of São Paulo (public information) and fill the gaps manually. Subsequently, we identify that the mean of the registers with measurement represents the average therefore it was possible to implement a code to correct it automatically.

4) The values of sales that diverge too much from what is typical were treaded by a bell distribution were the values whose frequency were smaller than 1% were eliminated from the sample. Note that we did not eliminate the whole row (Each row had sales for each one of the eight types of products) we just didn´t count the line when predicting the specific item.

The evaluation was important because allowed us to create a cleaning layer in the R code where we check these factors (2 and 3) and adjust it automatically – It is important because we assume that a new samples will be added to the training data as time goes by and this new data probably will suffer from the same problems.

Here we check if the columns tempmin, tempmax, tempmed, humidit and insulation are 0. If they are we replace the value for the mean

```
  tempmax    <-data[[11]]
  tempmin    <-data[[12]]
  tempmed    <-data[[13]]
  umidade    <-data[[14]]
  insolacao  <-data[[15]]

  media1 <- mean(tempmax,   trim = 0, na.rm = TRUE)
  media2 <- mean(tempmin,   trim = 0, na.rm = TRUE)
  media3 <- mean(tempmed,   trim = 0, na.rm = TRUE)
  media4 <- mean(umidade,   trim = 0, na.rm = TRUE)
  media5 <- mean(insolacao, trim = 0, na.rm = TRUE)
```

```
sum(is.na(df))
```

```
## [1] 0
```

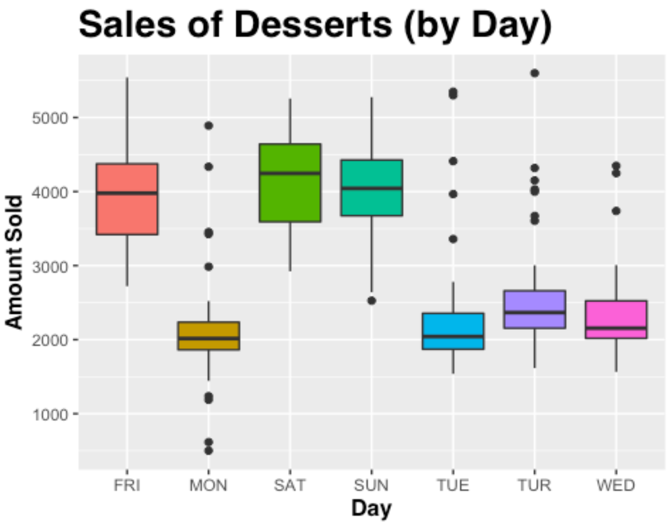
There are no missing values

```
summary(df)
```

```
##          data          praca         total          precipitac    
##  ,,,,,     :  1   SAO PAULO:396   Min.   : 10445   Min.   : 0.100  
##  01/01/2018:  1                   1st Qu.: 37251   1st Qu.: 2.890  
##  01/01/2019:  1                   Median : 48768   Median : 3.750  
##  01/02/2018:  1                   Mean   : 49665   Mean   : 5.629  
##  01/03/2018:  1                   3rd Qu.: 60379   3rd Qu.: 3.750  
##  01/04/2018:  1                   Max.   :100711   Max.   :76.600  
##  (Other)   :390                                                    
##     tempmax         tempmin         tempmed         umidade     
##  Min.   :14.50   Min.   : 9.10   Min.   :11.78   Min.   :42.00  
##  1st Qu.:24.10   1st Qu.:14.90   1st Qu.:18.88   1st Qu.:67.75  
##  Median :27.20   Median :17.75   Median :21.43   Median :74.75  
##  Mean   :26.65   Mean   :17.20   Mean   :21.04   Mean   :73.84  
##  3rd Qu.:29.70   3rd Qu.:19.50   3rd Qu.:23.44   3rd Qu.:81.75  
##  Max.   :35.10   Max.   :23.40   Max.   :28.06   Max.   :93.25  
##                                                                 
##    insolacao      diasemana     diasem           mes          desserts   
##  Min.   : 0.100   FRI:56    Min.   :1.000   Min.   :1.00   Min.   : 501  
##  1st Qu.: 3.970   MON:57    1st Qu.:1.000   1st Qu.:1.00   1st Qu.:2088  
##  Median : 4.350   SAT:56    Median :1.000   Median :1.00   Median :2692  
##  Mean   : 5.232   SUN:56    Mean   :1.823   Mean   :1.77   Mean   :3045  
##  3rd Qu.: 7.300   TUE:57    3rd Qu.:3.000   3rd Qu.:3.00   3rd Qu.:4030  
##  Max.   :11.700   TUR:57    Max.   :4.000   Max.   :3.00   Max.   :5599  
##                   WED:57                                                 
##      pizzas          beverage        sfiha           snack     
##  Min.   :  46.0   Min.   : 369   Min.   : 5164   Min.   : 216  
##  1st Qu.: 161.0   1st Qu.:1367   1st Qu.:21590   1st Qu.: 724  
##  Median : 265.5   Median :1782   Median :26914   Median :1010  
##  Mean   : 438.3   Mean   :1862   Mean   :27312   Mean   :1119  
##  3rd Qu.: 387.2   3rd Qu.:2272   3rd Qu.:32641   3rd Qu.:1479  
##  Max.   :4295.0   Max.   :3974   Max.   :59153   Max.   :2396  
##                                                                
##      pastas           dishes          savory          salads     
##  Min.   : 168.0   Min.   :133.0   Min.   : 2874   Min.   : 4.00  
##  1st Qu.: 348.0   1st Qu.:318.2   1st Qu.: 8637   1st Qu.:25.00  
##  Median : 428.0   Median :408.0   Median :10062   Median :32.00  
##  Mean   : 471.7   Mean   :451.3   Mean   :10570   Mean   :32.77  
##  3rd Qu.: 564.0   3rd Qu.:576.0   3rd Qu.:12262   3rd Qu.:40.25  
##  Max.   :1064.0   Max.   :883.0   Max.   :19352   Max.   :68.00  
##                                                                  
##       range    
##  Range-11: 48  
##  Range-12: 35  
##  Range-10: 32  
##  Range-20: 30  
##  Range-13: 25  
##  Range-14: 23
##  (Other) :203
```

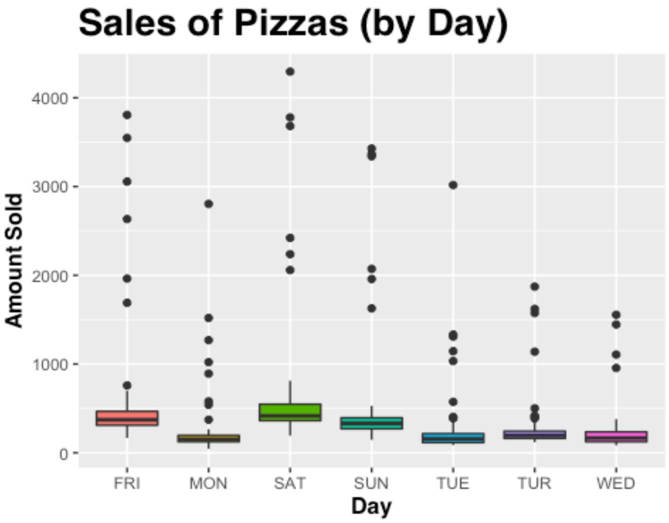
It seems some columns have outliers. By plotting whisker plots for each product we can observe outliers more closely.

Plotting boxplots for desserts



There are some spikes in sales during certain days (Friday, Saturday, Sunday) and other days have lower sales but with many outliers.

Plotting boxplots for pizzas

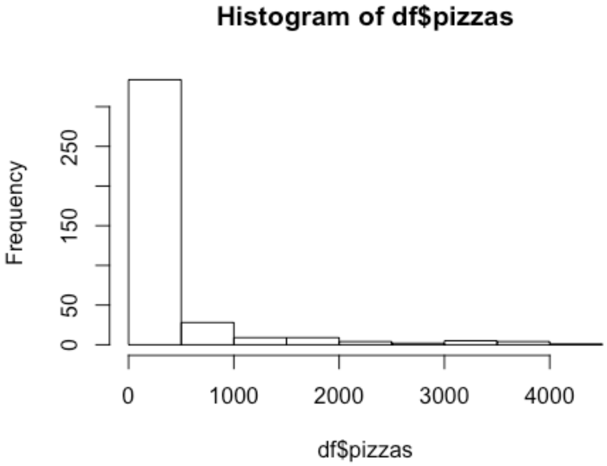


Pizza sales seem very spread with a lot of outliers, we will explore it more closely.

```
## [1] 3.74455
```

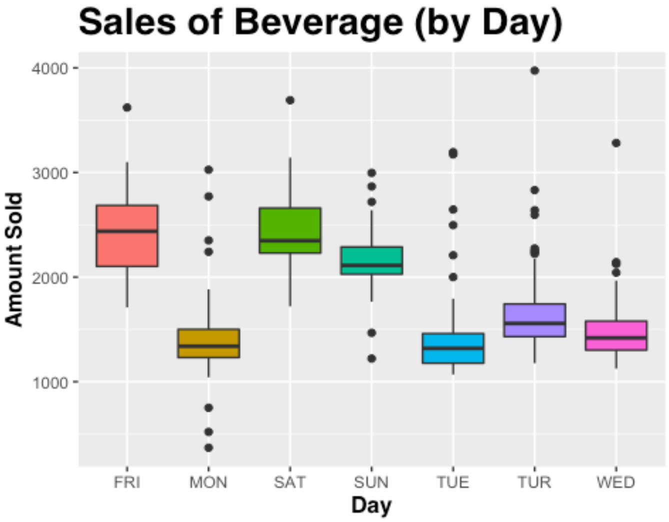
Pizza column has negative skewness of 3.74455

```
hist(df$pizzas)
```



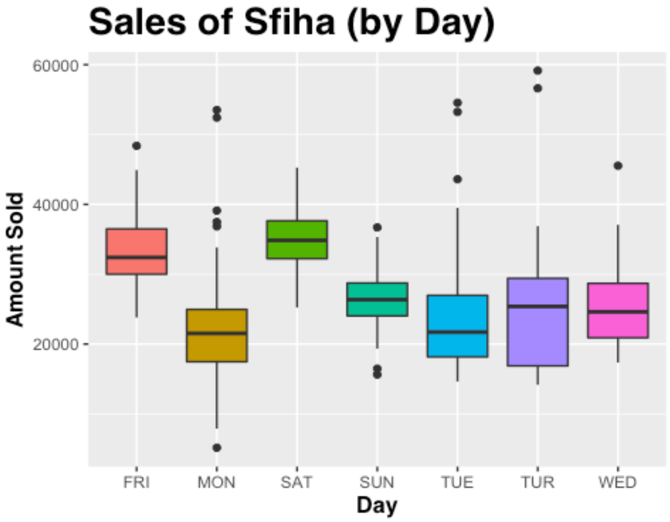
As we can see the average daily number of pizza sales is somewhere around 300 pizzas and there are a lot of outliers with over 5000 sales per day.

Plotting boxplots for beverage



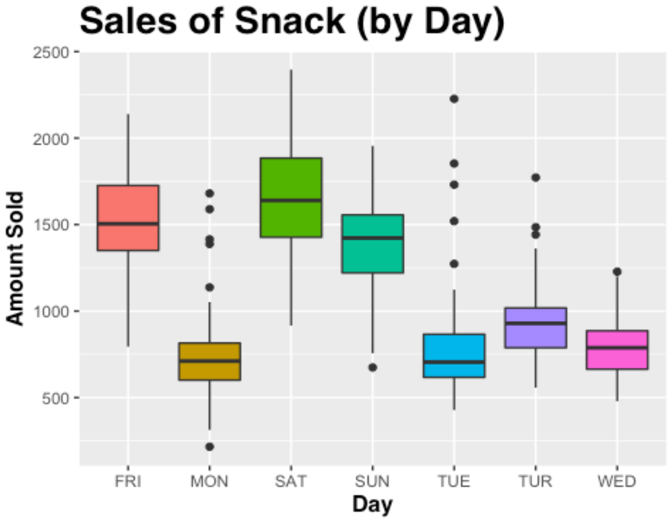
Beverage sales are quite spread. Again more sales happened on Friday, Saturday, Sunday.

Plotting boxplots for sfiha



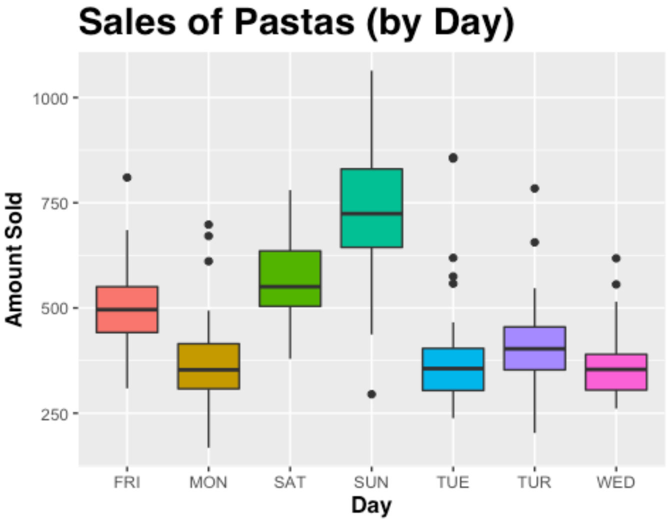
More sfiha sales happened on Friday and Saturday, some days have outliers.

Plotting boxplots for snack



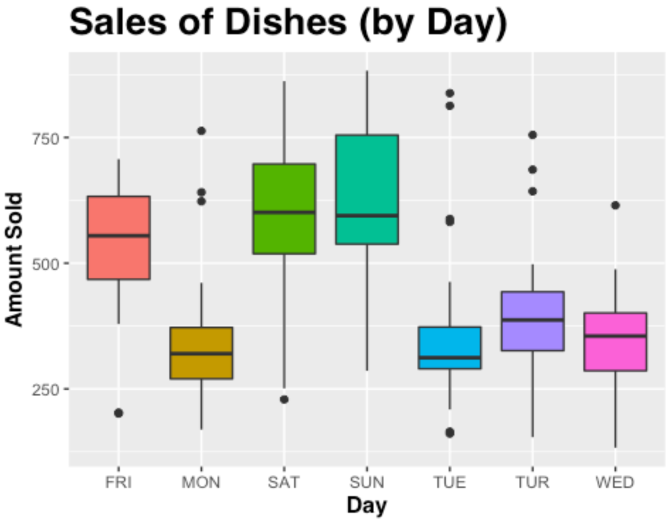
More snack sales happened on Saturday, Sunday and Friday and Tuesday, Monday and Thursday have most outliers.

Plotting boxplots for pastas



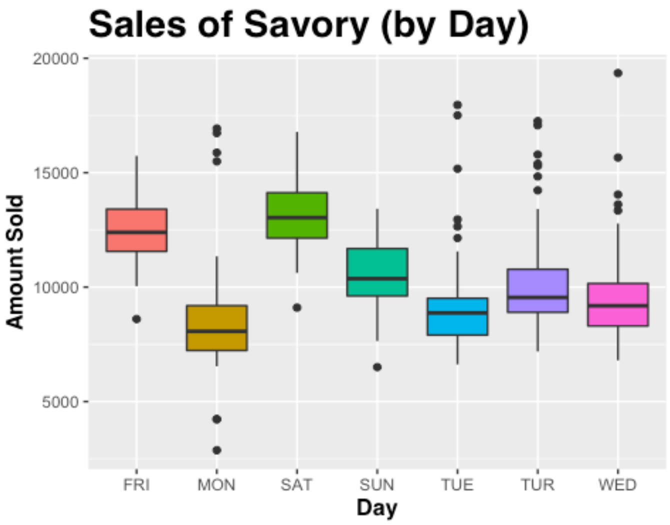
More sales of pasta happened on Sunday, very interesting.

Plotting boxplots for dishes

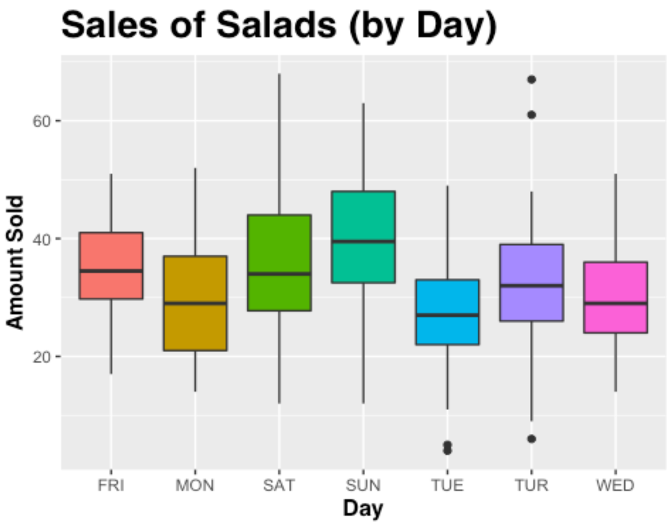


Very similar to sales of snacks.

Plotting boxplots for savory



Plotting boxplots for salads



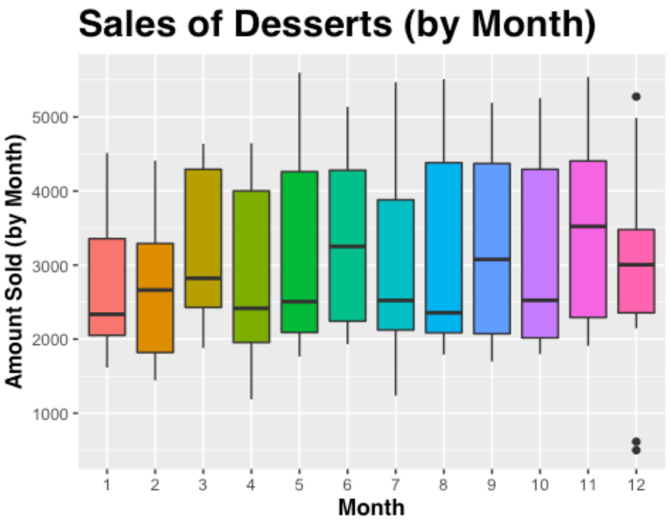
Sales of salads relatively low and consistent and have a few outliers.

Let’s make boxplots using the month of the year

```
## [1] 1

##          data     praca total precipitac tempmax tempmin tempmed umidade
## 1 09/01/2018 SAO PAULO 45909       9.70    25.5    18.3   21.16   83.00
## 2 13/01/2018 SAO PAULO 68321       3.75    26.8    19.0   23.08   64.75
## 3 14/01/2018 SAO PAULO 56313       3.75    30.4    19.2   23.20   80.00
## 4 12/01/2018 SAO PAULO 57792      14.80    24.4    19.3   22.22   75.00
## 5 10/01/2018 SAO PAULO 49405       0.80    29.0    19.0   22.84   78.75
## 6 11/01/2018 SAO PAULO 50214      18.50    30.2    20.0   23.68   72.50
##   insolacao diasemana diasem mes desserts pizzas beverage sfiha snack
pastas
```

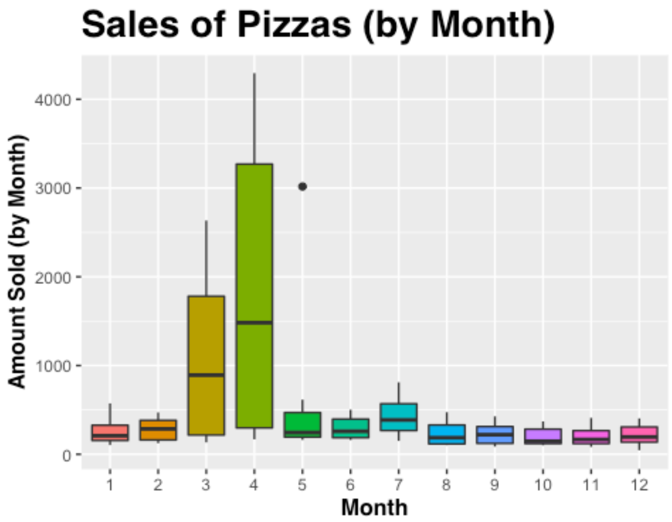
```
## 1      3.97       TUE      1   3     1938    157     1153 31772   948
356
## 2      2.40       SAT      3   3     3062    407     2083 43783  1891
589
## 3      5.10       SUN      3   3     3760    360     1988 33867  1793
856
## 4      0.50       FRI      2   3     3311    398     1924 35623  1762
529
## 5      5.30       WED      1   3     1972    163     1270 34514   959
373
## 6      3.70       TUR      1   3     2065    200     1446 34451  1103
403
##   dishes savory salads    range month
## 1    165   8893     33 Range-09     1
## 2    229  14120     41 Range-20     1
## 3    308  11964     62 Range-18     1
## 4    201  12237     49 Range-18     1
## 5    133   9442     51 Range-10     1
## 6    154   9821     36 Range-12     1
```

Plotting boxplots for desserts/month



December shows some outliers, lowest months are December, January and February.

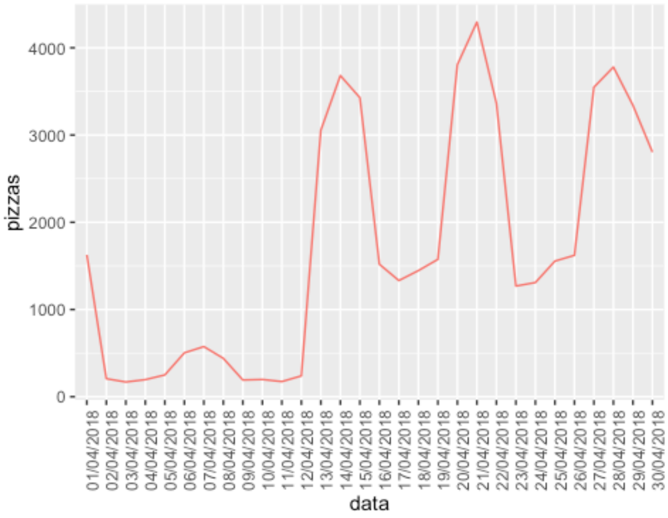
Plotting boxplots for pizzas/month



Pizza sales had a huge spike in sales in April, but why? Let’s explore…

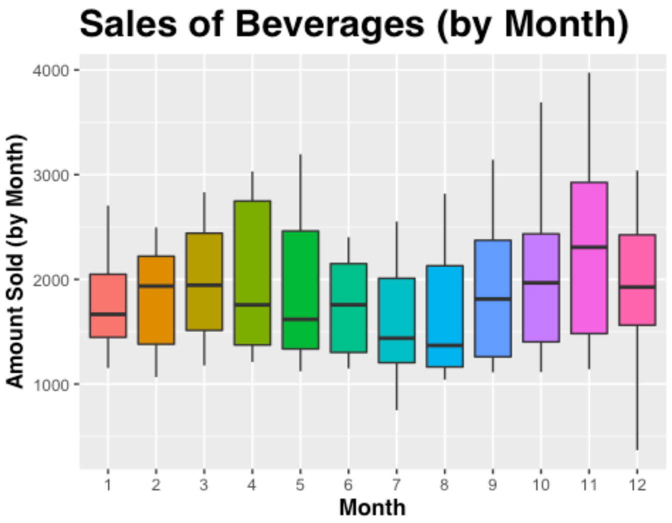
```
april <- df %>%
  filter(month == 4)
head(april$pizzas, n = 30)
```

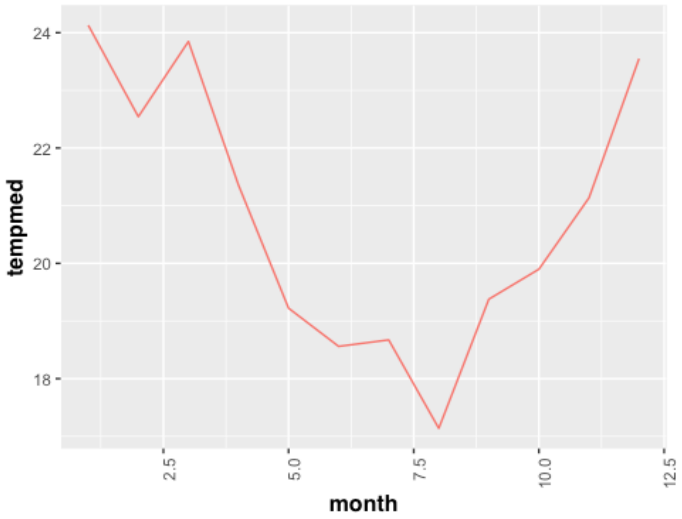
```
##  [1] 1627  169  207  196  250  575  440  503  239 3056  198  174  192 3682
3428
## [16] 1519 1575 1446 3807 1332 4295 3365 1269 1309 1621 3548 1555 3779 3341
2804
```



There were sales spikes between April 13 - 15, April 20 - 22 and April 27 - 29, seems like those days are Friday, Saturday and Sunday. We couldn’t find any publicly available information on events that can trigger these sales spikes. We think that the company might have been running promotions that included pizzas.

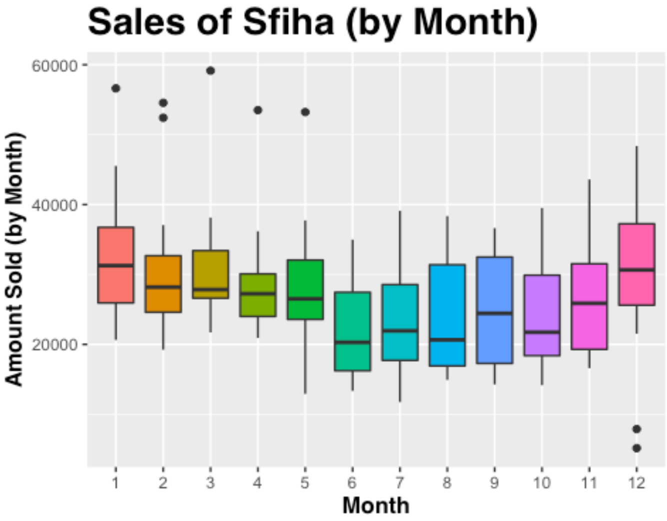
Plotting boxplots for beverage/month and average temperature of the month





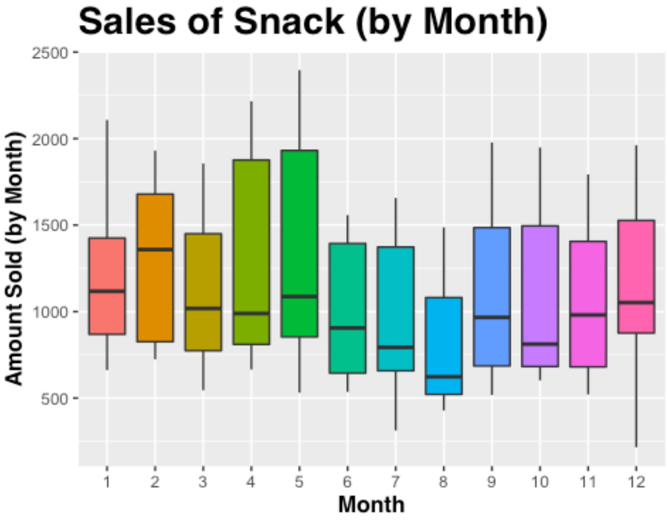
It seems like beverage and temperature has a strong correlation.

Plotting boxplots for sfiha/month

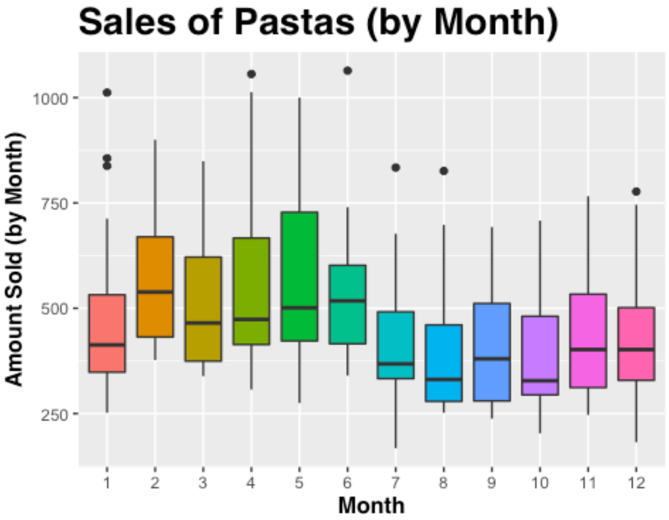


Sfiha sales had a few outliers in a certain month, sales month over month look stable and follows temperature trend.

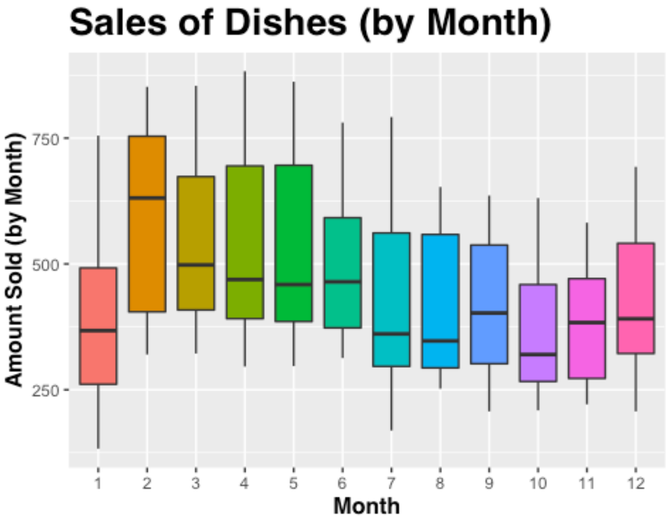
Plotting boxplots for snack/month



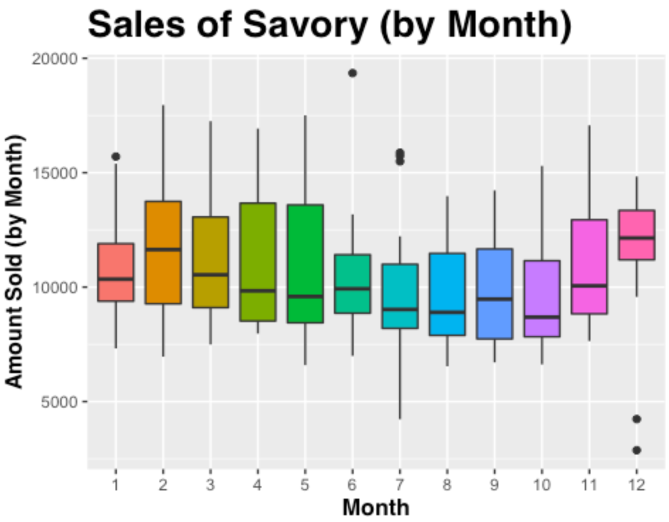
Plotting boxplots for pastas/month



Plotting boxplots for dishes/month

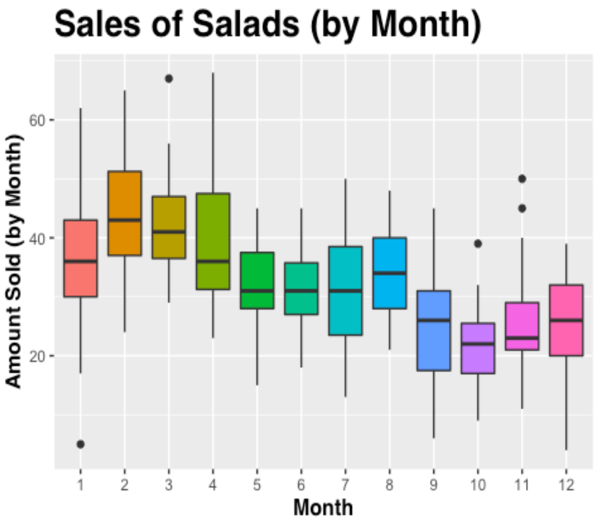


Plotting boxplots for savory/month



Savory slaes follow temperature trend as well.

Plotting boxplots for salads/month



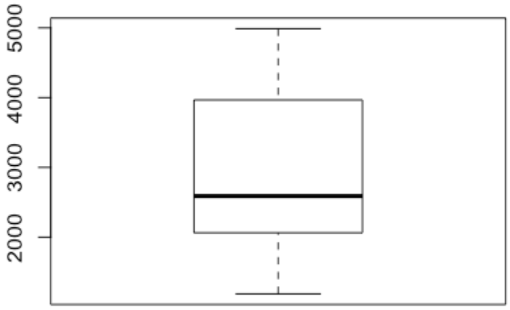
Sales of salads were declining month over month.

Removing Outliers

Removing outliers for desserts

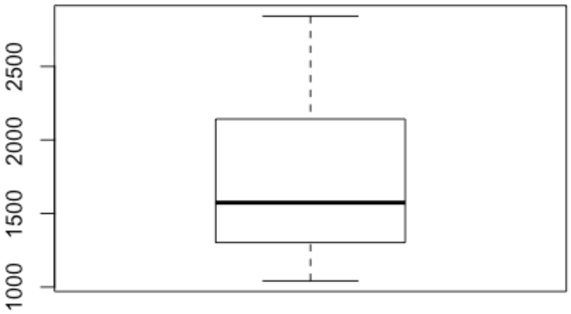
```
desserts_c <- df$desserts
qnt <- quantile(desserts_c, probs=c(.25, .75), na.rm = T)
caps <- quantile(desserts_c, probs=c(.05, .95), na.rm = T)
H <- 1.5 * IQR(desserts_c, na.rm = T)
desserts_c[desserts_c > (qnt[1] + H)] <- caps[1]
desserts_c[desserts_c < (qnt[2] - H)] <- caps[2]
```

```
df$desserts_c <- desserts_c

boxplot(desserts_c)
```



Removing outliers for beverage

```
beverage_c <- df$beverage
qnt <- quantile(beverage_c, probs=c(.25, .75), na.rm = T)
caps <- quantile(beverage_c, probs=c(.05, .95), na.rm = T)
H <- 1.5 * IQR(beverage_c, na.rm = T)
beverage_c[beverage_c > (qnt[1] + H)] <- caps[1]
beverage_c[beverage_c < (qnt[2] - H)] <- caps[2]
boxplot(beverage_c)
```



Removing outliers for sfiha

```
sfiha_c <- df$sfiha
min(sfiha_c)
```

```
## [1] 5164
```

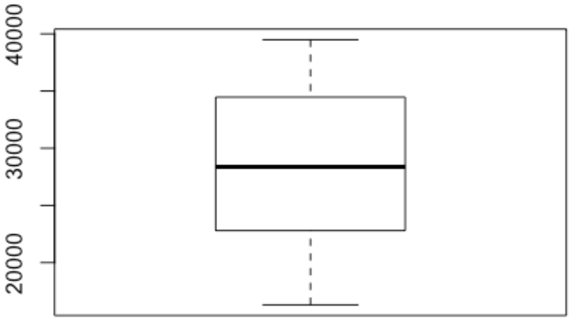
```
sfiha_c[sfiha_c ==  5164] <- 26774.38
sfiha_c[sfiha_c ==  7892] <- 26774.38
qnt <- quantile(sfiha_c, probs=c(.25, .75), na.rm = T)
caps <- quantile(sfiha_c, probs=c(.05, .95), na.rm = T)
head(qnt)
```

```
##      25%      75%
## 21663.75 32641.25
```

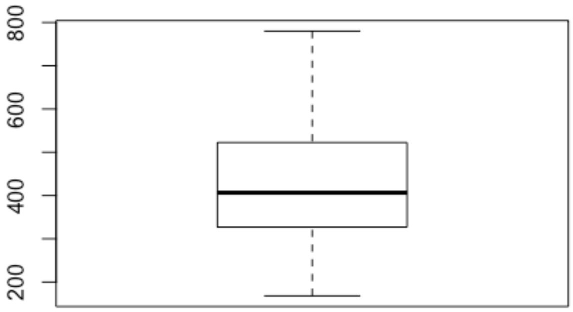
```
head(caps)
```

```
##       5%      95%
## 16078.75 39489.75
```

```
H <- 1.5 * IQR(sfiha_c, na.rm = T)
sfiha_c[sfiha_c > (qnt[1] + H)] <- caps[1]
sfiha_c[sfiha_c < (qnt[2] - H)] <- caps[2]
boxplot(sfiha_c)
```



```
pastas_c <- df$pastas
qnt <- quantile(pastas_c, probs=c(.25, .75), na.rm = T)
caps <- quantile(pastas_c, probs=c(.05, .95), na.rm = T)
H <- 2 * IQR(pastas_c, na.rm = T)
pastas_c[pastas_c > (qnt[1] + H)] <- caps[1]
pastas_c[pastas_c < (qnt[2] - H)] <- caps[2]
df$beverage_c <- beverage_c
boxplot(pastas_c)
```



Converting diasemana to numbers

```
df$day_number <- recode(df$diasemana,
                        "SUN"= 0,
                        "MON"= 1,
                        "TUE"= 2,
                        "WED"= 3,
                        "TUR"= 4,
                        "FRI"= 5,
                        "SAT"= 6)
```

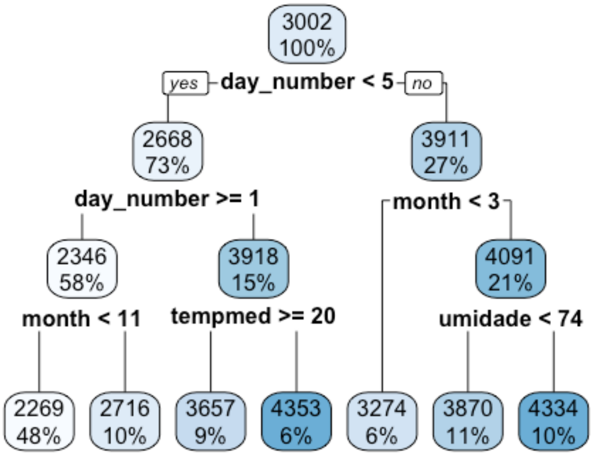
Models

Splitting data train and test %80 and %20

```
date.copy <- df
data.split <- createDataPartition(date.copy$total, p=0.8, list = F)
train <- date.copy[data.split, ]
test <- date.copy[-data.split, ]

dim(train)

## [1] 320  26

dim(test)

## [1] 76 26

train.copy <- train
test.copy <- test
```

Decision Tree Model

```
set.seed(2020)
decision.tree.model <- rpart(desserts_c ~ precipitac + tempmax + tempmin +
tempmed + umidade + insolacao
                             + month + day_number, data=train.copy)

rpart.plot(decision.tree.model)
```



Let’s predict desserts using

```
predict.decision.tree.model <- predict(decision.tree.model, test.copy)

RMSE <- RMSE(pred = predict.decision.tree.model, obs = test.copy$desserts_c)
RMSE/mean(desserts_c)*100

## [1] 26.47081
```

This model gives error around 26%.

Bagging Model

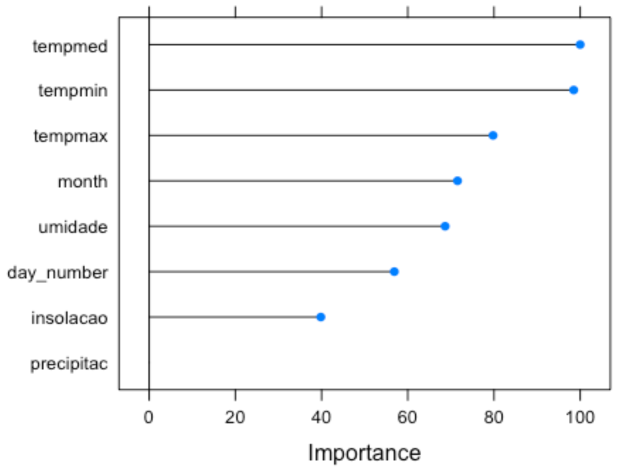
Let’s predict beverages

```
set.seed(2020)
cross.validation <- trainControl(method='cv', number=10)

bagged_cv <- train(
  beverage_c ~ precipitac + tempmax + tempmin + tempmed + umidade + insolacao
+ month + day_number,
  data = train.copy,
  method = "treebag",
  trControl = cross.validation,
  importance = TRUE
)

bagged_cv
```

```
## Bagged CART
##
## 320 samples
##   8 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 289, 288, 288, 288, 288, 288, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   383.5136  0.4006794  272.1513
```

Checking importance of features

```
plot(varImp(bagged_cv), 8)
```



It seems like precipitation does not impact sales of beverages

```
predict.bagged <- predict(bagged_cv, test.copy)

RMSE <- RMSE(predict.bagged, test.copy$beverage_c)
RMSE/mean(beverage_c)*100

## [1] 21.59024
```

This model gives error around 21%.

Random Forest Model

```
set.seed(2020)
library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
random.forest.model <- randomForest(desserts_c ~ precipitac + tempmax +
tempmin
                                    + tempmed + umidade + insolacao +
day_number + month, data = train.copy)

random.forest.model
```

```
##
## Call:
##  randomForest(formula = desserts_c ~ precipitac + tempmax + tempmin +
tempmed + umidade + insolacao + day_number + month, data = train.copy)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 473075
##                     % Var explained: 53.02
```

```
which.min(random.forest.model$mse)
```

```
## [1] 372
```

```
sqrt(random.forest.model$mse[which.min(random.forest.model$mse)])
```

```
## [1] 684.8695
```

```
predict.random.forest <- predict(random.forest.model, data = test.copy)
```

```
RMSE <- RMSE(predict.random.forest, test.copy$desserts_c)
```

```
## Warning in pred - obs: longer object length is not a multiple of shorter
object
## length
```

```
RMSE/mean(desserts_c)*100
```

```
## [1] 40.63209
```

This model gives error around 40%.

The results

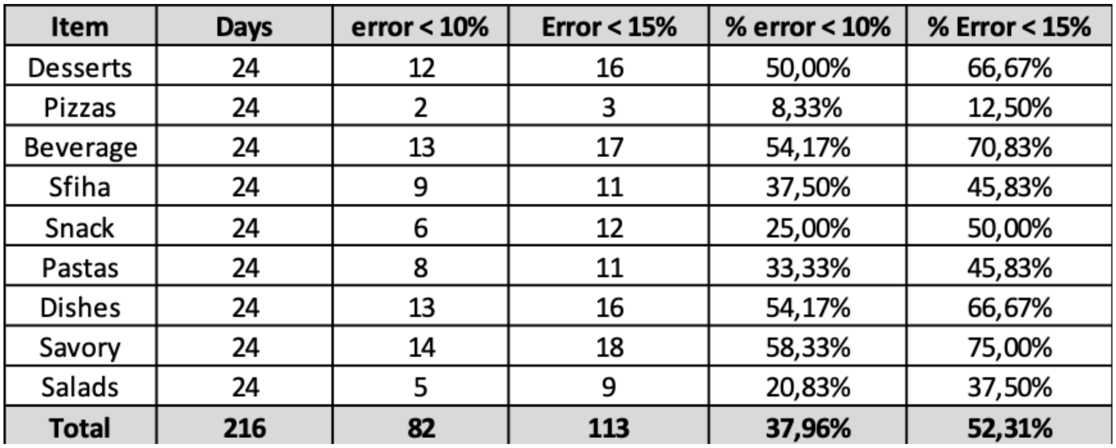
After running all models, we picked a linear regression since it gives a lower margin of errors. When analyzing how we should evaluate the results at the beginning the answer seemed to be very straight forward: How many times the system got the prediction right. However, thinking a bit more some additional considerations appear:

1) Very unlikely that the prediction would be exactly right (How far from the mark seems to be the best measure.

2) We have nine predictions per cycle and have several cycles (Each day is a cycle in this context), how many times it got within a giving limit seems to be the best measurement.

3) We have a current process to compare with, how many times it got better than the current one.

With all this in mind we produced the following spreadsheet:



Item	Days	error < 10%	Error < 15%	% error < 10%	% Error < 15%
Desserts	24	12	16	50,00%	66,67%
Pizzas	24	2	3	8,33%	12,50%
Beverage	24	13	17	54,17%	70,83%
Sfiha	24	9	11	37,50%	45,83%
Snack	24	6	12	25,00%	50,00%
Pastas	24	8	11	33,33%	45,83%
Dishes	24	13	16	54,17%	66,67%
Savory	24	14	18	58,33%	75,00%
Salads	24	5	9	20,83%	37,50%
Total	**216**	**82**	**113**	**37,96%**	**52,31%**

As we can see the system got an uneven performance, being much better predicting some types of products, than others. In general this first version didn´t perform better than the manual process today in place.

What could be done to improve the results

Analyzing the results we start understanding the reasons why the performance wasn´t the one we expected. We identified at least four initiatives which if taken surely would improve the results :

1) We are segmenting the training data by type of day and type of month, doing that we create training subsets which sometimes were very small: E.g. the combination type of day 2 (Sunday) and month 2 (Dec and June) have only 8 samples (maybe less if part of them were segmented to the test data). Adding more historical data to the training data surely will improve system performance. This problem tends to be solved over time as new data is added to the system.

2) There is a problem with some of the sales volumes of our sample. Some products have completely abnormal volumes of sales (too high or too low). We did coded a Bell-curve tail with the objective of expurgating the samples which would distort the results. However, giving problem 1 (lack of samples) we were unable to eliminate the adequate number of such cases; otherwise, we would end up with even fewer samples.

3) We believe that there is some cross-correlation between the weather parameters which may be more effective as predictors than the parameters themselves. Due time constraints we didn’t explore this avenue.

A secondary issue that may or may not have relevance in this process is the bias regards under-producing. We realized that real sales may hide a demand which wasn´t met due to the unavailability of products. That means we may have a day with an unusually low sales of a specific product. We don´t know and didn´t figure out a way of guessing how frequently this phenomenon may happen.