Brazil Rent Report - Data Analysis for Business 2023

Eduardo Mosca, Chakir El Arrag, Iraj Nayebkhil

2023-05-23

In this report, we look at the dataset for rent prices in Brazil, specifically along the south border, looking for specifics of variables and characteristics which might help paint the picture for our (hypothetical) employer's entry in the market. We now take a look at the dataset and its variables.

1 - Overview of the dataset

Of the 12 variables on hand, five of them continuous (fire insurance, property tax, HOA fee, rent amount and area), three are categorical(animal accepted,apartment furnished, and city where property is located). The remaining four are numerical discrete, they represent the floor number and the number of rooms, bathrooms, and parking spots. We have a rough dataset of 10692 observations, not a lot but enough to extract valuable info. We start cleaning the dataset: there are no specific NA's but the floor has "-" values. By calling anyDuplicated we also notice some duplicate rows, we want to remove them with "unique()".

The floor column has a '-' value which is pretty occurent in our dataset.

```
## [1] "The number of observations with floor '-' is: 2369"
```

We have 2369 observations with floor "-" on our dataset, interestingly the minimum value in the floor column is 1, which suggests that "-" would represent the ground floor or houses that are not in a condominium. To be sure, we extract from the dataset the HOA (Homeowners Association Tax) for houses on the '-' floor.

```
## [1] "Number of observations with floor '-' and HOA = 0 2015"
```

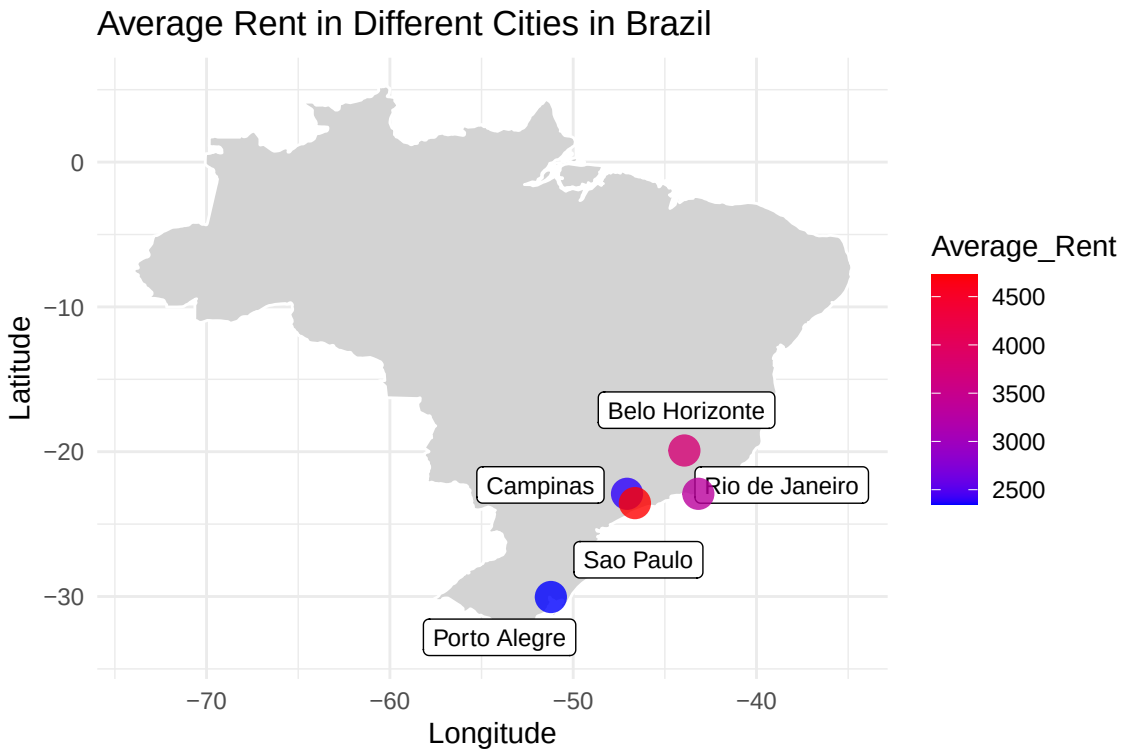
```
## [1] "Proportion of observations with floor '-' and HOA = 0 0.850569860700718"
```

As we can see, 85,06% of the observations has an HOA of 0, which strengthens our hypothesis since owners of normal houses don't have an HOA tax to pay because the houses are not situated in a condominium. The rest of the occurences with HOA higher than 0, can be justified by the presence of some condominiums with houses on the ground floor, not very frequent but this would justify why they're paying the HOA tax while at the same time being at floor '-'. Since the data is compatible with our hypothesis, we'll proceed by replacing '-' by 0.

Next, we set the variables to be either numeric or factors.

2 - Looking at the response variable

We can look at the geographical position of the cities we have info on, and visualize which ones are more expensive on average.



As we can see Sao Paulo tends to be the most expensive place to rent a house.

A high correlation is observed between fire insurance and rent, suggesting that fire insurance could be a valuable variable for modeling.

```
## [1] 0.9872013
```

Defining our objective

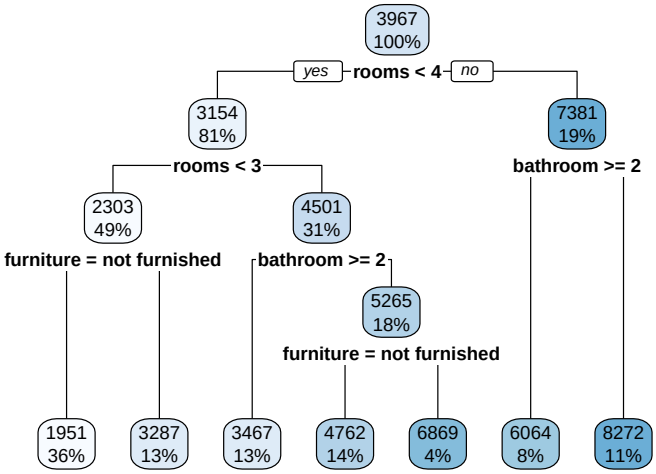
Our objective is to *Build a predictive model and find out the rent amount according to the house specifics*, by using regression methods on the response variable rent.amount..R..

3 - Lower Dimensional Models

When thinking about low dimension models, we must understand which ones might apply to our use case, yielding information that is actually valuable. We need to better understand relations between the response and some variables. Its valuable to know how much the rent increases as the number of rooms or area increases. We first implement a Log-Linear regression model to see what percentage increase there is when the area goes up by one unit:

```
##                Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) 7.9303056926 7.946043e-03 998.01942 0.000000e+00
## area        0.0002528586 1.401744e-05  18.03886 1.207089e-71
```

By looking at the coefficients, we see the expcted % increase following an increase in the area by 1 unit. We also want to model the reasoning of a person interested in renting an apartment in Brazil via deicision tree, by considering the main concerns when house shopping. This will give us insight on the price levels according to specifics:



The specifics leading to higher rent are to the right, while following the left-most branch we get cheaper rent.

4 - Getting the data ready

We start by removing outliers in the continuous variables using the IQR. For discrete numerical variables, we give a look at the boxplots and we remove outliers manually.

Then, we turn the variables animal and furniture to 0s and 1s, the variables in the original dataset take a binary value encoded as a character (ex. for animal we have acept and not acept)

After that we split the dataset into training validation and test sets, and we scale them using the mean and standard deviation of the training set. We save main training set (training + validation) to work with it later for the test set prediction.

Finally, we encode categorical variables (We keep an unencoded version of the training and validation sets for later).

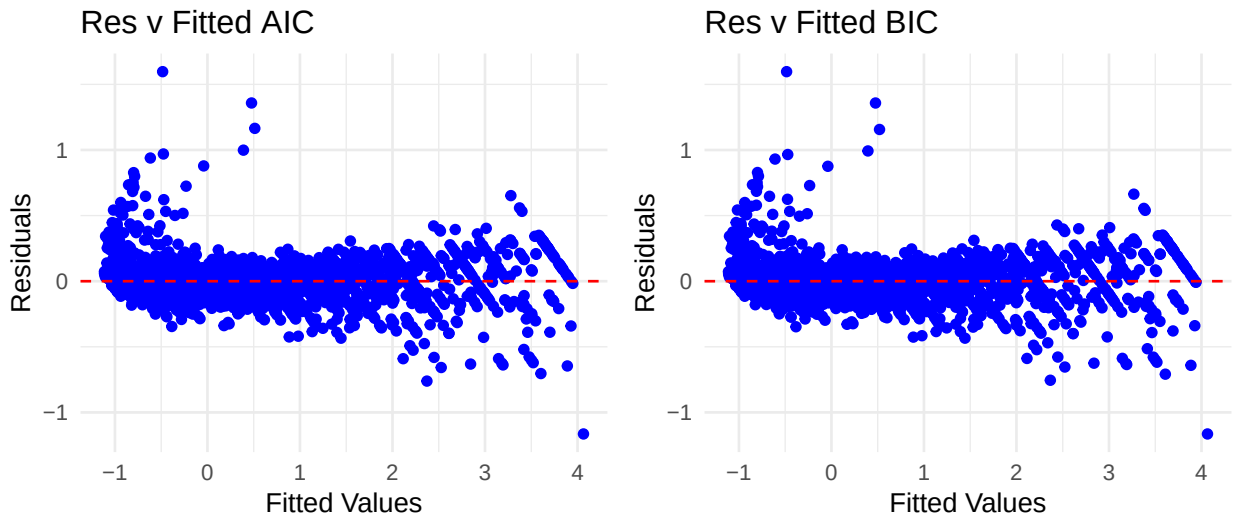
5 - Testing the models' performances, Model Selection

5.1 - AIC and BIC

First we implement AIC and BIC stepwise selection for multiple regression models, then we evaluate the performance on the validation set using the following metrics: **MSE**, **RMSE** and **R Squared**.

```
##   Model     MSE    RMSE R_squared
## 1   AIC 0.01437 0.11986   0.98842
## 2   BIC 0.01435 0.11978   0.98841
```

Our two models tend to yield similar results, BIC tends to be slightly better, so we're gonna consider it over AIC. We can also plot the residuals against the fitted values of each:



The two are pretty similar and for the most part the points hover around the 0 line, which indicats a good fit. When getting a similar result for AIC and BIC we can assume to be striking a good balance between model complexity and goodness of fit.

5.2 - Lasso and Group-Lasso

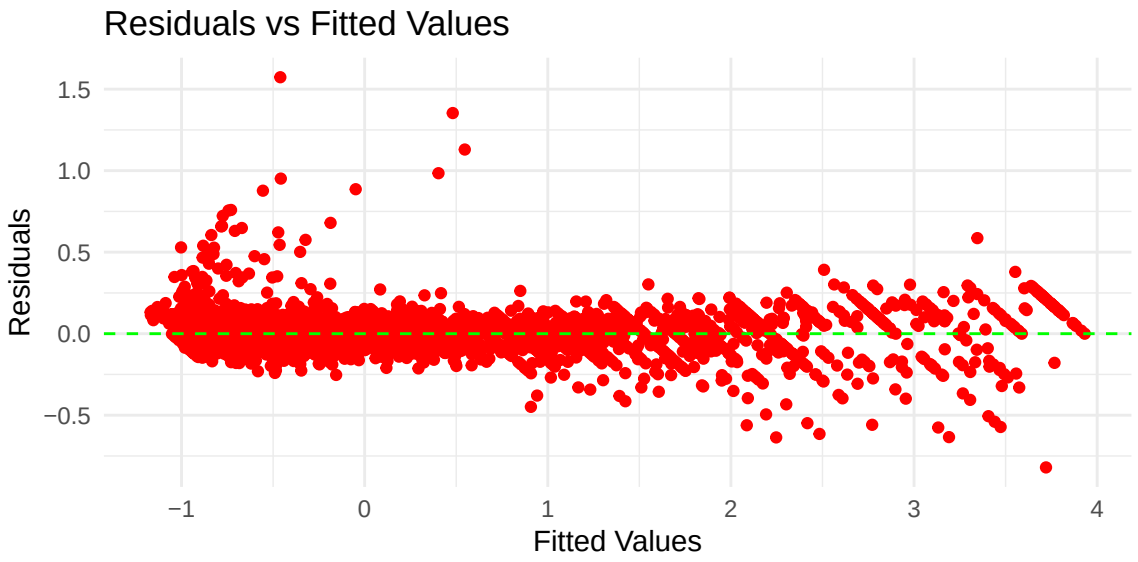
Let's implement a lasso regression using glmnet and group-Lasso using gglasso. We start directly from 10-fold cross validation, and then we evaluate the model on the validation set using the lambda.min and lambda.1se

The lambda.min values get a lower error on the validation set, so let's compare the two models using minimum lambda:

```
##     Model     MSE    RMSE R_squared
## 1   Lasso 0.01426 0.11942   0.98523
## 2 grLasso 0.01432 0.11967   0.98516
```

5.3 - GAM

We create the smoothing terms for numerical and categorical variables (We're going to use the unencoded versions of the training and validation sets we saved earlier), then we create the GAM formula summing the smoothing terms and we feed it to the model to do the fitting.



In this case as well, most of the points hover around the 0 line, with most of the residuals being in the interval $[-0.5, 0.5]$, which indicates a good fit. After, we predict on the validation set and we evaluate the prediction using the metrics mentioned previously. We'll save them for the models comparison.

5.4 - XGBoost

Now we create an XGBoost model. We start by splitting dependent and independent variables and transforming the training and validation sets to xgb.Dmatrices which will be used for training and prediction.

We use 10-fold cross validation for parameter tuning. The parameters we want to tune are the following: nrounds(Number of boosting rounds), max_depth(Maximum tree depth), eta(Learning rate)

We put the parameters to tune and their respective values the we would like to test in a parameter grid. Then, we use the parameter grid as an input to perform 10-fold cross validation in order to get the best combination of values for the parameters.

Finally, we fit the final XGBoost model with the best parameters we got, we proceed to predict on the validation set and we evaluate the prediction, We save the errors models comparison.

5.5 - Model comparison

We make a table with the errors for the models we tested until now

```
##     Model     MSE    RMSE R_squared
## 1     BIC 0.01435 0.11978   0.98841
## 2   LASSO 0.01426 0.11942   0.98523
## 3     GAM 0.01308 0.11438   0.98644
## 4 XGBoost 0.00787 0.08874   0.99184
```

We notice that the model that performs the best is XGBoost, we're going to use it to predict on the test set.

6 - Prediction on the test set

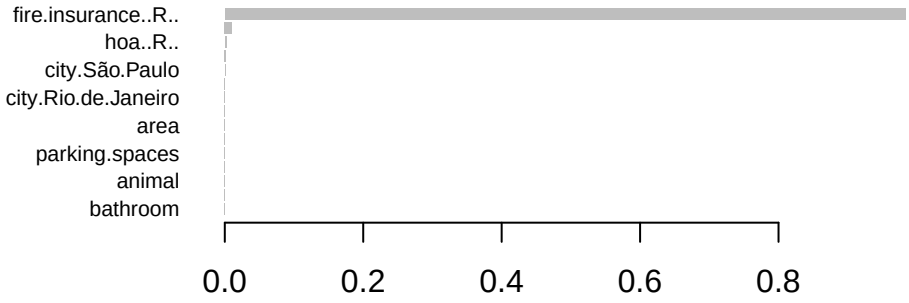
We start by scaling the training set (training + validation) and we encode its categoricals, then we transform the training and test sets to xgb.Dmatrices. Next, we fit the model and we make predictions on the test set without forgetting to evaluate our predictions.

```
## [1] "MSE:  0.01279"
```

```
## [1] "RMSE:  0.08874"
```

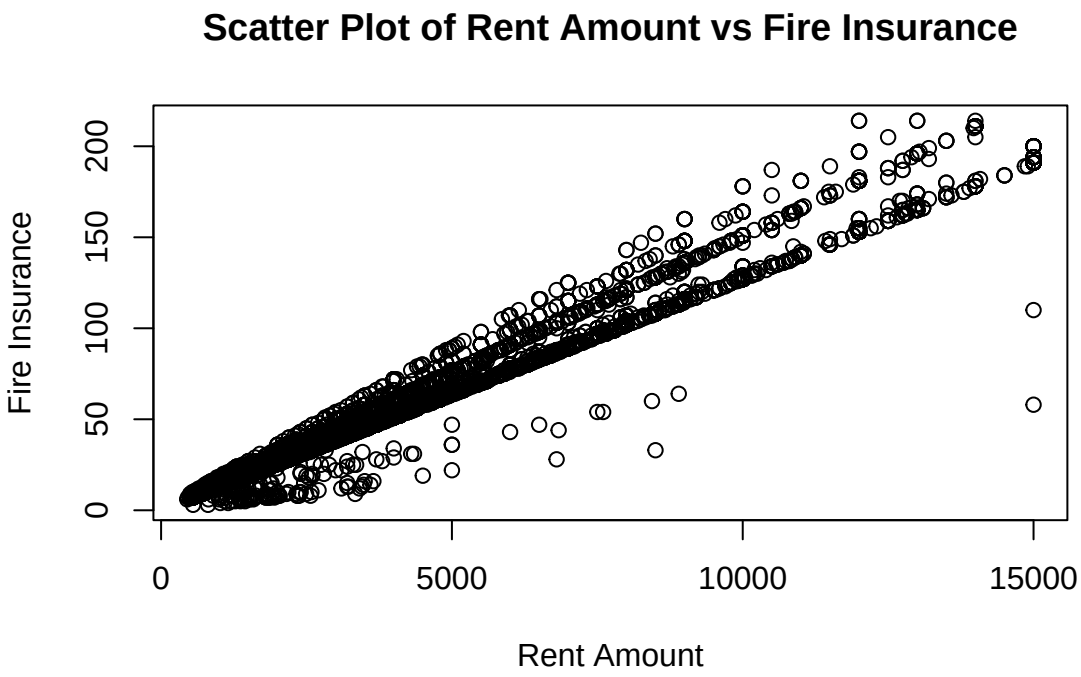
```
## [1] "R Squared:  0.99184"
```

Test and validation set values are similar, this means that the model is consistent and performs well on unseen data. Having low errors, we can allow accurately predict rent prices in important cities in Brazil, we can use this model to assess the returns and company concerns regarding chargeable rent before investment. By leveraging cost data we can spot houses that our business can invest in with higher turnover. Moreover, an high R-squared value indicates that a large proportion of the variance is explained by the variables used in our model. It implies that there is a strong and consistent relationship between the independent variables and the rent prices. We can say that the independent variables could be enough and we might not need to add more variables for this purpose. Now we should evaluate each variables' importance.

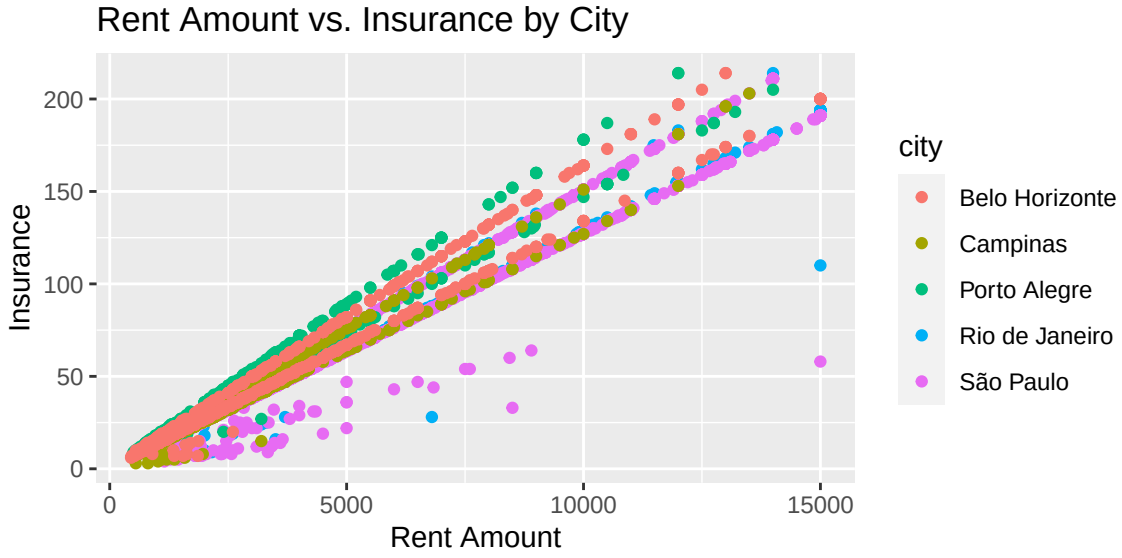


From the plot, fire.insurance..R.. is the most important variable. City variables stand out, with Porto Alegre and São Paulo being the most significant, probably since Porto Alegre has the lowest average rent price, while São Paulo has the highest. Floor and hoa..R.. are also important variables. Surprisingly, the house area and number of rooms are less significant, suggesting that location matters more. For instance, renting a larger house in Porto Alegre may be cheaper than a smaller apartment in São Paulo's city center.

The `fire.insurance..R..` variable, predictably, has a very high importance, let's make a scatterplot to see if there's any linear/non-linear relationship with the response.

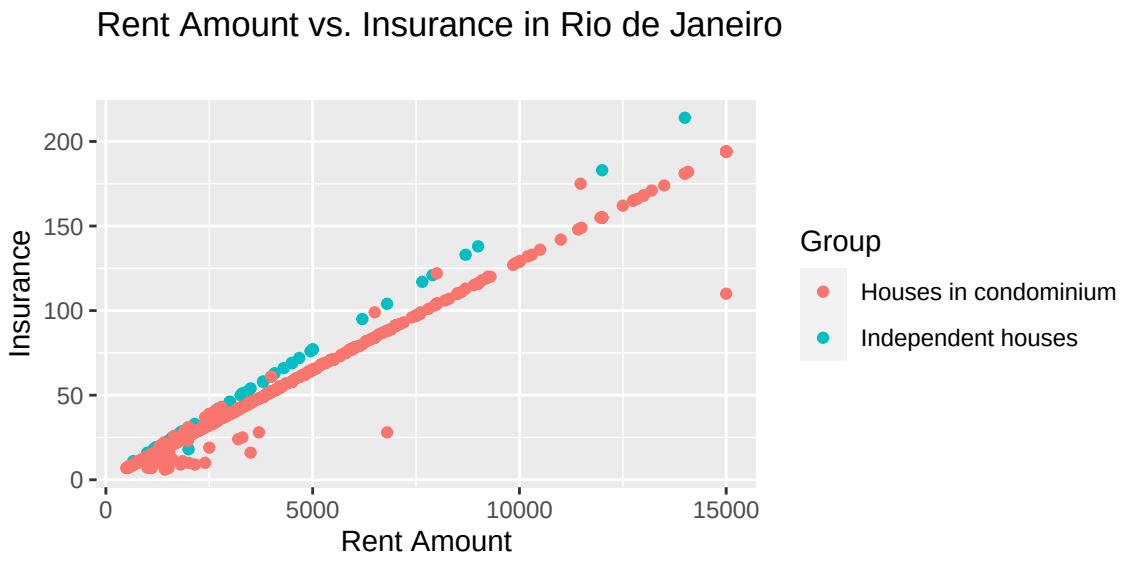


There is a strong linear relationship between fire insurance and rent amount. This is not strange since fire insurance premiums are often based on the value of the insured property and the associated risk factors. In this case, higher rent prices are typically associated with properties of higher value. Properties with higher values are likely to have higher replacement costs and, consequently, higher fire insurance premiums. It is worth to mention that insurance premiums can be also influenced by the location of the property. Indeed, from the plot we see multiple lines, we can assume that each line could be representing a city.



The plot confirms that location affects fire insurance and rent amount. Lines representing different cities show that lower average rent cities have steeper slopes. This indicates that for properties of the same value,

the city with lower average rent has higher fire insurance due to larger properties with higher replacement costs. Each city has two distinct lines, suggesting another factor at play. We suspect that the combination of floor and hoa..R.. indicates whether a house is in a condominium. A floor value of 0 signifies either a ground floor condo or an independent house. The presence of a non-zero hoa..R.. value indicates a condominium. To investigate further, we can create a plot for Rio de Janeiro, differentiating between houses in condominiums and independent houses.



As we can see, our assumption was correct.

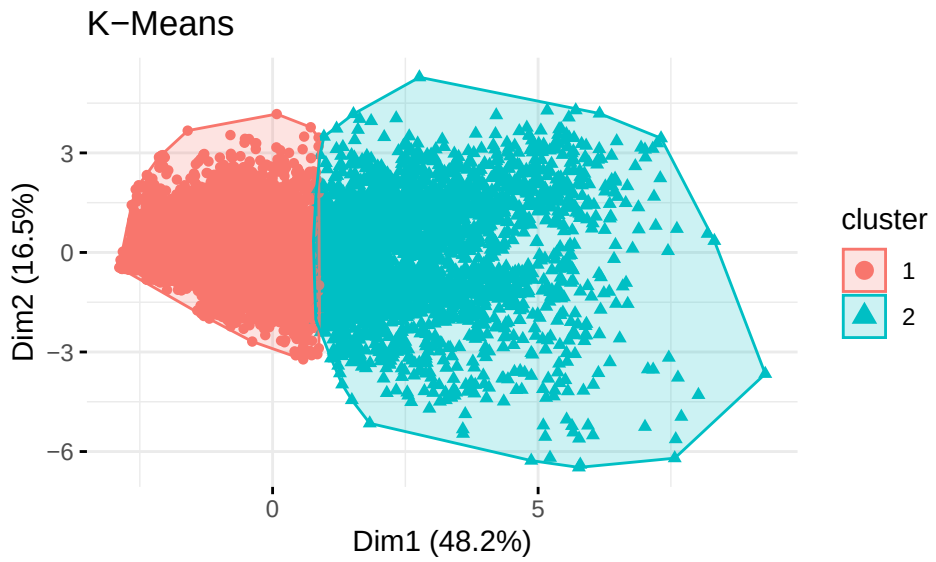
Task 2

Our second objective is to *Cluster the houses for rental according to their characteristics*; we are going to compare K-Means and Hierarchical Clustering choosing an optimal number k of clusters with the silhouette and elbow methods. We're gonna compare partitions and average silhouette width to determine which one best resembles reality, we're gonna evaluate agreement between the partitions and the different values in each cluster with respect to the initial dataset.

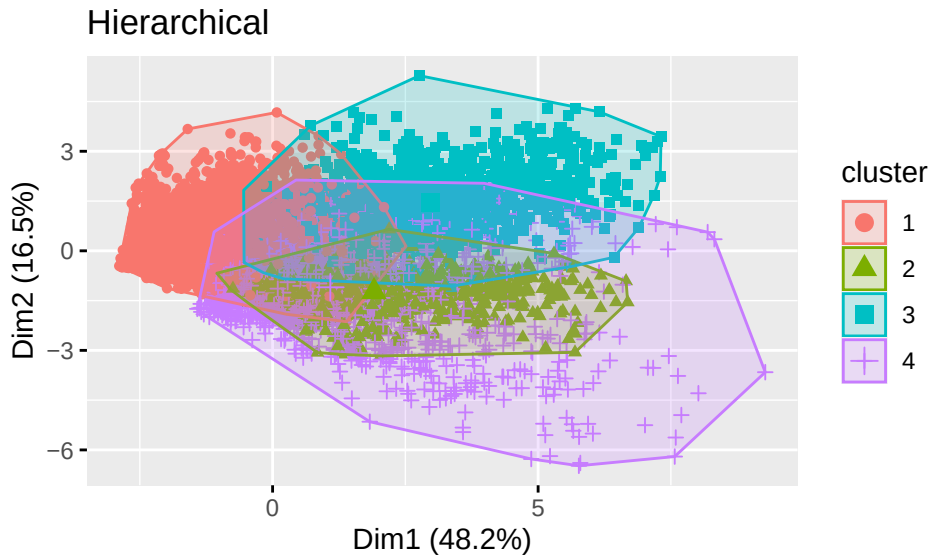
We get following results for best number of clusters silhouette wise:

```
## [1] "Best k for K-Means(sil): 2 , for HC(sil): 4"
```

Also consulting the elbow method plot, we want to employ K-Means with k = 2 and Hierarchical with k = 4, both using euclidean distance as it proved more efficient for us. We get the following partitions:




```
## [1] "K-Means AVG Silhouette width: 0.391492088470082 , totWSS: 57591.2398825153"
```




```
## [1] "Hierarchical's AVG Silhouette width: 0.308704751306795"
```

We're more convinced with the K-Means partition as it has higher average silhouette and partitions the datapoints clearly, while the hierarchical one has a lot of overlap between clusters, so we could reasonably say it has lower between-sum-of-squares and higher within-sum-of-squares.

By looking at the agreement index we can deduce more about the difference between partitions:

```
## [1] "Agreement betweenn K-Means and HC 0.479226606836096"
```

The two partitions differ in number of clusters and algorithms used, since we feel more confident about the K=2-Means partition as it best represents our image of the data(i.e. lower end cheaper housing and higher end more expensive housing) we're going to rely on the K-Means partition for further analysis, so let's see what it represents with respect to the initial dataset:

```
## [1] "Average rent amount in: Cluster 1: 2208.685 , Cluster 2:  6968.054"
```

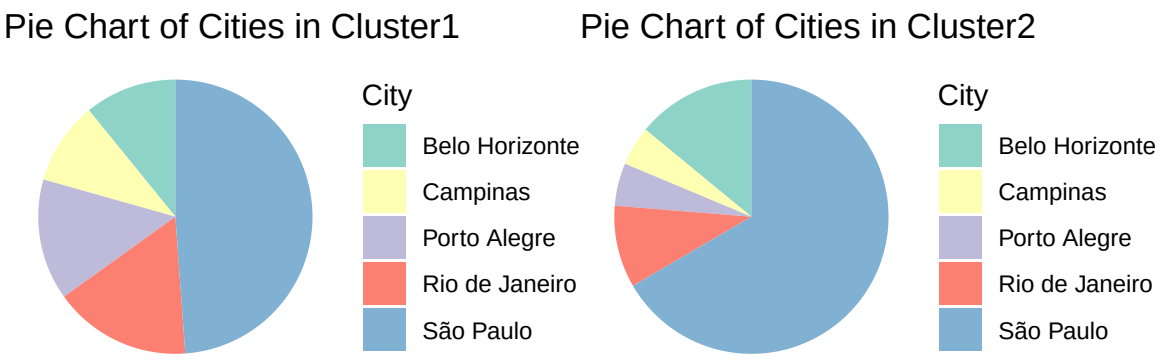
As we can see average rent amount in cluster 2 is more than 3 times that of cluster 1, as we've mentioned before we think this reflects lower and higher end housing, but to come to more solid conclusions let's see some more data:

```
## [1] "Percentage of houses in Cluster 1: 0.713 , in Cluster 2: 0.287"
```

```
## [1] "Average property tax in Cluster 1: 106.086 , in Cluster 2: 583.458"
```

```
## [1] "Average rent per room in Cluster 1: 1277.183 , in Cluster 2: 2159.968"
```

Let's also plot a pie chart for cities with respect to cluster



Drawing conclusions on Task 2

A partition in higher and lower end housing shows dynamics of the region of brazil we have data about, where the majority of houses are located in Sao Paulo which is the most expensive place on average but also the most frequent city overall, which means that there can be both more and less affordable housing there. The data also confirms the fact that tax and rent per room follow act similarly as rent in the different clusters. As for our business' interests: If they want to acquire a larger volume of real estate with lower rent charged but probably lower acquisition cost, we could suggest to go for one type of house (cluster 1) If on the other hand they're more interested in high end houses to charge higher rent(still, with higher acquisition cost), we'd point them to the second cluster. If they wished to diversify their real estate we would be able to identify a potential acquisistion as more, or less expensive.