Universitat Politècnica de Catalunya
Facultat d'Informàtica de Barcelona

2024/2025

Project development

D4

24.03.2025

Teacher: Xavi Angerri
Authors: Bednár Maroš, Escofet González Gina, Foroudian Kimia,
Lafuente González Alex, Smyhelskyy Yaskevych Sergio

Table of contents

1. Motivation and Problem Analysis

Today's hospitality industry faces the ongoing challenge of optimizing room availability and pricing strategies while reducing last-minute cancellations. With the increasing amount of data collected by hotels—ranging from guest demographics to booking channels—data mining techniques can reveal patterns that help hoteliers make data-driven decisions. This project focuses on a **hotel booking dataset**, aiming to explore the factors that may lead to reservation cancellations and to derive meaningful customer segments based on booking behaviors. These insights can guide management in creating targeted marketing campaigns, fine-tuning pricing strategies, and improving guest satisfaction overall.

The broader goal is to illustrate how systematic data mining (from preprocessing and descriptive statistics through more advanced methods such as Principal Component Analysis (PCA) and Hierarchical Clustering) can provide practical value in a real-world business context. The problem under analysis is to understand which variables or booking features are most influential in cancellation patterns, how bookings can be grouped into distinct clusters, and how these insights can help hotel managers optimize operations. Moreover, by taking a structured approach to data mining, teams can practice and demonstrate key techniques like data cleaning, feature transformation, exploratory analysis, and interpretive modeling.

2. Data Source - Hotel Bookings

All data for this project was obtained from a Kaggle repository (https://www.kaggle.com/datasets/mojtaba142/hotel-booking) and originally described in the article "Hotel Booking Demand Datasets" by Nuno Antonio, Ana Almeida, and Luis Nunes (https://www.sciencedirect.com/science/article/pii/S2352340918315191). The records capture bookings made at two hotels from July 1, 2015, to August 31, 2017, with detailed information on reservation dates, number of guests (adults, children), and whether each booking was ultimately cancelled. The original dataset contains 119,390 rows and 36 variables, although a subset of 5,000 rows is used here.

3. Data Structure and Metadata

3.1 What the Rows of the Data Matrix Contain

Each **row** in this dataset represents an **individual hotel booking**. Specifically, a row provides information about:

- **Booking dates** (arrival date, lead time, etc.)
- **Guest demographics** (number of adults, children)
- **Stay details** (number of weekend nights vs. weekdays, room type requested, etc.)
- **Financial features** (average daily rate, deposit type, etc.)
- **Booking channel data** (agent, distribution channel)
- **Outcome variables** (whether the booking was canceled, whether the reservation was repeated, etc.)

Hence, each record is a unique booking event with attributes describing the reservation details and the guests involved.

3.2 Metadata Table

Below is a **sample** (not exhaustive) metadata table for the most critical variables. You can expand this table to include all 36 variables as needed:

Variable	Variable Short Name	Description	Variable_Type	Measuring_ Unit	Range	Role
hotel		Hotel type (City Hotel or Resort Hotel)	Categorical	-	-	-
is_canceled	can	Indicates whether the booking was canceled (1) or not (0)	Binary	-	[0, 1]	Response
lead_time	lt	Days between booking date and arrival date	Numerical	days	[0, 737]	Explanatory
arrival_date_year	arr_y	Year of the arrival date	Categorical	-	[2015, 2017]	Explanatory
arrival_date_month	arr_m	Month of the arrival date	Categorical	-	-	Explanatory
arrival_date_week_number	arr_wn	Week number of a year of the arrival date	Categorical	-	[1, 53]	Explanatory
arrival_date_day_of_month	arr_dm	Day of the month for the arrival date	Categorical	-	[1, 31]	Explanatory
stays_in_weekend_nights	s_wend_n	Number of weekends	Numerical	nights	[0, 19]	Explanatory

		stayed				
stays_in_week_nights	s_wday_n	Number of weekday nights booked or stayed	Numerical	nights	[0, 50]	Explanatory
adults		Number of adults in the reservation	Numerical	count	[0, 55]	Explanatory
children		Number of children in the reservation	Numerical	count	[0, 10]	Explanatory
babies		Number of babies in the reservation	Numerical	count	[0, 10]	Explanatory
meal		Meal type chosen by the client	Categorical	-	-	Explanatory
country		Origin country name of the client/s	Categorical	-	-	Explanatory
market_segment	mkt_s	Market segment destination	Categorical	-	-	Explanatory
distribution_channel	dst_ch	Booking distribution channel	Categorical	-	-	Explanatory
is_repeated_guest	NOT COVERED		Binary	-	[0, 1]	Explanatory
previous_cancellations	prev_can	Number of previous cancellations made by the client	Numerical	-	[0, 26]	Explanatory
previous_bookings_not_canceled	pb_nc	Number of previous bookings not canceled made by the client	Numerical	-	[0, 72]	Explanatory
reserved_room_type	res_rt	Code of room type reserved	Categorical	-	-	Explanatory
assigned_room_type	as_rt	Code for the type of room assigned to the booking	Categorical	-	-	Explanatory
booking_changes	bk_ch	Indicates if the client made a deposit	Numerical	-	[0, 21]	Explanatory
deposit_type	dp_t	Type of deposit (No Deposit, Refundable, Non Refund)	Categorical	-	-	Explanatory
agent		Booking agent ID	Categorical	-	[1, 535]	Explanatory
company		Company ID for corporate bookings	Categorical	-	[6, 543]	Explanatory
days_in_waiting_list	d_wl	Number of days booking was in the waiting list before it was confirmed to customer	Numerical	-	[0, 391]	Explanatory
customer_type	c_type	Type of booking	Categorical	-	-	Explanatory
adr		Average Daily Rate	Numerical	currency (daily rate)	[-6.38, 5400]	Explanatory

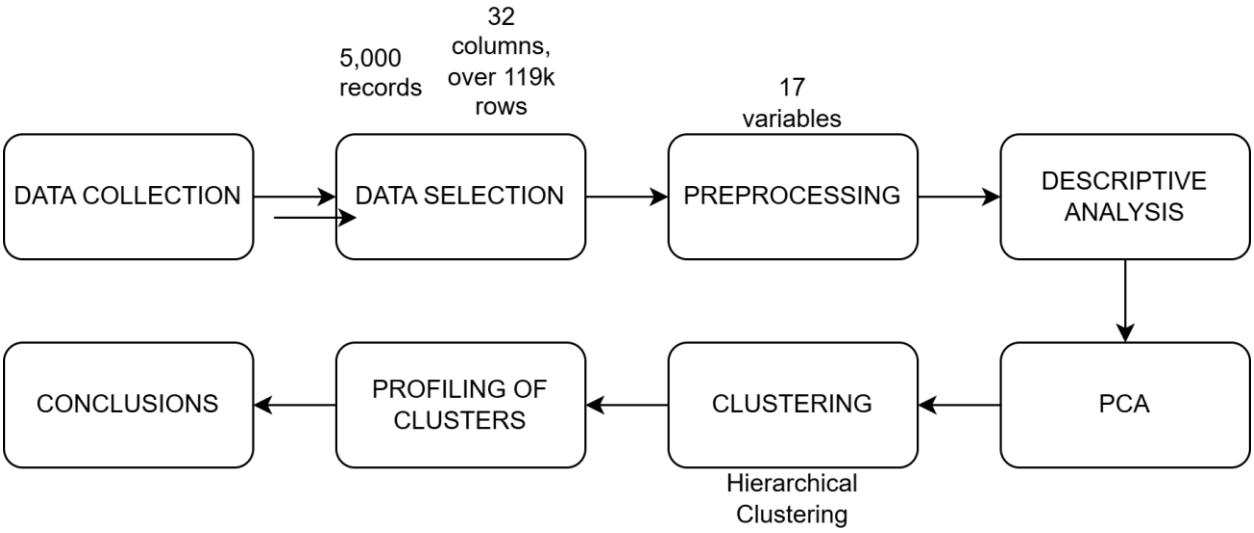
required_car_parking_spaces	req_prk	Number of parking spaces required by the customer	Numerical	count	[0, 8]	Explanatory
total_of_special_requests	sp_req	Number of special request made by the customer	Numerical	count	[0, 5]	Explanatory
reservation_status	res_s	The last status of the booking (Check-Out, Canceled, etc.)	Categorical	-	-	-
reservation_status_date	res_sd	Date at which the last status was set	Date	-	-	-
name	NOT COVERED		Categorical	-	-	-
email	NOT COVERED		Categorical	-	-	-
phone.number	NOT COVERED		Categorical	-	-	-
credit_card	NOT COVERED		Categorical	-	-	-
arrival_date	arr	Date at which the customer arrived	Date	-	-	Explanatory

4. Data mining

Data mining can be applied to hotel booking datasets to predict the likelihood of a reservation being canceled. By identifying cancellation patterns and estimating their probability, it becomes possible to analyze customer behavior more effectively. The dataset used in this research consists of hotel booking information.

The data mining process began with selecting a suitable dataset from Kaggle that aligned with the project's objectives. After choosing the dataset, its contents (comprising 32 columns and over 119,000 rows) were examined, and a sample of 5,000 records was extracted. During data cleaning and preprocessing, the dataset was simplified to focus on the 17 most relevant variables (as justified in section 5 below).

Once preprocessing was complete, a descriptive analysis (both univariate and bivariate) was carried out, followed by a Principal Component Analysis (PCA) to evaluate data separability. Finally, hierarchical clustering was performed to identify potential groupings, allowing for a detailed profiling and individual analysis of the resulting clusters.



1. Data Collection: Identify and choose a dataset from Kaggle that fits the project criteria.
2. Data Selection: Examine the full dataset (32 columns, over 119k rows) and extract a representative sample (5,000 *random* records).
3. Data Cleaning and Preprocessing: Simplify the dataset by selecting the 17 most significant variables, and introduce controlled missing data if necessary (not our case).
4. Descriptive Analysis: Perform univariate and bivariate analyses to understand variable distributions and relationships.
5.PCA - Dimensionality Reduction: To evaluate data separability and reduce complexity.
6. Hierarchical Clustering: Execute clustering on the processed data to identify distinct groupings.
7. Profiling Interpret and understand the characteristics of each group.

5. Preprocessing and Data Preparation

5.1 Determine formatting issues:

We did not have any formatting issue when introducing the command needed to read our CSV file (read.csv).

5.2 Determine data matrix:

5.2.1 Object selection

Since the initial database had 119390 rows, we created an R script to generate a random subset of 5000 rows.

5.2.2 Variables selection

Firstly, we decided to discard three columns: “credit_card”, “phone_number”, and “email_address”. Those variables were personal information that did not contribute to any of this study’s objectives. In addition, they did not even contain real data for privacy purposes, which made them more pointless.

We also removed “reserved_room_type” and “assigned_room_type”, since the dataset and its source didn’t give enough information about each room type. We did the same with “market_segment” and “distribution_channel”, because we considered they were not relevant, and we could not get a grasp on their differences.

Other variables we decided not to include are “reserved_room_type” and “assigned_room_type”, because representative information about room types were not included neither in their values nor in the source (room types were “A, B, C, D” with no real information about their features).

Variables like “required_car_parking_spaces”, “deposit_type”, “is_repeated_guest”, “total_of_special_requests” and “customer_type” did not add relevant information regarding the objective of this study.

Following the same criteria, and in order not to make the data analysis too complex for the learning purposes of this project, we chose to remove other variables, as “booking_changes”, “previous_bookings_not_cancelled” and “previous_cancellations”.

The variables “arrival_date_day_of_month”, “arrival_date_month” and “arrival_date_year” could be merged, while conserving the same information. Therefore, we decided to join them in a single

column, “arrival_date”, in date format. However, we decided to keep “arrival_date_month” (while discarding the others), since it might be useful to analyze the bookings on that scale.

Furthermore, the “company” variable’s missing data was of 94%. This deemed the information unusable and also untreatable by any of the methods presented in class. Firstly, since its values are more than 30-50% null, it is more practical to discard it, and secondly, considering that this variable is qualitative, the null values cannot be imputed.

There was another variable, “agent”, which had null values, around 14%. This column’s problem was that their non-null data wasn’t descriptive enough for it to be a useful field in our project. Mainly since it was agents’ identifiers (like the previously mentioned variable “company”), and both are too limited due to anonymity reasons, we have classified both variables, “company” and “agent”, as non-random missing values

5.3 Check missing data

Since we discarded all the variables which contained missing values, it is not necessary to apply any kind of treatment or algorithm to normalize the dataset.

5.4 Outlier detection

In “adr”, we had found an outlier, it being a value of 508 in Average Daily Rate. Since it could be caused by a day in which the earnings were very high, due to rental of several expensive rooms, we kept it by classifying it as en extreme value of the population.

The “babies” variable indicates that the vast majority of bookings do not include babies. This made us consider the other bookings that have at least a baby as outliers, but their existence still makes total sense, as some costumers might be parents. The same logic applies to “children”.

The “lead_time” variable has many instances where people book far before than their arrival time, but considering that there are a considerable amount of them, we thought it was appropriate to treat them as extreme values, which would also give us useful information overall. Moreover, removing them would lower the quantity of data and the quality of the results.

Regarding the “days_in_waiting_list”, we found a few outliers, of which treatment will be explained in the next section.

5.5 Identification and treatment of errors

In “days_in_waiting_list”, we had found a set of three outliers which had been more than 300 days on the waiting list. This value could be considered as an extreme outlier, but a customer staying on a waiting list for a hotel booking for almost a full year is not logical: if that was the case, they would book another reservation on another place. Consequently, we deemed them as errors.

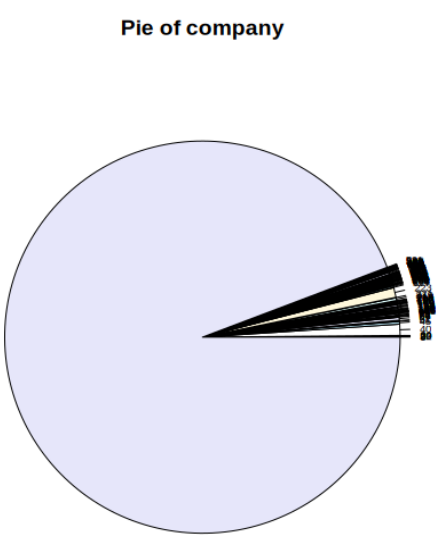
Given this situation, we considered that in order to treat the error appropriately, but keep the important information, we couldn’t just remove the affected rows from our database. Considering that we only had 5000, our results would be heavily skewed if we did that, so we concluded that it was better to impute values via Knn method to those that were missing.

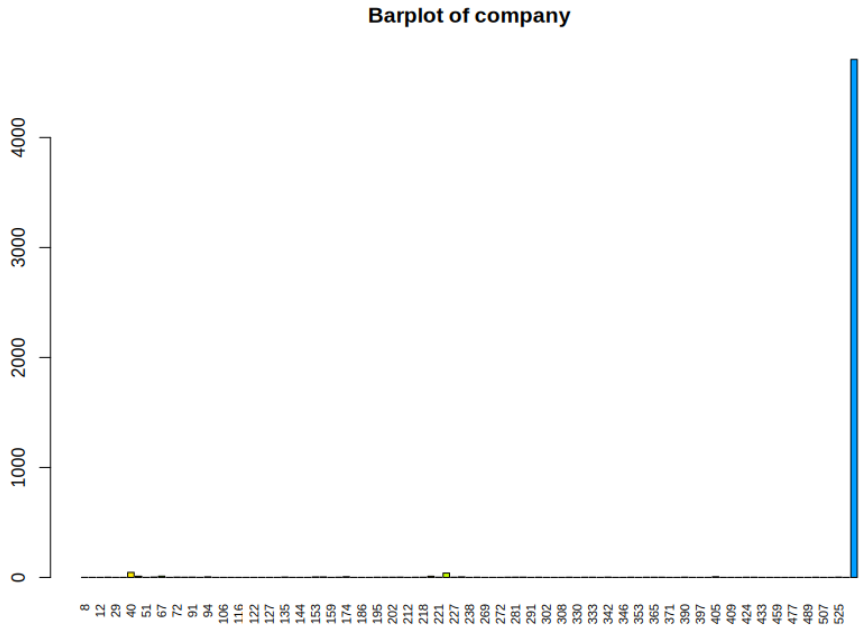
5.6 Data transformation

We have not needed to transform any data during the preprocessing of our dataset, but we have replaced the missing values of instances which presented errors. On the other hand, we have abbreviated the names of our variables in order to facilitate data handling and PCA.

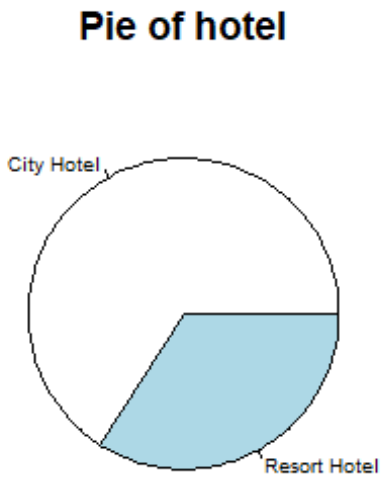
6. Statistical Descriptive Analysis

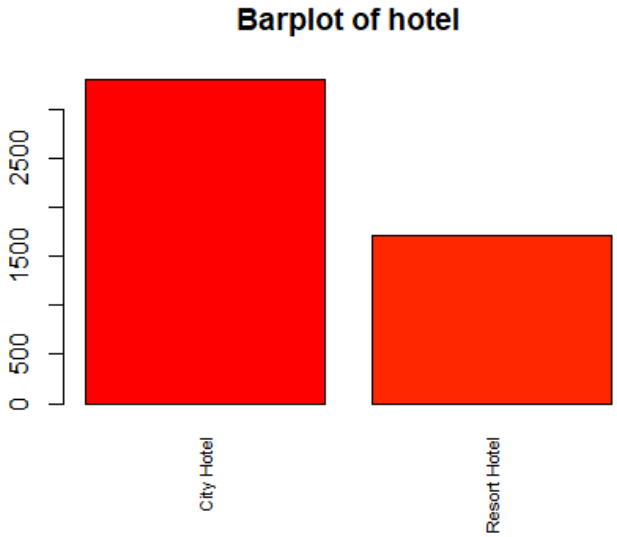
6.1 Univariate descriptive statistics





These graphs show the **high degree of missing values** that we talked about in our preprocessing steps: this variable has 101 modalities, while **4710 out of the 5000** (94.2% NA's) of the instances remain as **missing**. This can be seen in the pie char as the **light blue portion**, and the blue column in the histogram where the **values are represented as "9999…"**.

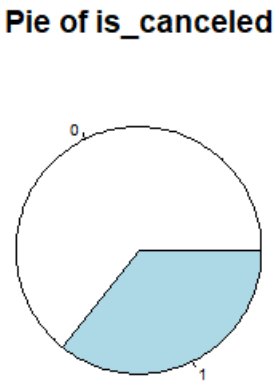


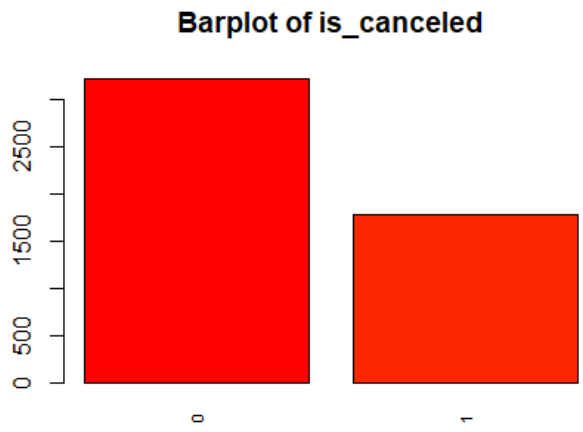


City Hotel	Resort Hotel
3297	1703
65.94%	34.06%

In these graphs, we can see that there are two types of hotels in this dataset: city hotels and resort hotels. From those, only about **one third are resort hotels**, while the other **two thirds are city hotels**. We can also point out that the number of instances of resort hotels is **half the amount of city hotels** in our dataset.

This could indicate that most of the **people prefer to book hotels inside the city than a resort**, or that **there are more urban hotels** than resort hotels in general.

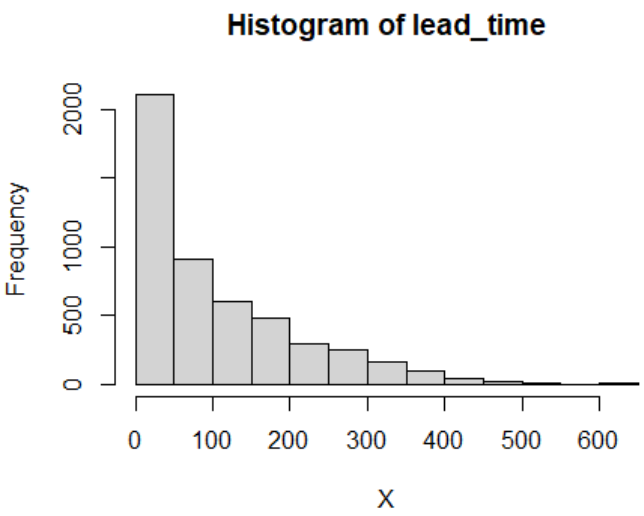


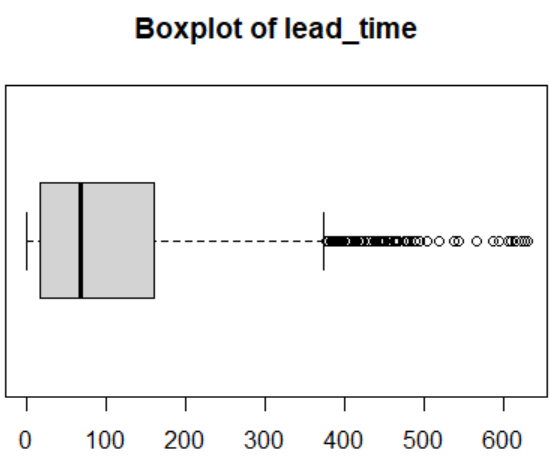


0 (False)	**1 (True)**
3218	1782
64.36%	35.64%

In these plots, we can see a similar result to the previous ones, where **almost 2/3 of the bookings were not cancelled**, and **1/3 were**. There are almost **twice as many non-cancelled** reservations compared to those cancelled.

This makes sense, since customers might be quite certain about their decision, as it involves prior organization in addition to the impact it has on their entire schedule during, and even around, the time of their stay. Therefore, these results may mean that more people who decided to book a hotel room were **confident and definitive** in their decision, while others **had some kind of inconvenience** that made them cancel their reservation.





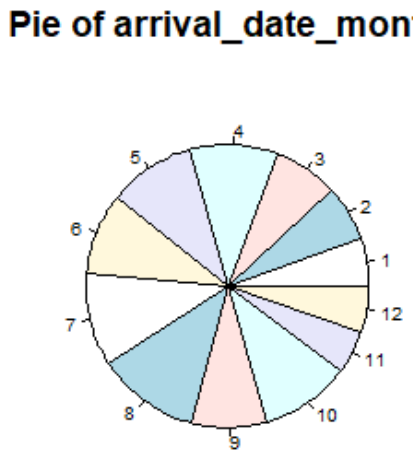
Min.	1st Qu.	Median	Mean	3rd Qu.	Max	sd	vc
0	18	69	105.3	160	629	109.14	1.04

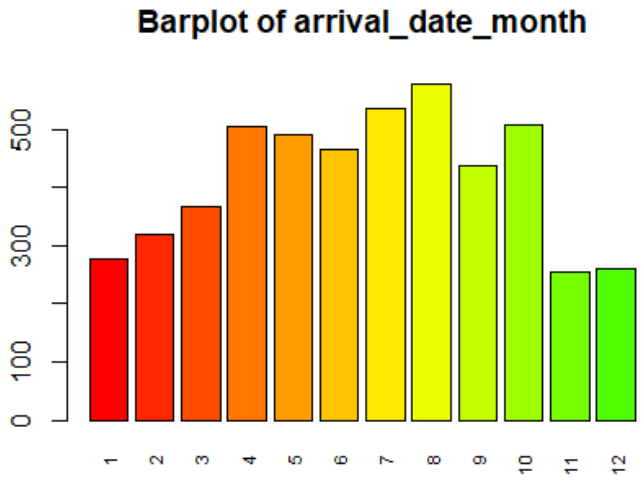
These plots and summary show us the information about the time elapsed between the customers' booking until their arrival.

If we analyze the summary, we can see that on **average, the arrival time is about 105 days**, while a **quarter** of bookings were made with a lead time of **18 days or less**, which means that they booked their stay relatively close to the arrival date. Half of the bookings had a lead time of **69 days or less**, and the other half more, so this could be considered a **typical value**. Lastly, **75% of the bookings** were made **160 days or less**, before their arrival.

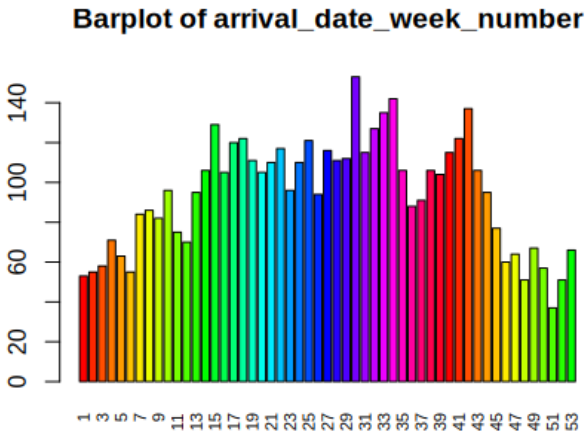
If we look at the **histogram**, we can see that many customers book their rooms approximately from **0 to 50 days**, and in the **box plot**, shows us that there are many outliers which book with a **lead time of more than a year**, which **skew the mean** towards that value of 105.

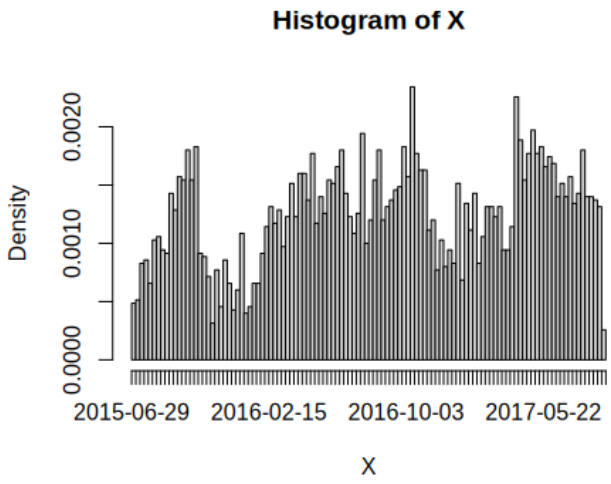
Overall, the results indicate that more customers **tend to book closer to their arrival date**, but a **significant minority plan far ahead**.





In both of these plots, we can observe that the months in which **more clients** arrive to the hotel are **August and June**, as **summer** is the season with the **highest demand** in hotels (holiday season). Furthermore, **April and May** also present **high results**, while the months from **November to February** have the **least** guests staying. These results seem to be **correlated with the weather**, since the warmest months with a **more pleasant climate** have the **most customers**, while the **coldest** months have **less**.

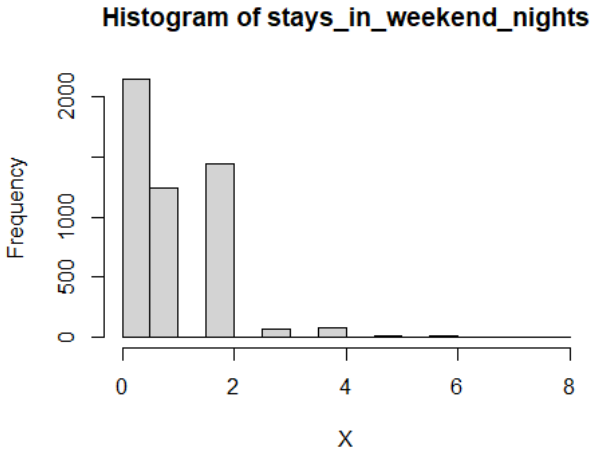


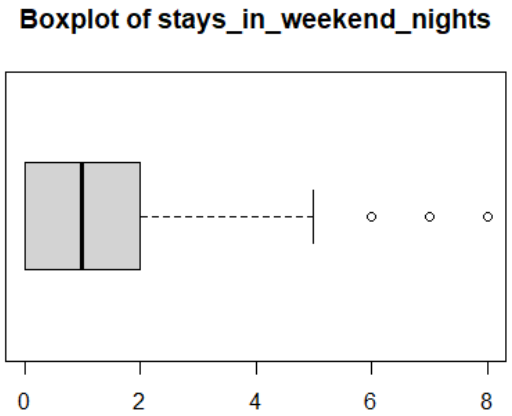


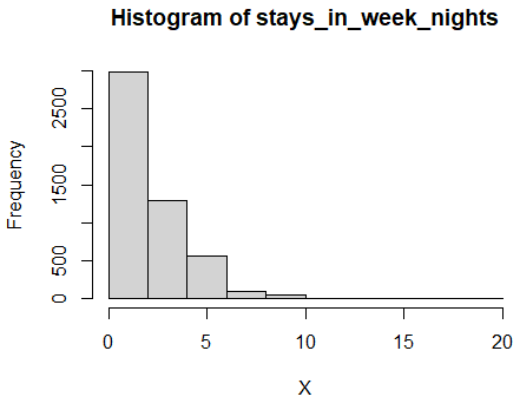
These two histograms are about the variable “arrival_date_week_number” and “arrival_date”. The first histogram represents the week number in the year (with 1 being the first week), giving us information

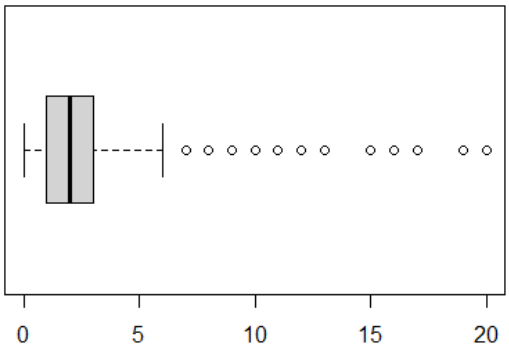
about the customers' time of arrival. The second histogram represents the exact date, being year, month and day of arrival of the given information.

By taking a look at the histograms, we can see that both follow a very similar pattern, in which most of the arrivals take place in the middle of the year. This corresponds to vocational season, and may also be correlated with the climate experienced in that period of time.









Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	sd	vc
stays_in_ week	0	0	1	0.9478	2	8	0.9966	1.0515

end_nights								
stays_in_week_nights	0	1	2	2.522	3	20	1.8767	0.7442

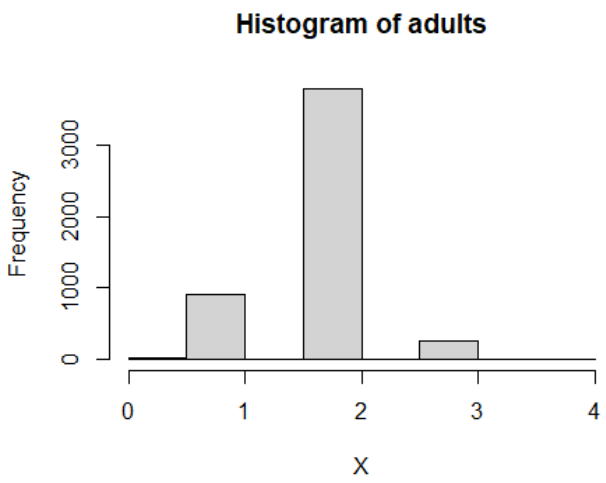
The summary of **"stays_in_weekend_nights"** shows us that, on **average, guests stayed 0.95** weekend nights, which is close to the **median** (1 night), indicating a fairly **good typical pattern**. In addition, it can be seen that at least **a quarter of them** did not stay **any weekends** at all, while **three quarters** of the guests stayed **2 or fewer** weekend nights. Overall, this results show that most of the people booked a short stay, being shorter than a full weekend in the hotels.

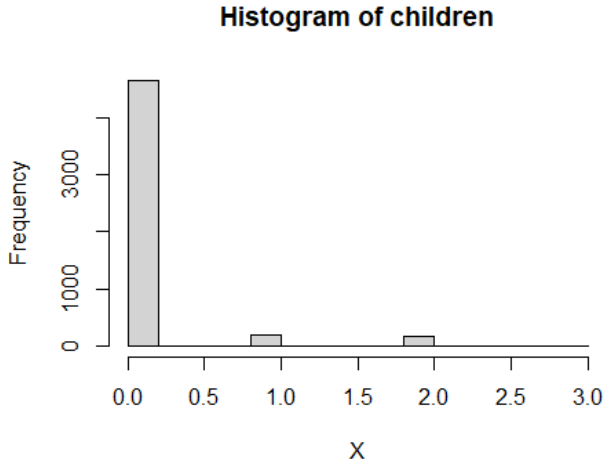
If we take a look at the histogram of "**stays_in_weekend_nights**", we can see that the **biggest group of people did not book** their stay on weekends, but a **significant number of people booked for 1 or 2** nights, which skewed the average towards 1 night.

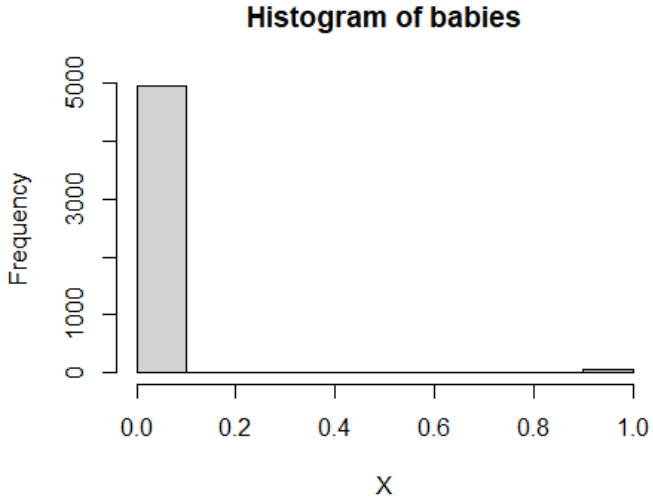
On the other hand, when considering "**stays_in_week_nights**", we can see that the **average is 2.52**, and the **median is 2 nights**, with most bookings falling **between 1 and 3 nights**.

Looking at the box plots, it can be noticed that the vast majority of people stay **up to 4–5 weekend nights**, and **up to 5–6 weekdays**.

In conclusion, people tend to stay a few weekdays and just a weekend night, which makes it less than a week. However, there are some people that stayed a larger amount of weekdays, but not all of them necessarily made it to many more full weekends.

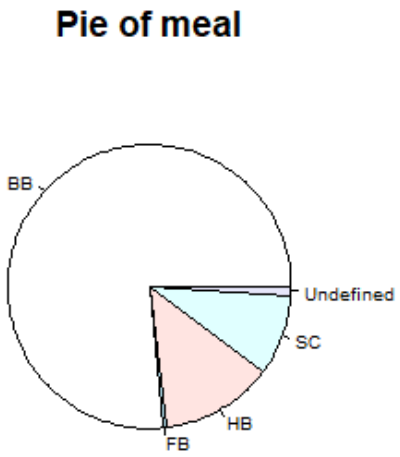






Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	sd	vc
adults	0.	1.	2.	2.522	3.	20	1.8766	0.7441
children	0.	2.	2.	1.862	2.	4.	0.4837	0.2597
babies	0.	0.	0.	0.1024	0.	3.	0.3979	3.886

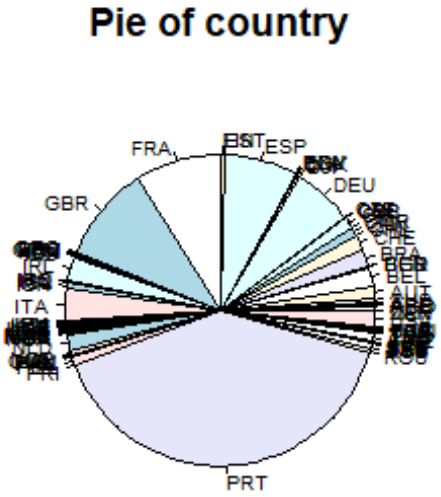
At a booking hotel reservation, we consider properly to analyze these 3 variables together, adults, children and babies. We can observe that most of the reservations are made by 1–3 adults, and we have a maximum value of 20 adults that suggests an extreme outlier that could mean a large group booking. As the same way for children, most of the reservations have 2 children. Children have low standard deviation a variation, and that indicates that this number of children is consistent in the reservations. Finally, for babies, we clearly see that most of the reservations have 0 babies. The mean of 0 indicates that making a reservation with babies is very rare.
So, this confirms that customers tend to travel alone or as a couple, with few families present.



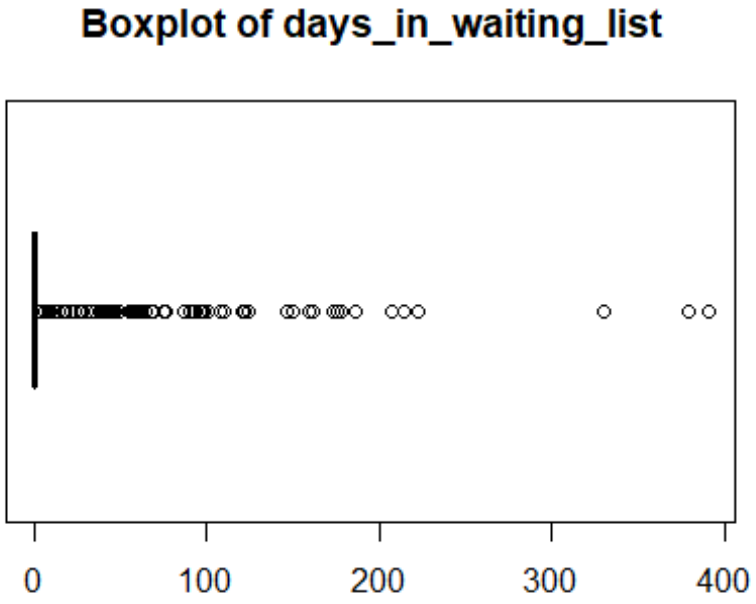
Modalities	Frequency	Proportions
BB	3822	0.7644
FB	28	0.0056
HB	637	0.1274
SC	464	0.0928
Undefined	49	0.0098

Analyzing the previous pie of the variable meal, we can conclude that the most wanted meal is BB, Bed & Breakfast with 76.44%. The second most wanted meal is HB, Half Board with 12.74%. The third one is SC, that is without a meal package with 9.28%. Then it comes a series of reservations with an undefined meal package, they are missing values. And finally there is a 0.56% that choose FB, Full Board.

The preference for basic plans suggests that customers prioritise cheap rates or flexibility to eat outside the hotel.



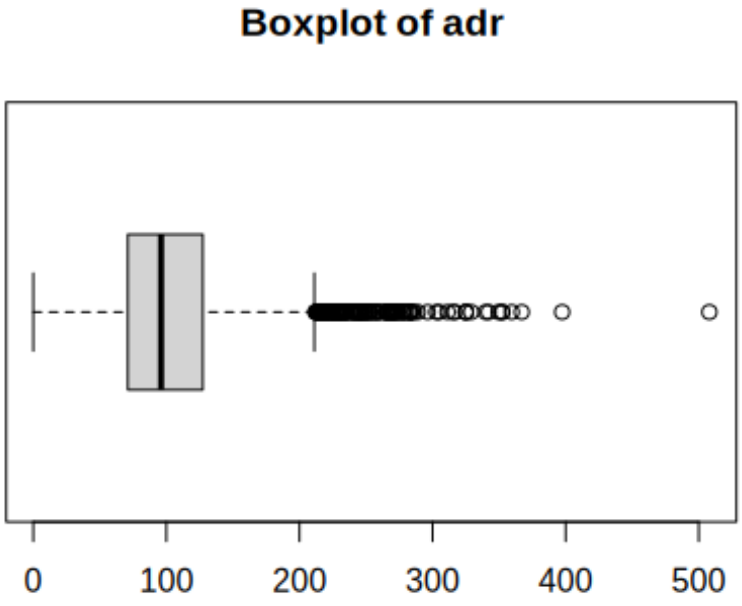
In this pie chart of country, there are 89 modalities. To facilitate the table analysis clarity we are not going to attach the frequency table but to mention the relevant information, the most useful one. The most frequent country is PRT, Portugal, being a 39.9% of the total. Second one is GBR, United Kingdom, with a 10%. Then it goes FRA, France with a 8.9%. The fourth one is DEU, Germany with the 6.06%. The others 85 countries represent the other 35.14%



Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	sd	vc

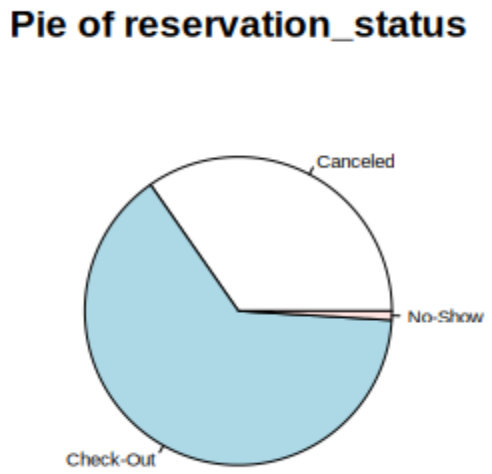
days_in_waiting_list	0.0	0.0	0.0	2.71	0.0	391.0	19.29	7.111

While analyzing the variable days_in_waiting_list, we can clearly see that the range of the majority of people spend between 0 and 3 days in the waiting list, having some outliers that spend more than 100 days (3 months approx being the extreme case of 391 days waiting. Although, this last case could bean error when collecting data or could be a rare case from a luxury hotel or hotel with big demand for rooms.



Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max	sd	vc
addr	0.0	71.0	96.0	103.3	127.3	508.0	48.831	0.47265

For this variable we see that the median is close to 100. As can be seen in the box plot, the dots indicate the many hotels that have an average rate higher than 200, and we can see a particular case of outlier (maximum value) of 508. This may be due to the price of the hotel or the number of people who have booked it, giving a high daily rate.



Modalities	Frequency	Proportions

Canceled	1735	0.3470
Check-Out	3218	0.6436
No-show	47	0.0094

We conclude with this pie, where we see that most of the reservations made are confirmed [Check-Out] having an 64%. However, it should be noted that it also has a considerable percentage of bookings cancelled by the customer, compared to the negligible part that do not show up at check-out (no show). As we will see below, this variable is related to the variable lead_time (lt).

6.2 Bivariate descriptive statistics

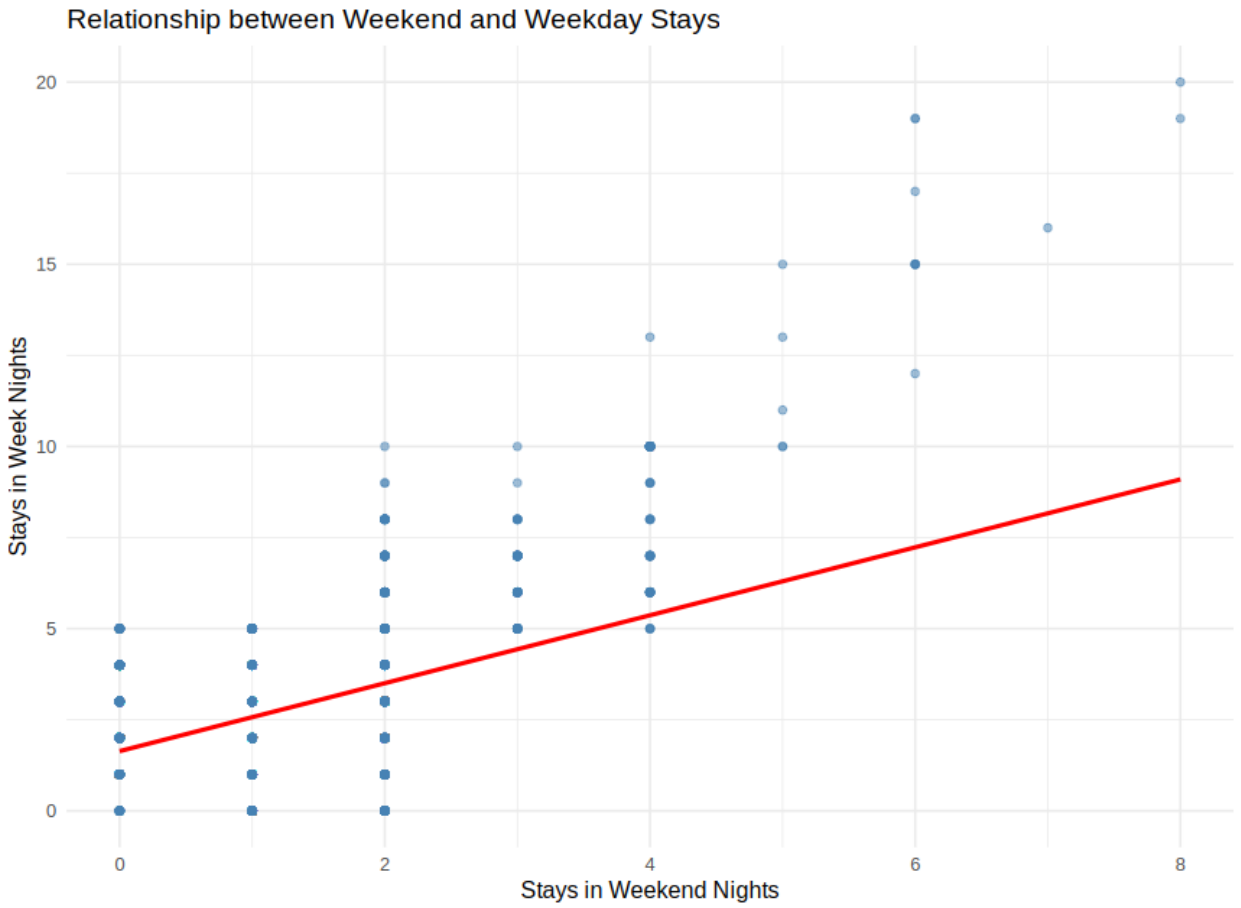
In the first place, we want to see the correlation between numerical variables in order to determine which variables need to be analyzed bivariately:

	lead_time	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	days_in_waiting_list	adr
lead_time	1	0.08	0.18	0.11	-0.04	-0.01	0.18	-0.09
stays_in_weekend_nights	0.08	1	0.5	0.12	0.04	0.01	-0.06	0.05
stays_in_week_nights	0.18	0.5	1	0.12	0.05	0.03	0	0.07
adults	0.11	0.12	0.12	1	0.03	0.02	-0.02	0.32
children	-0.04	0.04	0.05	0.03	1	0.03	-0.04	0.35
babies	-0.01	0.01	0.03	0.02	0.03	1	-0.01	0.01
days_in_wai	0.18	-0.06	0	-0.02	-0.04	-0.01	1	-0.04

ting_list								
adr	-0.09	0.05	0.07	0.32	0.35	0.01	-0.04	1

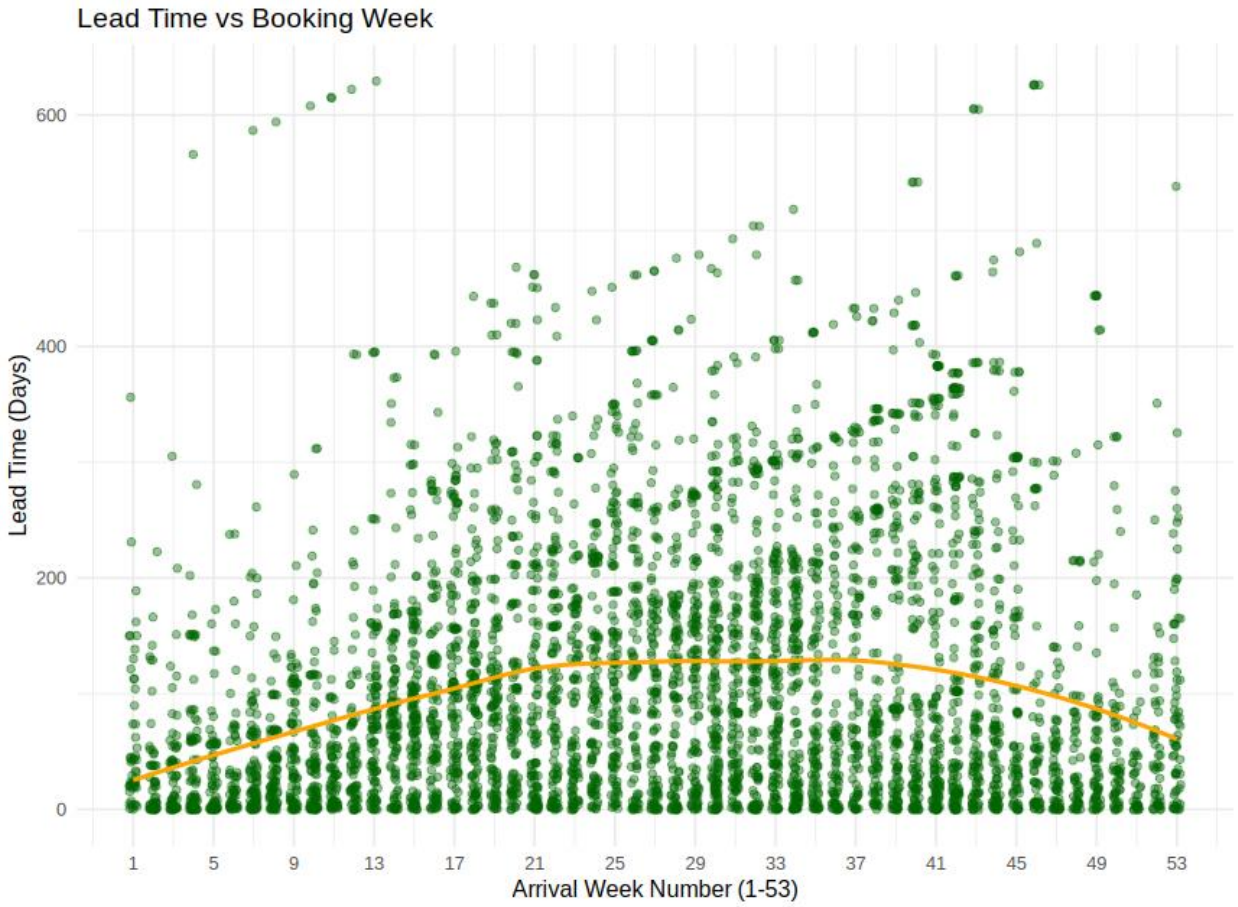
The resulting table shows us that the correlation between most of the columns is not too significant, except the highest one, which are **“stays_in_weekend_nights”** and **“stays_in_week_nights”**.

In order to visualize the correlation between those two variables, we have generated this scatter plot:



By looking at the plot, we can see a positive relationship between those variables, suggested by the positive slope of the data. In other words, as the number of weekday night increase, the number of weekend nights does as well. This implies that guests who book more weekend nights also often extend their stays into the weekdays. Moreover, we can see that for every full weekend (two weekend nights), guests stay approximately four weekdays, which is close to the expected full week. However, this tendency is not preserved in the right sector of the plot. This can happen because the guests that tend to stay more nights tend to book for vacation, which implies having a more flexible schedule where there is no difference between weekdays and weekends.

Furthermore, there are two other variables that might have a remarkable correlation, but are not shown in the numerical correlation matrix, since one of them is categorical. Those are **“lead_time” and “arrival_date_week_number”**, and represent the time elapsed between the booking and the arrival, and the week number of the year when the guest arrived. Consequently, we repeated the visual analysis by generating their corresponding scatter plot:



Firstly, we can notice a large amount of points at every arrival week. Those show that there is high variability on how far the people plan ahead their arrival. We can also see that the orange line, which represents the trend, takes the form of a positive arch, starting low, peaking at the middle, and ending low. This means that in early and late weeks of the year, on average, people book close to their arrival, while in the middle part of the year, people book further in advance. This may because people are likely to book with prevision for their vacation, and the high demand experienced during those weeks. Of which, both things may cause an increase of the price, the closer the booking is done to those dates (due to holiday season and good climate). On the other hand, the rest of the year, people tend to book closer to their arrival for the opposite reasons.

6.3 Overall descriptive conclusion

Regarding our results, both univariate and bivariate analysis show us reasonable outcomes as for what is expected from this dataset and the context behind it. This leads us to think that in the following steps, where we will apply different techniques to categorize and further analyze our data, we will be able to extract more specific and useful information for the purposes of our study.

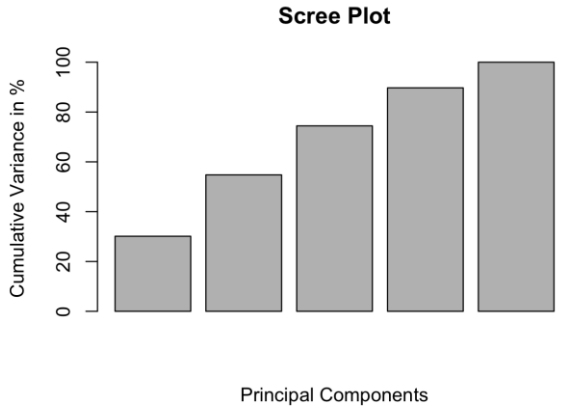
7. PCA Analysis for Numerical Variables

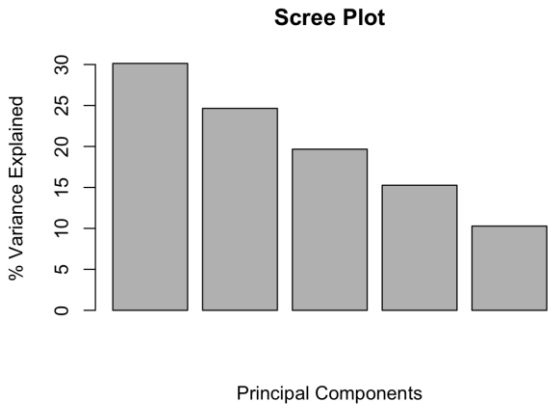
7.1 Factorial Maps

Factorial maps are visual representations of data that we have. They display the percentage of quantity across different variables in the dataset. If we want to reduce the amount of data, we must apply the PCA.

Data with a quantity around 75 - 80% is enough, so in order to reduce the number of variables and increment the information that those new variables have, we have decided to merge “children”, “adults” and “babies” into a new variable “people”, and “stays_in_weekend_nights”, “stays_in_week_nights” into “stays”. Therefore, we are now using only the first three largest variables. Later we will create all combinations of these variables.

In the first scatter plot, we can see percental variance of PCx in given dataset. Every column is one PC. The second graph shows cumulative variance in percentages of these PCs. We can see that first 3 columns are approximately 75% therefore we will use only them.





In the following table, we can seethe result of PCA. Each number in the table represents how much a specific original variable contributes to a given principal component (PC).

- Large positive values (close to 1) → The variable strongly contributes to that principal component in the same direction.
- Large negative values (close to -1) → The variable strongly contributes to that principal component in the opposite direction.
- Small values (close to 0) → The variable has little influence on that principal component.

PC1 (First Principal Component):

- adr (-0.8145) and people (-0.8412) have large negative loadings
  - These variables are strongly negatively correlated with PC1.
  - This means that as PC1 increases, adr and people tend to decrease.
- d_wl (0.1361) and lt (-0.0105) have small values
  - These variables contribute little to PC1.

PC2 (Second Principal Component):

- lt (0.8105) and d_wl (0.5353) have large positive values
  - They strongly contribute to PC2 in a positive direction.
- adr (-0.1695) and people (0.0336) have small values
  - They contribute weakly to PC2.

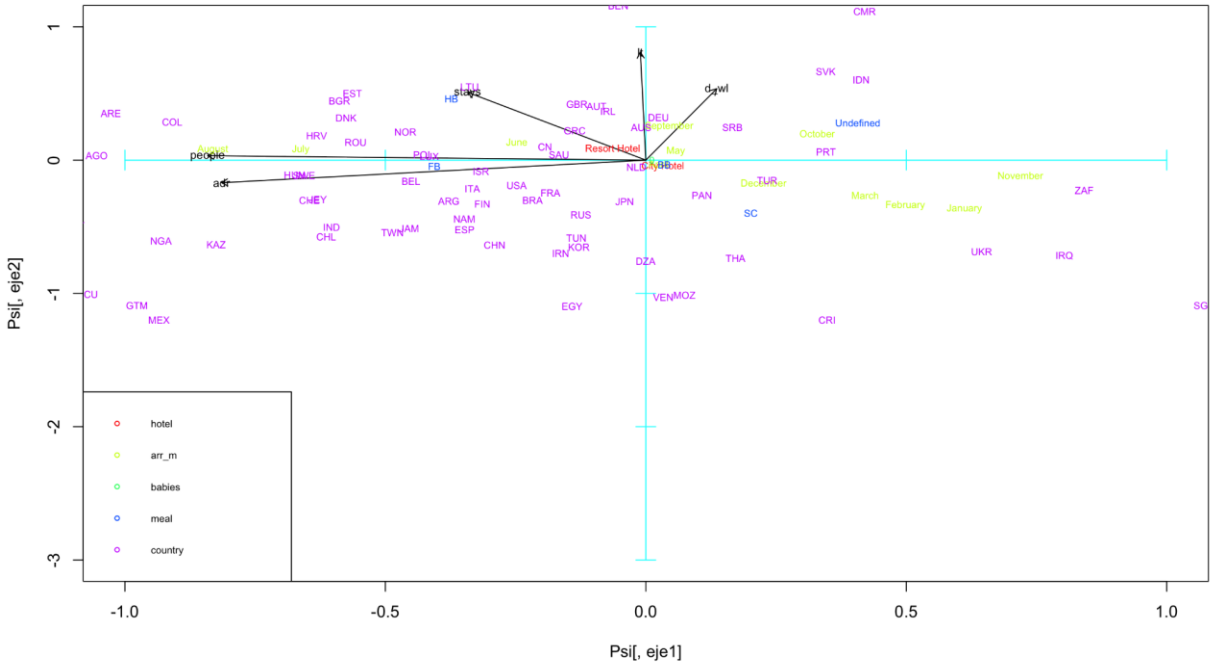
PC3 (Third Principal Component):

- d_wl (-0.7361) has a strong negative contribution.
- stays (0.5992) has a strong positive contribution.
- Other variables have smaller contributions.

	PC1	PC2	PC3
lt	-0.0105120	0.81046818	0.06147419
d_wl	0.1361327	0.53533360	-0.73610089
adr	-0.8145321	-0.16950190	-0.25467258
people	-0.8412287	0.03366039	-0.11700768
stays	-0.3420930	0.50894186	0.59929775

7.2 Interpretation of The Graphs

PC1 and PC2

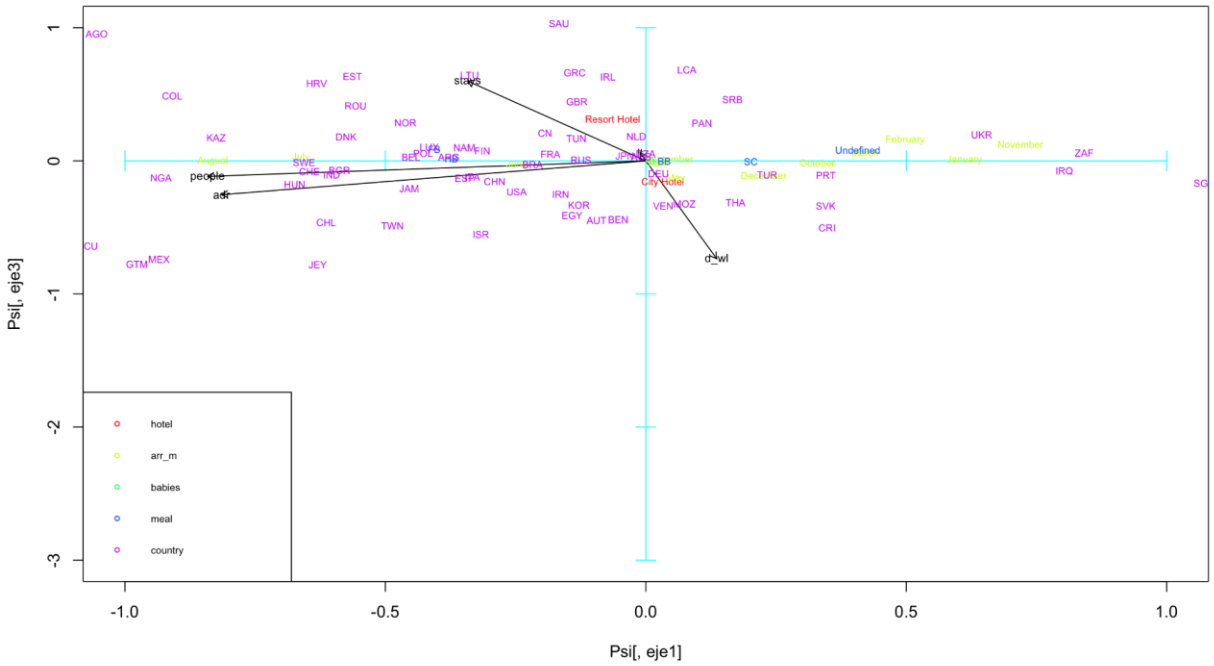


Moving left to right on the plot goes from summer- oriented patterns (often with more people or different ADRs - Average Daily Rates) toward winter months (November, December) and possibly higher wait- list days or different booking behaviors.

Moving bottom to top shifts from shorter, simpler stays (lower “stays,” “dwl” - Days in Waiting List or lower correlation with ResortHotel) to higher values on those same metrics (longer stays, longer waiting lists, or more Resort- type bookings).

Traveler or booking in the top- right quadrant is more likely a Resort Hotel booking with a longer stay, often in months on the right side of the calendar (late summer to winter), and probably with a higher daily rate or more days on the waitlist. Conversely, bookings in the lower- left quadrant (e.g. certain countries in Latin America plus midyear months) suggest shorter stays, fewer waitlist days, and possibly lower ADR.

PC1 and PC3



Horizontal axis (PC1) is dominated on the left by “adr” - Average Daily Rate and “people,” while on the right it is more closely aligned with “CityHotel,” “ResortHotel,” and (somewhat) “dwl” - Days in Waiting List. This suggests that PC1 is capturing a contrast between booking cost/size (left) and type of hotel (right).

Meanwhile, the vertical axis (PC3) splits upward toward “stays” and downward toward “dwl.” This implies that PC3 separates longer stays at the top from longer wait- list periods at the bottom. So, as you move up on PC3, the length of stay increases; as you move down, the days waiting on the list grow.

“stays,” “ResortHotel” both pull toward the top or upper- right, indicating that longer stays are linked more strongly to resort- type bookings.

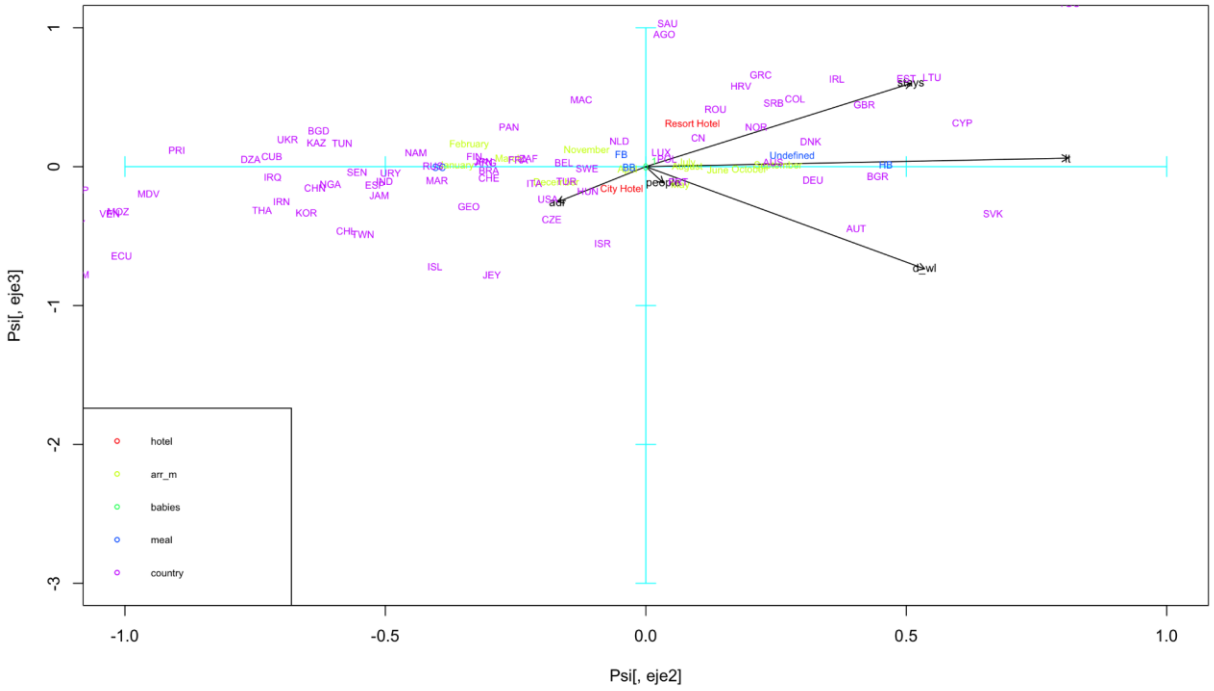
“dwl” points down and to the right, suggesting that bookings that spend more days on the waitlist also show up on the positive side of PC1 (i.e., more associated with “CityHotel” in this PC dimension).

Months (in yellow) are again strung out primarily along the horizontal direction, from August (farther left, near higher “adr”/“people”) across to December/Undefined (farther right, closer to “CityHotel” and higher “dwl”).

Countries (in purple) scatter throughout, reflecting how travelers from each place align with these two principal components. Some cluster near the top (longer stays), others near the bottom (more wait- list days), and still others split left vs. right based on ADR vs. hotel type.

Overall, moving left- to- right goes from higher ADR / more people (negative PC1) to city/resort hotel bookings (positive PC1), and moving bottom- to- top goes from longer waitlists (negative PC3) to longer stays (positive PC3).

PC2 and PC3



Horizontal axis (PC2) is pulled strongly to the right by “people,” “stays,” “dwl” - Days in Waiting List, and “ResortHotel,” whereas relatively few variables sit far out on the negative (left) side. This suggests that moving left to right on PC2 distinguishes smaller or simpler bookings (fewer people, fewer days waiting, shorter stays) on the left from larger or more complex bookings (more people, longer stays, more wait- list days, and a stronger ResortHotel association) on the right.

Meanwhile, the vertical axis (PC3) is driven upwards by “stays” (length of stay) and downwards by “dwl.” That implies a contrast between longer- stay bookings (high PC3) and longer- wait- list bookings (low PC3). So, at the top you see points that reflect extended stays, and toward the bottom you find those that spend more days waiting.

“stays” and “ResortHotel” cluster in the top- right quadrant, implying bookings that have longer stays and tend to be at resorts.

“dwl” angles down- right, indicating that higher wait- list durations still show up on the positive side of PC2 (i.e. more complex bookings) but come out negatively on PC3.

“people” is mostly horizontal, aligning strongly with PC2 but not much with PC3, suggesting group size is part of that “complex vs. simpler bookings” distinction.

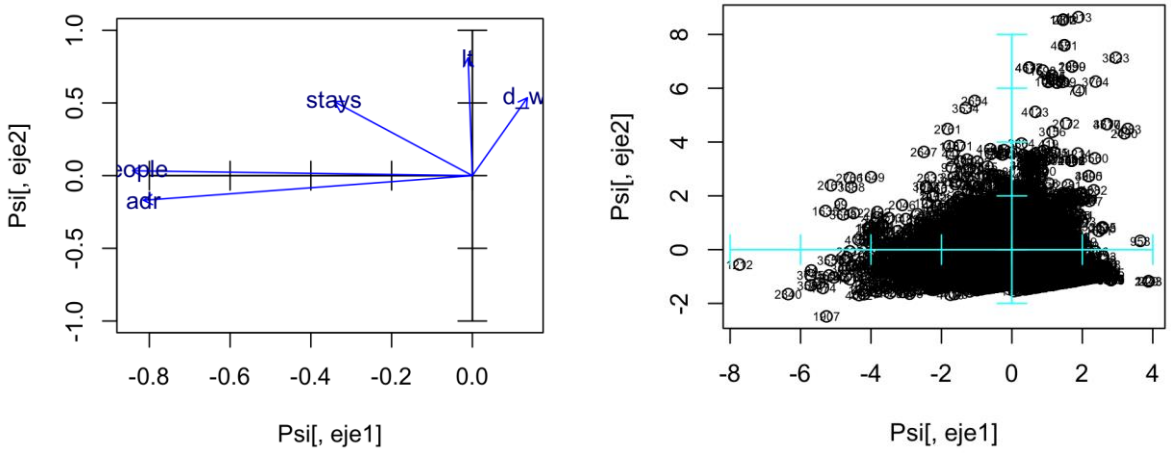
The months (in yellow) are scattered mostly near the center of PC2–PC3, implying their seasonality effect is less dominant here than in, say, PC1; and the countries (purple) spread across both axes according to which booking patterns they align with (short stays vs. long stays, fewer people vs. more people, etc.).

Left to Right (PC2): from simpler, smaller- party bookings to more people, longer stays, higher wait- lists, and a strong link to resort hotels.

Bottom to Top (PC3): from bookings characterized by more wait- list days (lower end) to those with longer stays (upper end).

7.3 Additional Comparisons

PC1 and PC2



Variables Plot (Blue Vectors)

- Each arrow represents a numerical variable—people, adr (Average Daily Rate), stays, lt (Lead Time), and d_wl (days in waiting list).

- Notice that “people” and “adr” lie mostly along the negative side of the PC1 axis, while “stays” fans out toward the upper- right quadrant and “d_wl” is angled to the right (positive PC2).
- This positioning implies that PC1 contrasts bookings that have higher “people”/“adr” (toward the negative end) vs. those that trend more toward additional variables like wait-list days or length of stay (moving positive on PC1). Meanwhile, PC2 distinguishes longer stays (higher on PC2) from shorter or simpler bookings (lower on PC2).

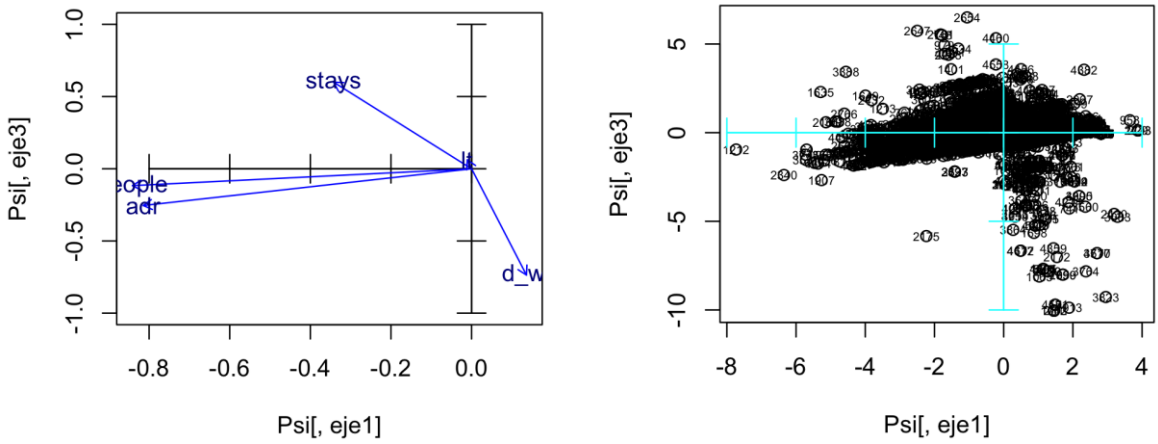
Individuals Plot (Black Points)

- Each dot is one booking (or individual observation), projected onto the same PC1–PC2 space.
- The cloud of points is centered near (0,0), with most observations sprawling out left (negative PC1) and upward (positive PC2). This suggests that many bookings lean toward slightly higher “people” or “adr” values on the left side, yet there’s also a strong upward trend reflecting moderately longer stays.
- A noticeable spread toward the right side (positive PC1) would indicate bookings correlated more with days on waiting list or possibly the “k” variable—though from this image, the bulk of points seems to remain on the negative or center- left portion of PC1.
- We can also spot a few outliers in the upper or far- right edges, which may represent unusually long stays or extended waiting periods.

To sum it up

- PC1 primarily captures the trade- off between higher “people”/“adr” on one end and stronger “d_wl”/lt/“stays” tendencies on the other.
- PC2 appears more driven by the length of stay dimension, separating shorter vs. longer bookings. In the individual plot, most points cluster around the center- left region (indicating moderate “people”/“adr” values and moderate stays), with a few outliers reaching up or right, hinting at more extreme booking patterns.

PC1 and PC3



Variables Plot (Blue Vectors)

- Horizontally (PC1), “people” and “adr” both extend toward the negative side, indicating bookings with more guests and higher average daily rates sit on the left. On the right side of PC1, there aren’t any strong arrows pulling in that direction, suggesting these variables don’t dominate positive PC1.
- Vertically (PC3), “stays” goes sharply up, whereas “dz_wl” (days on waiting list) points down. This shows that PC3 is mainly contrasting longer stays (positive PC3) vs. more waiting- list days (negative PC3). The variable “t” (possibly children or another factor) sits near the midpoint, having only a mild positive slope on PC3.
- From this, we can see that PC1 roughly separates larger- party/higher- ADR bookings to the left from other types of bookings to the right, while PC3 splits out whether a booking is characterized by long stays (up) or extended waiting periods (down).

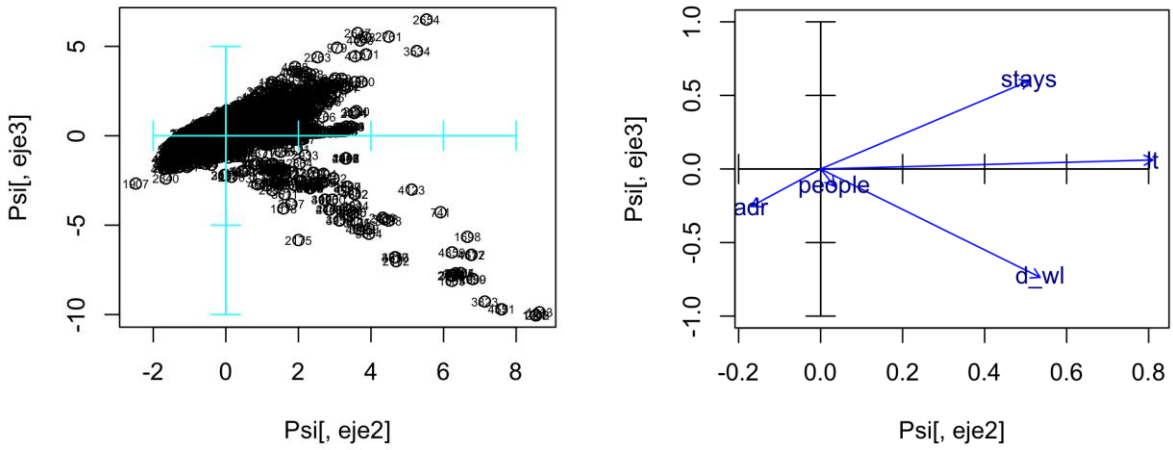
Individuals Plot (Black Points)

- Each point represents a single booking. The majority cluster between –6 and 0 on PC1 and –5 and 2 on PC3, suggesting most reservations lean toward moderate or slightly higher “people/adr” (since they’re on the negative side of PC1) and fairly balanced stays vs. waiting- list days.
- A visible elongation of the cloud from the lower- left to the upper- right indicates that many bookings have a small- to- moderate number of people, with occasional outliers on the extreme negative side (very high “people” or “adr”) or far up/down in PC3 (long stays or long wait lists).
- The very top outliers on PC3 correspond to bookings with especially long stays, whereas the lowest points likely reflect reservations with many days on the waiting list.

To sum it up

- PC1 vs. PC3 shows how guest/ADR characteristics (horizontal axis) interact with the trade- off between stay length and wait- list (vertical axis), with most bookings clustering in a moderate range and a few outliers revealing extreme values of either stay duration or waiting days.

PC2 and PC3



Variables Plot (Blue Vectors)

- On the horizontal axis (PC2), we see “stays,” “d⊥wl,” and “t” all extending to the right, while “adr” skews slightly to the left. “people” is near the origin, indicating it is less strongly associated with either positive or negative PC2.
- Vertically (PC3), “stays” goes up, whereas “d⊥wl” and “adr” tilt down. This suggests PC3 distinguishes longer stays (top) from longer wait lists / higher ADR (bottom), while PC2 differentiates lower-ADR bookings (left) from bookings with more days on waiting list, possibly larger party sizes, or additional variables (right).
- In short, moving from left to right (negative to positive PC2) captures a contrast between lower ADR vs. higher waiting- list and “t” values, and moving bottom to top (negative to positive PC3) highlights a difference between higher ADR / more wait- list days vs. longer stays.

Individuals Plot (Black Points)

- Each black point is an individual booking mapped onto the same PC2–PC3 space. The cloud centers around $(0, 0)$ but spreads to the right (positive PC2) and both up and down along PC3.
- A large portion of bookings cluster around negative- to- center PC2 (where ADR is not too high) and moderate PC3 values (moderate stays / moderate wait- list). However, we also see a noticeable extension toward the positive side of PC2, indicating some bookings are more associated with higher days on waiting list (or the “t” variable).
- Vertically, many points spread from about –5 to +5 on PC3. Those that go far up on PC3 likely correspond to longer stays, while those far down may reflect higher ADR or more wait- list days (in line with the arrows in the variables plot).

To sum it up

- Most bookings lean toward relatively moderate ADR and moderate stays, but there are outliers—some with very long waiting lists or extended stays—pushing far to the right or far up/down in the plot.
- The result is that PC2 vs. PC3 nicely teases apart distinctions like how long people stay vs. how long they might wait (or how high the daily rate is) in each reservation.

7.4 Conclusion

PC1 captures strong seasonal effects on bookings. Meanwhile, **PC2** separates small/simple vs. large/complex booking profiles (party size, total stays, wait- list days). Furthermore, **PC3** highlights whether a booking is driven by length of stay vs. days on the waiting list.

As a whole, these three principal components reveal distinct patterns in how seasonality, booking complexity, and the stay- length / wait- list tradeoff all shape the overall structure of the dataset.

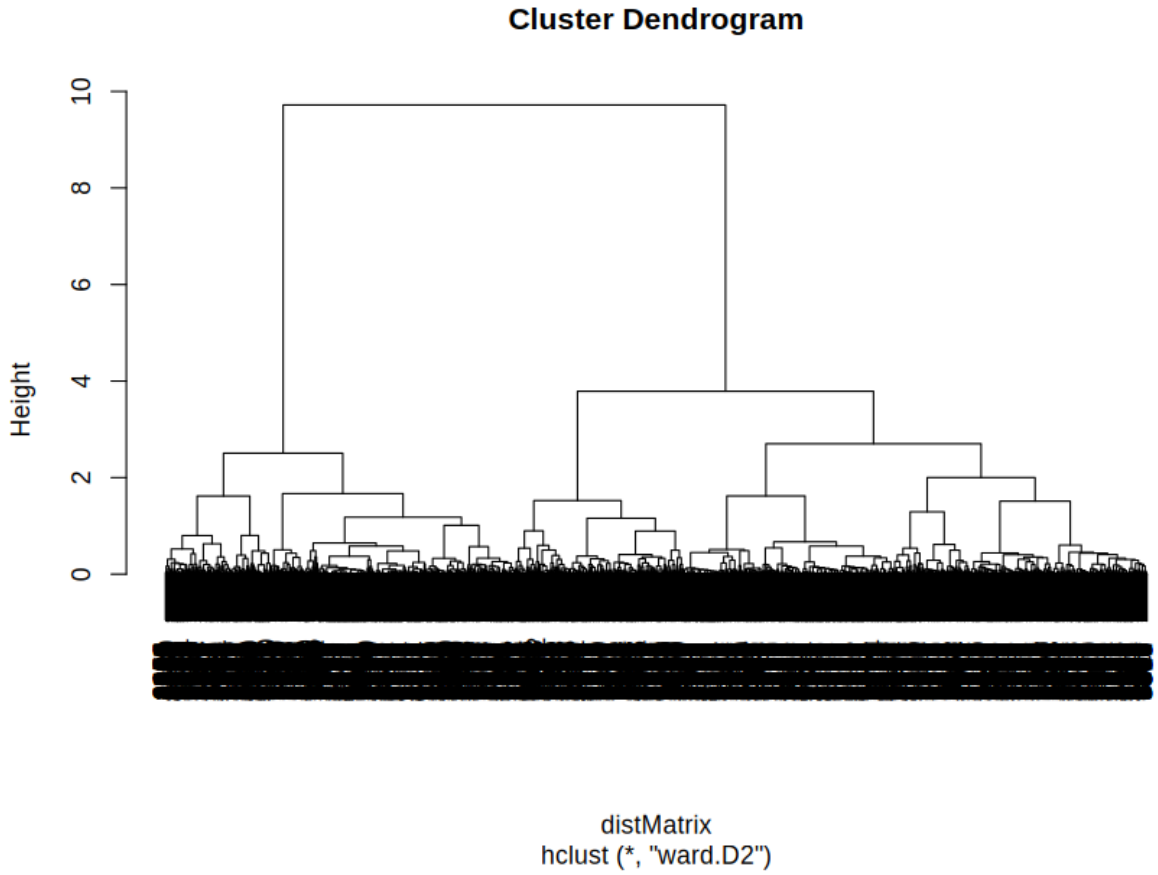
8. Hierarchical Clustering

In this section of the report, we are going to discuss the clustering of all our variables with all our selected variables, **except "arrival_date", "arrival_date_week_number", "**arrival_week_number**"**, since we **already have the variable "arrival_date_month"**, which also refers to the date in which customers arrived, and in order **not to overweight the month** in our clustering, we decided to discard one of those columns. The removed one ended up being "arrival_date" because "arrival_date_month" represents the **most appropriate granularity** for the purposes of this study. In summary, we decided to include the following variables:

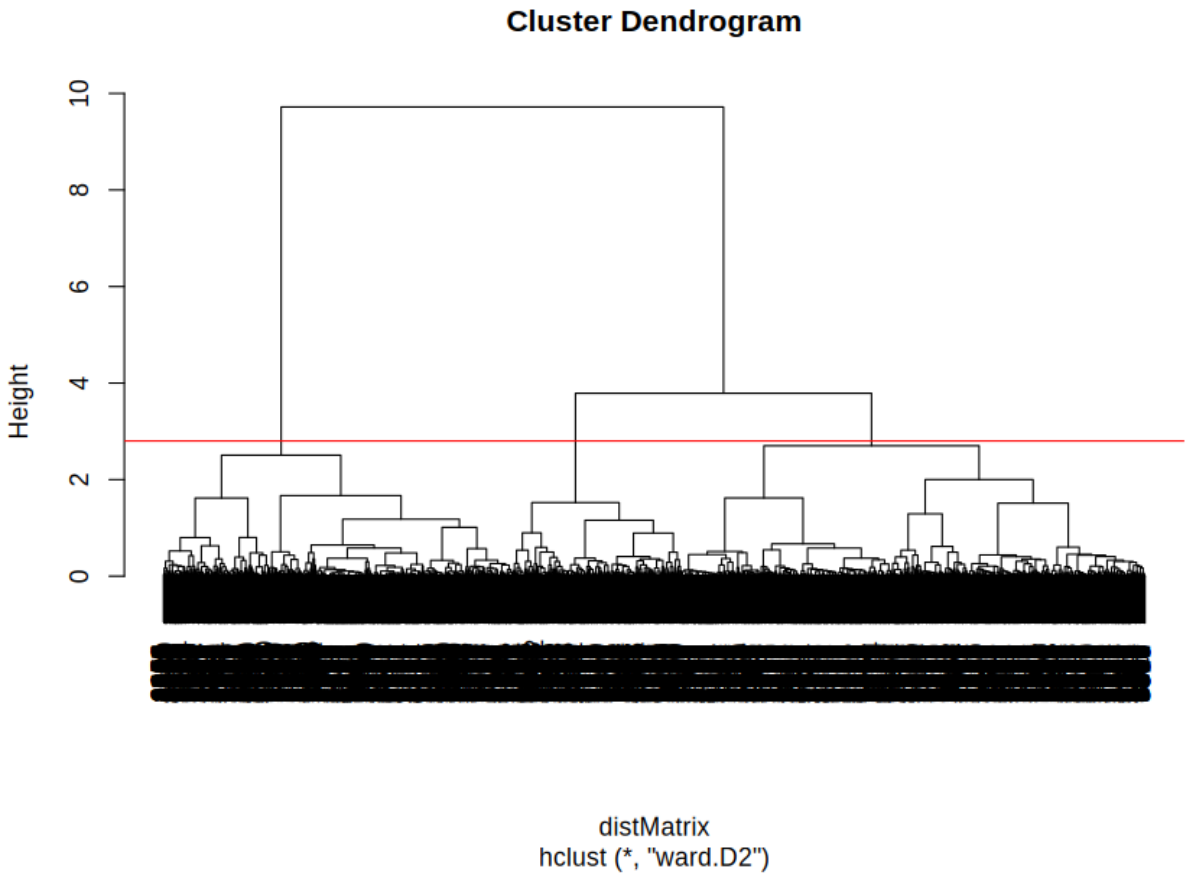
"hotel", "is_cancelled", "lead_time", "arrival_date_month", "stays_in_weekend_nights", "stays_in_week_nights", "adults", "children", "babies", "meal", "country", "days_in_waiting_list", "adr", and "res_s".

In order to do the analysis, we decided to employ **Ward's method**: a hierarchical clustering procedure, which uses **aggregation** between the groups, based on the **minimum intra-group induced variance** in the process. In addition, we used the square of **Gower's dissimilarity** as the **metric**, because it will allow us to work with both **numerical and categorical** variables.

After executing the previously specified algorithm, we obtained the following **dendogram**:



By looking at the resulting dendogram, we can see that its structure is heavily conditioned by the variable “is_cancelled”, which separates the clusters in two clearly defined groups. Furthermore, we can notice that the **longest height distance** is **between ~1.5 and ~3**, without considering the distance between ~3 and the highest level, which would leave us with a trivial number of clusters (just two). Therefore, we applied the cut on **height 1.5**, resulting in **3 clusters**, as shown in the following figure:



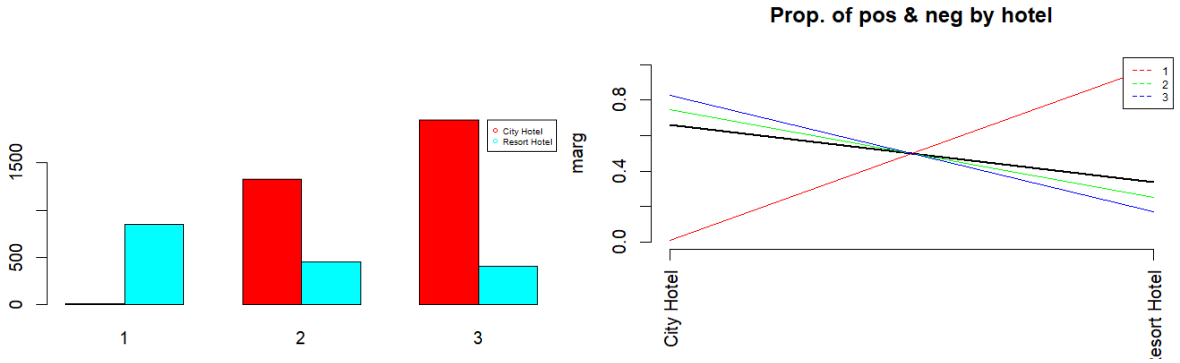
Finally, we obtained clusters with the following sizes:

Cluster	1	2	3
Size	857	1782	2361

9. Profiling of Clusters

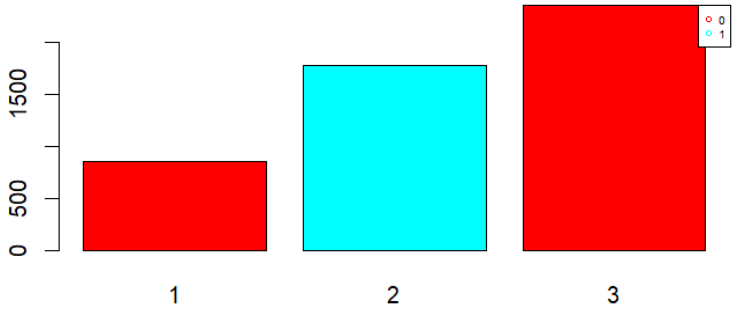
Based on the 3 obtained clusters, we can now do a deep analysis searching for patterns, tendencies or preferences that each group has, and then do a profiling interpretation for each one. We performed statistical tests like ANOVA, Kruskal-Wallis and Chi-square to compare variable distributions across clusters. Before concluding anything, it is important to determine which are those variables that are significant.

1. Hotel Type



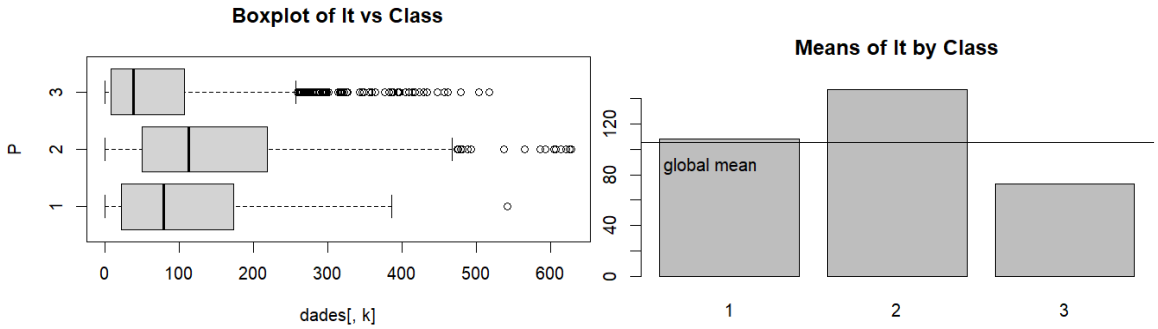
The variable “hotel” indicates whether a booking was made of a city hotel or else a resort. In the bar plots we can see that in cluster 1 we exclusively have reservations in resorts while in the other two clusters there are two types of bookings. But, city hotels are clearly the majority in cluster 2 and 3.

2. Is Canceled



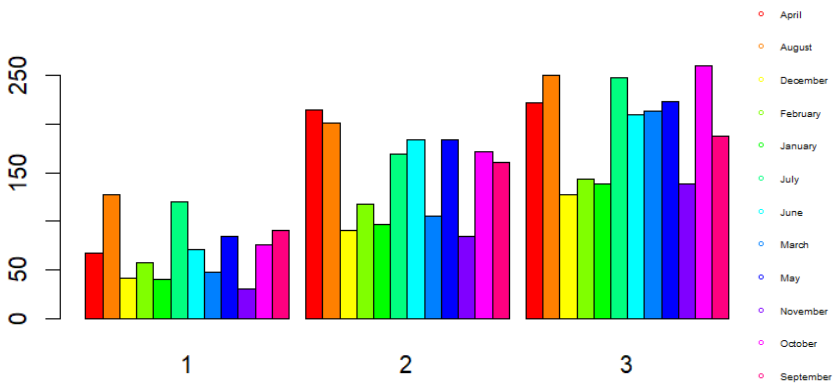
The variable “is_canceled” indicates whether a hotel booking was finally cancelled or not. In this plot it is visibly one-sided, Cluster 2 has all the bookings cancelled. Cluster 1 and 3 have those not cancelled.

3. Lead Time



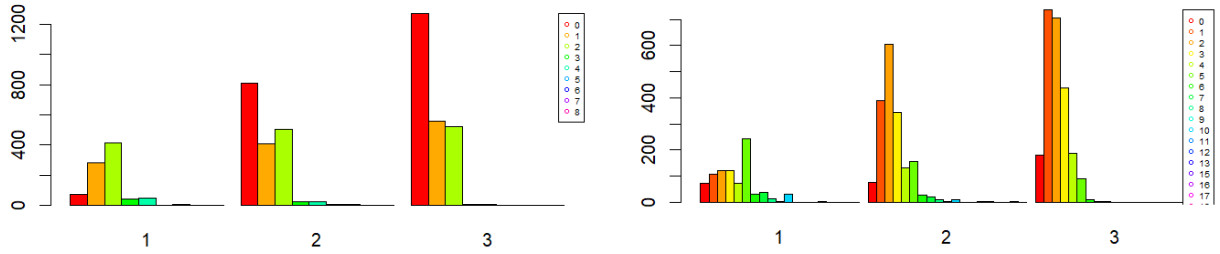
The variable “lead_time” indicates the period of advance notice. It refers to the amount of time since a reservation is made until the actual stay. In this plot, Cluster 1 is very distributed, Cluster 2 has the highest lead time and Cluster 3 has very short lead time.

4. Arrival Month



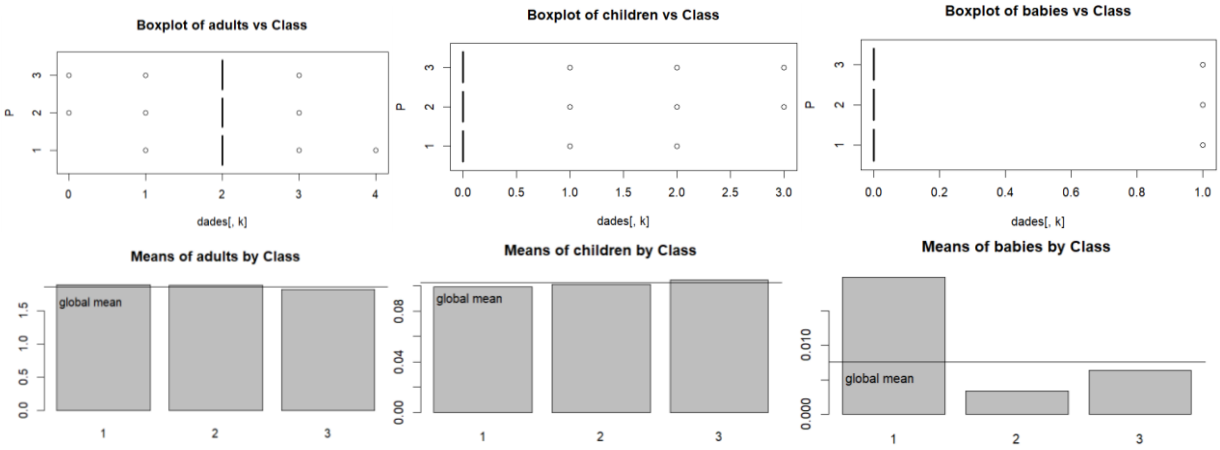
The variable “arrival_date_month” indicates the month of a booking staying. The analysis of this variable is crucial to set a seasonality profile. People in Cluster 1 are those who made fewer reservations during the period. In this class, there are 2 peaks in August and January that makes us relate them to vacation season. In Cluster 2 and 3 there are more reservations across the year.

5. Stays in weekend nights and stays in weekdays nights



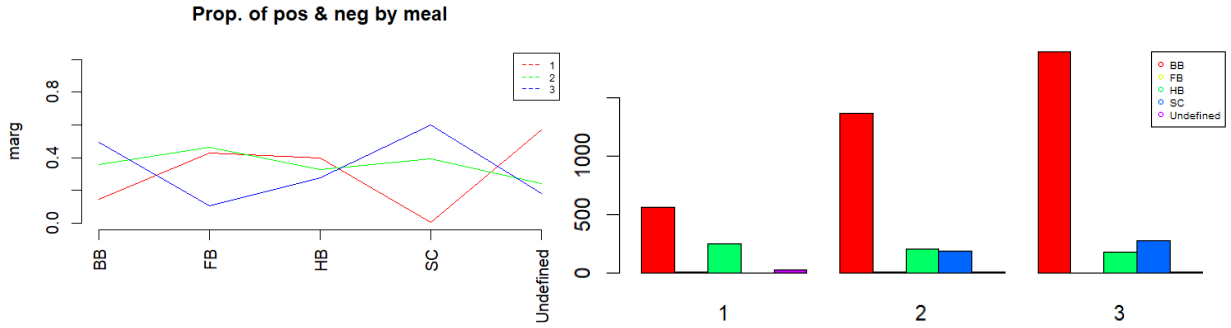
The variables “stays_in_weekend_nights” and “stays_in_weekday_nights” are two variables that indicates the number of days of the weekend or weekdays that each cluster is more likely to stay at a hotel. People in Cluster 1 are more likely to stay on weekend nights. Cluster 2 is more or less equally likely to stay on weekend nights than 2 or 3 weekdays. Cluster 3 is more likely to stay on 2 or 3 weekday nights than weekend.

6. Number of Adults, Children and Babies



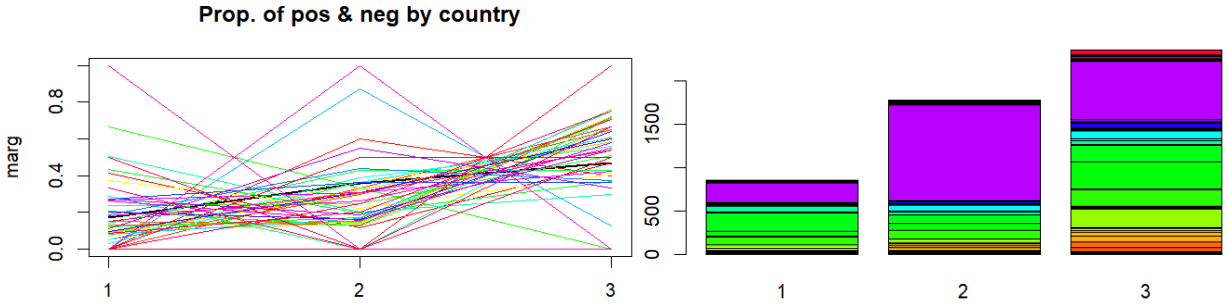
The variable “adults”, “children” and “babies” indicates how many adults, children and babies are on average in a reservation. For the variables adults and children, there is no significant special treat from each cluster. The one that leads us to a more profiling characteristic is the variable baby. People in cluster 1 are more likely to do a reservation with babies. Cluster 2 and 3 it is very rare and rare respectively for babies to be on a reservation.

7. Meal



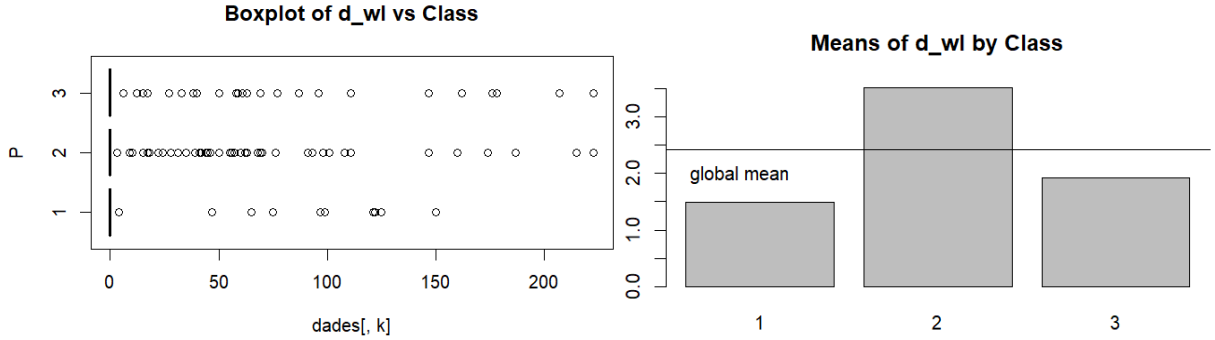
The variable “meal” indicates the type of food service people choose. In Cluster 1 there is the absence of SC, Self Catering. In Cluster 3 there is the absence of FB, Full Breakfast. In each cluster there is a common preference for BB, Bed&Breakfast.

8. Country



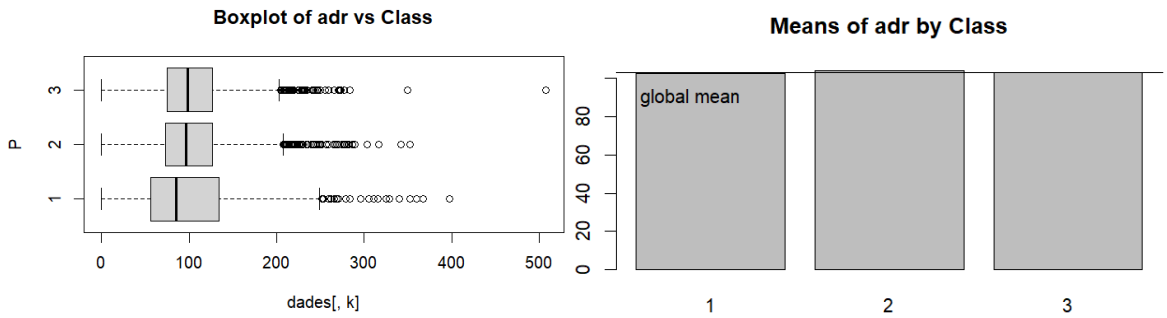
The variable “country” indicates the origin of clients. This variable has 89 modalities, but there is a majority country in general, that is Portugal (PRT). In Cluster 1 there is also the United Kingdom (GB). In Cluster 3 there are lots of European countries like France, Germany, Spain. In Cluster 2 as well, but in lower proportions.

9. Days in waiting list



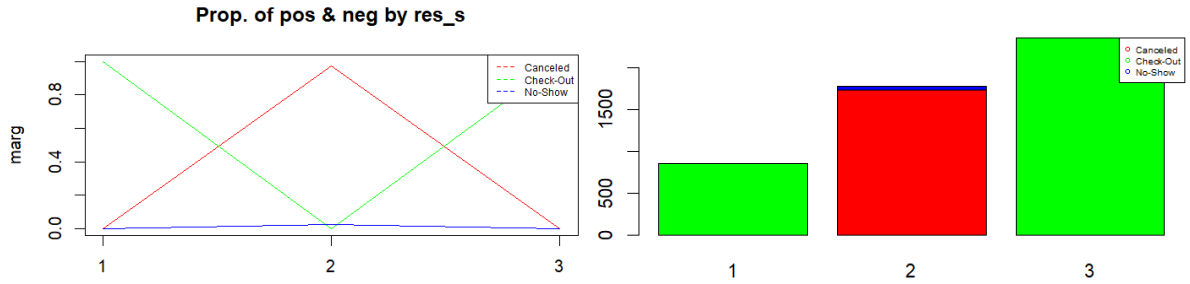
The variable “days_in_waiting_list” indicates the number of days the booking was in the waiting list before it was confirmed to the customer. Cluster 2 has the highest number of days in the waiting list. Cluster 1 and 2 have fewer days, less than the average.

10. Average Daily Rate



The variable “adr” indicates the amount of money spent on a day. This plot does not give any profiling feature significantly enough, as the range and the median of the three clusters is very similar, so the difference between them is not significant.

11. Reservation Status



The variable “reservation_status” indicates the final status of a booking. Cluster 2 is exclusively dedicated to canceled reservations and some little portion of not-showed. Cluster 1 and 2 are those that made a check-out.

9.1 Summary interpretation of clusters

In summary, our goal was to identify distinct groups with similar characteristics within the dataset. To determine the optimal number of clusters, we applied Gower's method, which resulted in three clusters.
We then performed hierarchical clustering using both numerical and categorical variables to extract meaningful insights. To assess whether the data was effectively grouped and to highlight key differences between clusters, we conducted a detailed profiling analysis of the relevant variables.
Taking the most important variables, we have reached the following interpretation.

	Cluster 1	**Cluster 2**	**Cluster 3**
Hotel type	Resort	City	City
Is canceled	Not canceled	Canceled	Not canceled
Lead Time	MID	HIGH	LOW
Arrival Month	Vacation Months	Across Year	Across Year (↑)
Stays week nights	Stay on weekend nights	Few weekdays (↓)	Few weekdays (↑)
Babies	HIGH	LOW	LOW-MID
Meal	Absence SC	Absence FB	Absence FB
Country	Portugal & UK (↑)	Portugal	Portugal && Europa
Days waiting list	LOW-MID	HIGH	MID
Reservation Status	Check-out (↓)	Canceled	Check-out (↑)

1. **Class 1 : Family**
    - Cluster 1 is associated with families, as it has **more resort bookings** and a **higher presence of babies**.
2. **Class 2 : Canceled bookings**
    - Cluster 2 consists exclusively of **cancelled bookings**, with **longer lead times** and **more days on the waiting list**.
3. **ClaSs 3 : Commercial workers**
    - Cluster 3 represents **commercial workers**, with **short stays during the week** in **urban hotels**.

Which variables are related to each other within a cluster?

- The variable ‘lead_time’ suggests that earlier bookings with a long period of time have a higher probability of cancellation, which is a relevant finding for hotel management.
- Also, a similar situation occurs with “days_in_waiting_list”, if booked a few days during the week for a city hotel, in non-holiday periods, it is more likely to have more days on the waiting list, causing many of these bookings to be cancelled.

10. Conclusion

Descriptive analysis:

The descriptive analysis reveals a majority profile of new customers who book urban hotels at short notice, prefer basic rooms with breakfast and rarely include children. High cancellations and low loyalty point to opportunities to improve retention and risk management.

PCA vs Clustering

On one hand, PCA helped us to understanding travel patterns, customer types and seasonal reservations, comparing different PCAs with different dimensions because it allows us to get a more complete picture of the variability in the data and facilitates the interpretation of patterns that may not be evident in a single projection.

On the other hand, clustering + profiling has been useful for us to definitively classify these groups that appeared when doing the PCA, thanks to Gower's method and statistical variables such as Chi-square and P-value that emphasize the importance of each of the plots of the variables when interpreting each cluster.

We could say that both have provided us with useful information to know more about the database, but we have to say that it has been easier for us to reach important conclusions in clustering + profiling.

Final conclusion:

✓ We have learned how to treat the information that the variables of a database can give us (some more than others).

✓ We have seen the importance of cleaning/simplifying our database in order to carry out statistical methods to extract more information.

✓ We have been able to detect patterns between variables in the database that describe the hotel's behaviour, thanks to PCA.

✓ We have been able to identify and classify groups in the database, based on the difference between them, thanks to clustering.

✓ We have learned different statistical methods of DM and how to carry this out with R and how to correctly interpret the generated plots.

By clustering our dataset, grouping variables and discarding some, we obtained 3 clusters. When analysed by profiling with the different statistical tests, we were able to distinguish and classify groups as mentioned in section 9.1. Like the PCA, it has been useful to distinguish and find the commonalities between data, which are difficult to perceive with the naked eye.

The fact that we have been able to cluster individuals into groups with certain characteristics has been possible due to reducing the number of dimensions with the PCA and then clustering with the Gower dissimilarity coefficient, with clustering + profiling being the key to this report. Finding patterns and being able to categorise data is fundamental for data mining or other future treatments. Moreover, this can be useful to show real statistical information for clients and investors, who are interested in this hotel chain.

11. Working Plan

Initial Gantt chart:

Name	Responsible	Date	Chronogram - Start	Chronogram - End
Database search and selection	Alex, Gina, Sergio		2025-02-13	2025-02-13
Metadata file	Sergio		2025-02-17	2025-02-23
Descriptive analysis	Gina, Alex, Sergio		2025-02-17	2025-02-17
Preprocessing Process	Alex, Sergio		2025-03-03	2025-03-03
List and justifying decisions for each preprocessing step	Alex		2025-03-03	2025-03-03
Additional descriptive statistics	Gina		2025-03-03	2025-03-03
D4: Motivation of the work	Maros, Kimia		2025-03-04	2025-03-07
D4 :Data Source presentation	Maros, Kimia		2025-03-04	2025-03-07
D4: Formal description of Data structure and metadata	Sergio		2025-03-04	2025-03-09
D4: Complete Data Mining process performed	Gina		2025-03-04	2025-03-09
D4: Detailed description of Preprocessing and data preparation.	Alex		2025-03-04	2025-03-09
D4: Basic statistical descriptive analysis	Alex, Sergio, Gina		2025-03-04	2025-03-09
D4: PCA analysis for numerical variables:	Kimia, Maros		2025-03-10	2025-03-16
D4: Hierarchical Clustering on original data:	Kimia, Maros		2025-03-10	2025-03-16
D4: Profiling of clusters:	Kimia, Maros		2025-03-10	2025-03-16

D4: Global discussion and general conclusions	Sergio, Alex, Gina, Kimia, Maros		2025-03-17	2025-03-19
D4: Working plan (based on this)	Alex, Kimia		2025-02-17	2025-02-17
D4: design slides presentation	Maros, Kimia			
		Since 2025-02-13 to 2025-04-23	**2025-02-13**	**2025-03-24**

Final Gantt diagram:

Name	Responsible	Date	Chronogram - Start	Chronogram - End
Database search and selection	Alex, Gina, Sergio	2025-02-13	2025-02-13	2025-02-13
Metadata file	Sergio	2025-02-16	2025-02-17	2025-02-23
Descriptive analysis	Gina, Alex, Sergio	2025-02-23	2025-02-17	2025-02-17
Preprocessing Process	Alex, Sergio	2025-02-18	2025-03-03	2025-03-03
List and justifying decisions for each preprocessing step	Alex	2025-02-18	2025-03-03	2025-03-03
Additional descriptive statistics	Gina, Alex	2025-02-18	2025-03-03	2025-03-03
D4: Motivation of the work	Maros, Kimia	2025-03-10	2025-03-04	2025-03-07
D4 :Data Source presentation	Maros, Kimia	2025-03-10	2025-03-04	2025-03-07
D4: Formal description of Data structure and metadata	Sergio	2025-03-09	2025-03-04	2025-03-09
D4: Complete Data Mining process performed	Gina	2025-03-09	2025-03-04	2025-03-09
D4: Detailed description of Preprocessing and data preparation.	Alex	2025-03-09	2025-03-04	2025-03-09
D4: Basic statistical descriptive analysis	Alex, Sergio, Gina	2025-03-09	2025-03-04	2025-03-09

D4: PCA analysis for numerical variables:	Kimia, Maros	2025-03-16	2025-03-10	2025-03-16
D4: Hierarchical Clustering on original data:	Alex	2025-03-16	2025-03-10	2025-03-16
D4: Profiling of clusters:	Sergio, Gina	2025-03-16	2025-03-10	2025-03-16
D4: Global discussion and general conclusions	Sergio, Alex, Gina	2025-03-19	2025-03-17	2025-03-19
D4: Working plan (based on this)	Alex, Kimia	2025-03-17	2025-02-17	2025-02-17
D4: design slides presentation	Maros, Kimia	2025-03-17	2025-03-19	2025-03-17
		Since 2025-02-13 to 2025-04-23	**2025-02-13**	**2025-03-24**

11. R scripts

11.1 Descriptive

```
---
title: "D3. Project development: descriptive analysis"
output: word_document
editor_options:
  chunk_output_type: console
---
#7TotaldescriptivaClean5.Rmd
#install.packages("rmarkdown")
#library("rmarkdown")

## Introduction
This document provides the initial univariate descriptive statistics of the raw variables in the hotel booking dataset.

#loading libraries
install.packages("ggplot2")
library(ggplot2)
install.packages("dplyr")
library(dplyr)

#WARNING: data must have been properly declared before (factors, dates...)

#Set the folder where the word file is going to be generated
```{r, echo=FALSE}

setwd("/home/alex/Pictures")
dd<-  read.table("hotel_booking_5000_rows.csv",header=T,  sep=",", dec=".")

```
```

```
#without including the R instruction in the final document
```{r, echo=FALSE}
class(dd)
```

#Get dimensions of the dataset
````{r, echo=FALSE}
dim(dd)
n<-dim(dd)[1]
K<-dim(dd)[2]

n
K
````

#Check the variables
```{r, echo=FALSE}
names(dd)
str(dd)
head(dd)
summary(dd)
```

#Decide if you need to declare some more factor or date
```{r, echo=FALSE}
#para la conversión numérica de los meses
library(dplyr)

month_mapping <- setNames(1:12, month.name)
dd <- dd %>%
  mutate(
     arrival_date_year = as.numeric(arrival_date_year),
     arrival_date_month                                    =
as.numeric(month_mapping[arrival_date_month]),
```
```

```
    arrival_date_day_of_month                                    =
as.numeric(arrival_date_day_of_month)
  )


# creación arrival_date, combinación 3 columnas
dd <- dd %>%
  mutate(
    arrival_date = as.Date(
    paste(arrival_date_year,                arrival_date_month,
arrival_date_day_of_month, sep = "-"),
    format = "%Y-%m-%d"
    )
  )

dd <- dd %>%
  mutate(
    hotel = as.factor(hotel),
    is_canceled = as.factor(is_canceled),
    meal = as.factor(meal),
    country = as.factor(country),
    market_segment = as.factor(market_segment),
    distribution_channel = as.factor(distribution_channel),
    is_repeated_guest = as.factor(is_repeated_guest),
    reserved_room_type = as.factor(reserved_room_type),
    assigned_room_type = as.factor(assigned_room_type),
    deposit_type = as.factor(deposit_type),
    customer_type = as.factor(customer_type),
    reservation_status = as.factor(reservation_status),
    company = as.factor(company),
    arrival_date_week_number                                     =
as.factor(arrival_date_week_number)
  )

dd <- dd %>%
```

```
  select(-c(name,      email,      phone.number,      credit_card,
arrival_date_year,                          arrival_date_month,
arrival_date_day_of_month))

K<-dim(dd)[2]

str(dd)

```

#Calculation of mean, median and standard deviation
```{r, echo=FALSE}
numeric_cols <- dd %>%
  select(where(is.numeric))

summary_numcols <- numeric_cols %>%
     summarise_all(list(
     mean = ~mean(., na.rm = TRUE),
     median = ~median(., na.rm=TRUE),
     sd = ~sd(., na.rm = TRUE),
     var = ~var(., na.rm = TRUE)
     ))

print(summary_numcols)

```

#Given scheme for descriptive analyis
```{r, echo=FALSE}

descriptiva<-function(X, nom){
  if (!(is.numeric(X) || class(X)=="Date")){
     frecs<-table(as.factor(X), useNA="ifany")
     proportions<-frecs/n
     #ojo, decidir si calcular porcentages con o sin missing values
```

```
    pie(frecs, cex=0.6, main=paste("Pie of", nom))
    barplot(frecs, las=3, cex.names=0.7, main=paste("Barplot of",
nom), col=listOfColors)
    print(paste("Number of modalities: ", length(frecs)))
    print("Frequency table")
    print(frecs)
    print("Relative frequency table (proportions)")
    print(proportions)
    print("Frequency table sorted")
    print(sort(frecs, decreasing=TRUE))
    print("Relative frequency table (proportions) sorted")
    print(sort(proportions, decreasing=TRUE))
  }else{
    if(class(X)=="Date"){
    print(summary(X))
    print(sd(X))
    #decide breaks: weeks, months, quarters...
    hist(X,breaks="weeks")
    }else{
    hist(X, main=paste("Histogram of", nom))
    boxplot(X, horizontal=TRUE, main=paste("Boxplot of",nom))
    print("Extended Summary Statistics")
    print(summary(X))
    print(paste("sd: ", sd(X, na.rm=TRUE)))
    print(paste("vc: ", sd(X, na.rm=TRUE)/mean(X, na.rm=TRUE)))
    }
  }
}

dataset<-dd
actives<-c(1:K)

```

#Basic descriptive analysis for numerical variables

```
#(decide the maximum number of colors you can need in a graph based on your metadata file)
```{r, echo=FALSE}
listOfColors<-rainbow(39)

par(ask=TRUE)

for(k in actives){
  print(paste("variable ", k, ":", names(dd)[k] ))
  descriptiva(dd[,k], names(dd)[k])
}
par(ask=FALSE)
```


#Bivariate analysis
```{r, echo=FALSE}

temp_numeric_cols <- numeric_cols

temp_numeric_cols <- temp_numeric_cols %>%
  select(-c(previous_cancellations,
previous_bookings_not_canceled,      booking_changes,      agent,
required_car_parking_spaces, total_of_special_requests)
     )


correlation_matrix <- cor(temp_numeric_cols, use = "complete.obs")
# Handles missing values
print(correlation_matrix)
correlation_matrix |>
  round(2) |>
  write.csv("correlation_matrix.csv", row.names = TRUE)

par(ask=TRUE)
library(ggplot2)
```
```

```
ggplot(dd, aes(x = stays_in_weekend_nights, y = stays_in_week_nights)) +
  geom_point(alpha = 0.5, color = "steelblue", na.rm = TRUE) +
  geom_smooth(method = "lm", color = "red", se = FALSE) +
  labs(title = "Relationship between Weekend and Weekday Stays",
     x = "Stays in Weekend Nights",
     y = "Stays in Week Nights") +
  theme_minimal()

# Scatter plot for lead_time vs arrival_date_week_number
ggplot(dd, aes(x = as.numeric(as.character(arrival_date_week_number)), y = lead_time)) +
  geom_point(alpha = 0.4, color = "darkgreen", na.rm = TRUE, position = position_jitter(width = 0.2)) +
  geom_smooth(method = "loess", color = "orange", se = FALSE) +
  labs(title = "Lead Time vs Booking Week",
     x = "Arrival Week Number (1-53)",
     y = "Lead Time (Days)") +
  theme_minimal() +
  scale_x_continuous(breaks = seq(1, 53, 4))

par(ask=FALSE)

```


```
#per exportar figures d'R per programa
#dev.off()
#png(file=mypath,width = 950, height = 800, units = "px")
#dev.off()
```

11.2 Preprocessing

```
setwd("/home/alex/Pictures")

#dd <- read.table("hotel_booking_5000_rows.csv",header=T);

#dd <- read.table("credsco.csv",header=T, sep=";",na.strings="\"\"");
#dd <- read.table("credsco.csv",header=T, sep=";");

dd <- read.csv("hotel_booking_5000_rows.csv", stringsAsFactors=TRUE);

class(dd)
dim(dd)
n<-dim(dd)[1]
n
K<-dim(dd)[2]
K
names(dd)

library(dplyr)

month_mapping <- setNames(1:12, month.name)
dd <- dd %>%
  # mutación temporal para crear "arrival_date"
  mutate(
      arrival_date_year = as.character(arrival_date_year),
      arrival_date_month =
as.character(month_mapping[as.character(arrival_date_month)]),
      arrival_date_day_of_month =
as.character(arrival_date_day_of_month),
  )


# creación arrival_date, combinación 3 columnas
dd <- dd %>%
  mutate(
      arrival_date = as.Date(
      paste(arrival_date_year, arrival_date_month,
arrival_date_day_of_month, sep = "-"),
```

```
      format = "%Y-%m-%d"
      )
  )

dd <- dd %>%
  mutate(
      hotel = as.factor(hotel),
      is_canceled = as.factor(is_canceled),
      meal = as.factor(meal),
      country = as.factor(country),
      market_segment = as.factor(market_segment),
      distribution_channel = as.factor(distribution_channel),
      is_repeated_guest = as.factor(is_repeated_guest),
      reserved_room_type = as.factor(reserved_room_type),
      assigned_room_type = as.factor(assigned_room_type),
      deposit_type = as.factor(deposit_type),
      customer_type = as.factor(customer_type),
      reservation_status = as.factor(reservation_status),
      company = as.factor(company),
      arrival_date_year = as.factor(arrival_date_year),
      arrival_date_month = as.factor(arrival_date_month),
      arrival_date_day_of_month = as.factor(arrival_date_day_of_month),
      arrival_date_week_number = as.factor(arrival_date_week_number)
  )

dd <- dd %>%
  mutate(arrival_date_month = factor(arrival_date_month,
                                     levels = 1:12,
                                     labels = month.name))


#select some variables
#DO NOT INCLUDE identifiers in the analysis!!!!
#actives<-c(1:5,7:11, 13:23, 26:32)
#actives<-c(1:5,7:11, 13:23, 26:32)
dd <- dd %>%
  select(-c(total_of_special_requests, previous_bookings_not_canceled,
previous_cancellations,
```

```
           reservation_status_date, deposit_type, company, agent,
           distribution_channel, market_segment,
required_car_parking_spaces, reserved_room_type,
           assigned_room_type, is_repeated_guest, name, email,
phone.number, credit_card, booking_changes,
           customer_type, arrival_date_day_of_month,
arrival_date_year)
     )
#ddActives<-dd[,actives]

#Care! update K

K<-dim(dd)[2]

str(dd)

# Change variables' names

dd <- dd %>% rename(can = is_canceled)
dd <- dd %>% rename(lt = lead_time)
dd <- dd %>% rename(arr_m = arrival_date_month)
dd <- dd %>% rename(arr_wn  = arrival_date_week_number)
dd <- dd %>% rename(s_wend_n = stays_in_weekend_nights)
dd <- dd %>% rename(s_wday_n  = stays_in_week_nights)
dd <- dd %>% rename(d_wl = days_in_waiting_list)
dd <- dd %>% rename(res_s = reservation_status)
dd <- dd %>% rename(arr = arrival_date)

#write.table(dd, file = "BookingCleanPrueba.csv", sep = ";", na =
"NA", dec = ".", row.names = FALSE, col.names = TRUE)

numeric_cols <- dd %>%
  select(where(is.numeric))

qualitative_cols <- dd %>%
  select(where(is.factor))
```

```
#seleccion some rows
#Dselection<-dd[Condition, ]

library(ggplot2)
library(GGally)

descriptiva<-function(X, nom){
  if (is.numeric(X) && class(X)!= "Date"){
      hist(X, main=paste("Histogram of", nom))
      boxplot(X, horizontal=TRUE, main=paste("Boxplot of",nom))
      print("Extended Summary Statistics")

      cat("Extended Summary Statistics:\n")
      print(summary(X))

      cat("Standard Deviation:\n")
      print(sd(X, na.rm = TRUE))

      cat("Variation Coefficient:\n")
      print(sd(X, na.rm = TRUE) / mean(X, na.rm = TRUE))

      #print(summary(X))
      #print(paste("sd: ", sd(X, na.rm=TRUE)))
      #print(paste("vc: ", sd(X, na.rm=TRUE)/mean(X, na.rm=TRUE)))
  }

}

dataset<-dd
actives<-c(1:K)

#Basic descriptive analysis for numerical variables
#(decide the maximum number of colors you can need in a graph based on
your metadata file)

listOfColors<-rainbow(39)

par(ask=TRUE)
```

```r
#Output graphs to a pdf
pdf("/home/alex/Downloads/histograms.pdf")
for(k in actives){
  print(paste("variable ", k, ":", names(dd)[k] ))
  descriptiva(dd[,k], names(dd)[k])
}
dev.off()  # Close PDF after all plots are generated
par(ask=FALSE)

# Replace instances in days_in_waiting_list that are > 300 with NA
class(dd)
#attach(dd)
# Directly reference the data frame column (d_wl)
dd$d_wl[dd$d_wl > 300] <- NA

# Now, you can check how many NA values you have
prueba <- dd$d_wl[is.na(dd$d_wl)]
length(prueba)

# Step 1: Identify numerical variables in the dataset
numeric_vars <- names(dd)[sapply(dd, is.numeric)]  # Select only
numeric variables

# Step 2: Ensure "d_wl" (or the variable you want to impute) is in the
selected variables
fullVariables <- unique(c("d_wl", numeric_vars))  # Add "d_wl" if not
already included

# Step 3: Create the auxiliary matrix with selected numerical
variables
aux <- dd[, fullVariables, drop = FALSE]  # Keep it as a data frame

# Step 4: Remove rows where any predictor (except "d_wl") has missing
values
aux_clean <- aux[complete.cases(aux[, -which(names(aux) == "d_wl")]),
]
```

```r
# divide in rows that had missing incomes or not on the target
variable to be imputed
aux1 <- aux_clean[!is.na(aux_clean$d_wl),]
dim(aux1)
aux2 <- aux_clean[is.na(aux_clean$d_wl),]
dim(aux2)

#Find nns for aux2
#knn.ing = knn(aux1,aux2,d_wl[!is.na(d_wl)])

# Step 6: Apply KNN for imputation
library(class)  # Load the necessary package
knn.ing <- knn(
  train = aux1[, -which(names(aux1) == "d_wl")],  # Predictors only
  test  = aux2[, -which(names(aux2) == "d_wl")],  # Predictors only
  cl  = aux1$d_wl,  # Known values of d_wl
  k   = 10  # Choose an appropriate k
)

#CARE: neither aux1 nor aux2 can contain NAs


#CARE: knn.ing is generated as a factor.
#Be sure to retrieve the correct values

days_in_waiting_listOriginal<-dd$d_wl

# Step 7: Assign Imputed Values
dd$d_wl[is.na(dd$d_wl)] <- as.numeric(levels(knn.ing))[knn.ing]

# Step 8: Verify Results
summary(dd$d_wl)  # Check if the missing values were imputed properly



#saving the dataframe in an external file
write.table(dd, file = "BookingClean.csv", sep = ";", na = "NA", dec =
".", row.names = FALSE, col.names = TRUE)
```

11.3 PCA

```
#PREREQUISITES:
#factors are properly labelled and reading data makes R to directly
recognize them
#Numerical variables do not contain missing values anymore. They have
been imputed in preprocessing step

# setwd("/home/alex/Pictures/")
setwd("/Users/marosbednar/Library/CloudStorage/OneDrive-
SlovenskátechnickáuniverzitavBratislave/Skola/Datamining")

dd <- read.table("BookingClean.csv",header=T, sep=";",
stringsAsFactors = TRUE);

objects()
attributes(dd)

#
# VISUALISATION OF DATA
#
# PRINCIPAL COMPONENT ANALYSIS OF CONTINcUOUS VARIABLES, WITH Dictamen
PROJECTED AS ILLUSTRATIVE
#

# CREATION OF THE DATA FRAME OF CONTINUOUS VARIABLES

attach(dd)
names(dd)

#is R understanding well my factor variables?
sapply(dd,class)

#set a list of numerical variables (with no missing values)

library(dplyr)

dd <- dd %>%
  mutate(
```

```
    # arr_y = as.factor(arr_y),
    arr_m = as.factor(arr_m),
    arr_wn = as.factor(arr_wn),
    can = as.factor(can),
  )

#month_names <- c("January", "February", "March", "April", "May",
"June",
#                 "July", "August", "September", "October", "November",
"December")
#dd$arr_m <- factor(dd$arr_m, labels = month_names)

numeriques<-which(sapply(dd,is.numeric))
numeriques

# exclude columns that were merged into other columns
excluded_cols <- c("s_wend_n", "s_wday_n", "adults", "children",
"babies")
numeriques <- numeriques[!names(dd)[numeriques] %in% excluded_cols]
numeriques

# Create an array of the names of numerical columns
numeriques_names <- names(dd)[numeriques]
numeriques_names

dcon<-dd[,numeriques]
sapply(dcon,class)

#dcon <- data.frame
(Antiguedad.Trabajo,Plazo,Edad,Gastos,Ingresos,Patrimonio,Cargas.patri
moniales,Importe.solicitado,Precio.del.bien.financiado,Estalvi,
RatiFin)

#alternatively
#dim(dd)
#indexCon<-c(2,4:5,9:16)
#dcon<-dd[,indexCon]
#names(dcon)
```

```r
#be sure you don't have missing data in your numerical variables.

#in case of having missing data, select complete rows JUST TO FOLLOW
THE CLASS
#dd<-dd[!is.na(dd[,indecCon[1]])& !is.na(dd[,indecCon[2]]) &
!is.na(dd[,indecCon[3]])& !is.na(dd[,indecCon[4]]),]
#then preprocess your complete data set to IMPUTE all missing data,
and reproduce
#the whole analysis again
# PRINCIPAL COMPONENT ANALYSIS OF dcon

pc1 <- prcomp(dcon, scale=TRUE)
class(pc1)
attributes(pc1)

print(pc1)

str(pc1)

# 7) Inspect the variance each component explains
variances <- pc1$sdev^2
prop_var_explained <- variances / sum(variances)
prop_var_explained

barplot(100*prop_var_explained,
     main="Scree Plot",
     xlab="Principal Components",
     ylab="% Variance Explained")


# WHICH PERCENTAGE OF THE TOTAL INERTIA IS REPRESENTED IN SUBSPACES?

pc1$sdev
inerProj<- pc1$sdev^2
inerProj
totalIner<- sum(inerProj)
totalIner
```

```r
pinerEix<- 100*inerProj/totalIner
pinerEix
barplot(pinerEix)

#Cummulated Inertia in subspaces, from first principal component to
the 11th dimension subspace
barplot(100*cumsum(pc1$sdev[1:dim(dcon)[2]]^2)/dim(dcon)[2],
     main="Scree Plot",
     xlab="Principal Components",
     ylab="Cumulative Variance in %")
percInerAccum<-100*cumsum(pc1$sdev[1:dim(dcon)[2]]^2)/dim(dcon)[2]
percInerAccum


# SELECTION OF THE SINGIFICNT DIMENSIONS (keep 80% of total inertia)

nd = 3

print(pc1)
attributes(pc1)
pc1$rotation

# STORAGE OF THE EIGENVALUES, EIGENVECTORS AND PROJECTIONS IN THE nd
DIMENSIONS
View(pc1$x)
dim(pc1$x)
dim(dcon)
dcon[2000,]
pc1$x[2000,]

Psi = pc1$x[,1:nd]
dim(Psi)
Psi[2000,]

# STORAGE OF LABELS FOR INDIVIDUALS AND VARIABLES

iden = row.names(dcon)
etiq = names(dcon)
```

```r
ze = rep(0,length(etiq)) # WE WILL NEED THIS VECTOR AFTERWARDS FOR THE
GRAPHICS

# PLOT OF INDIVIDUALS

#select your axis
eje1<-1
eje2<-2
eje3<-3

plot(Psi[,eje1],Psi[,eje3])
text(Psi[,eje1],Psi[,eje3],labels=iden, cex=0.5)
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

# install.packages("rgl")
# library(rgl)
# plot3d(Psi[,1],Psi[,2],Psi[,3])

# Projection of variables

Phi = cor(dcon,Psi)
View(Phi)

#select your axis

X<-Phi[,eje1]
Y<-Phi[,eje3]

# Maros - we need ZOOMS of this arrows only
# plot(Psi[,eje1],Psi[,eje2],type="n")
# axis(side=1, pos= 0, labels = F)
# axis(side=3, pos= 0, labels = F)
# axis(side=2, pos= 0, labels = F)
# axis(side=4, pos= 0, labels = F)
# arrows(ze, ze, X, Y, length = 0.07,col="blue")
```

```
# text(X,Y,labels=etiq,col="darkblue", cex=0.7)

#zooms 1
X<-Phi[,eje1]
Y<-Phi[,eje3]
plot(Psi[,eje1],Psi[,eje3],type="n",xlim=c(min(X,0),max(X,0)),
ylim=c(-1,1))
axis(side=1, pos= 0, labels = F)
axis(side=3, pos= 0, labels = F)
axis(side=2, pos= 0, labels = F)
axis(side=4, pos= 0, labels = F)
arrows(ze, ze, X, Y, length = 0.07,col="blue")
text(X,Y,labels=etiq,col="darkblue", cex=1.0)

#zooms 2
X<-Phi[,eje1]
Y<-Phi[,eje3]
plot(Psi[,eje1],Psi[,eje3],type="n",xlim=c(min(X,0),max(X,0)),
ylim=c(-1,1))
axis(side=1, pos= 0, labels = F)
axis(side=3, pos= 0, labels = F)
axis(side=2, pos= 0, labels = F)
axis(side=4, pos= 0, labels = F)
arrows(ze, ze, X, Y, length = 0.07,col="blue")
text(X,Y,labels=etiq,col="darkblue", cex=1.0)

# PROJECTION OF ILLUSTRATIVE qualitative variables on individuals' map
# PROJECCI? OF INDIVIDUALS DIFFERENTIATING THE Dictamen
# (we need a numeric Dictamen to color)

varcat=factor(dd[,1])
plot(Psi[,1],Psi[,2],col=varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")

legend("bottomleft",levels(factor(varcat)),pch=1,col=c(1,2), cex=0.6)
```

```
#select your qualitative variable
k<-1 #dictamen in credsco

varcat<-factor(dd[,k])
fdic1 = tapply(Psi[,eje1],varcat,mean)
fdic2 = tapply(Psi[,eje2],varcat,mean)
#points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))
text(fdic1,fdic2,labels=levels(varcat),col="RED", cex=0.7)


#Now we project both cdgs of levels of a selected qualitative variable
without
#representing the individual anymore

plot(Psi[,eje1],Psi[,eje2],type="n")
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#select your qualitative variable
k<-12 #dictamen in credsco

varcat<-dd[,k]
fdic1 = tapply(Psi[,eje1],varcat,mean)
fdic2 = tapply(Psi[,eje2],varcat,mean)

points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))
text(fdic1,fdic2,labels=levels(varcat),col="blue", cex=0.7)

# START IMPORTANT1

#all qualitative together
plot(Psi[,eje1],Psi[,eje2],type="n")
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
```

```
axis(side=4, pos= 0, labels = F, col="cyan")

#nominal qualitative variables

dcat<-c(1,4,10,11,12)
#divide categoricals in several graphs if joint representation
saturates

#build a palette with as much colors as qualitative variables

#colors<-c("blue","red","green","orange","darkgreen")
#alternative
colors<-rainbow(length(dcat))

c<-1
for(k in dcat){
  seguentColor<-colors[c]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.9)
  c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.9)




# START PC1 x PC2
#determine zoom level
#use the scale factor or not depending on the position of centroids
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL
GRAFIC
fm = round(max(abs(Psi[,1])))
fm=20

# wtf is this "U"?
```

```
#scale the projected variables
# X<-fm*U[,eje1]
# Y<-fm*U[,eje2]

X<-Phi[,eje1]
Y<-Phi[,eje2]

#represent numerical variables in background
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="black")
text(X,Y,labels=etiq,col="black", cex=0.7)

#add centroids
c<-1
for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.6)
  c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.6)


#add ordinal qualitative variables. Ensure ordering is the correct
# END PC1 x PC2
```

```
# START PC1 x PC3
#determine zoom level
#use the scale factor or not depending on the position of centroids
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL
GRAFIC
fm = round(max(abs(Psi[,1])))
#fm=20

X<-Phi[,eje1]
Y<-Phi[,eje3]

#represent numerical variables in background
plot(Psi[,eje1],Psi[,eje3],type="n",xlim=c(-1,1), ylim=c(-3,1))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="black")
text(X,Y,labels=etiq,col="black", cex=0.7)

#add centroids
c<-1
for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje3],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.6)
  c<-c+1
}
```

```
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.6)


#add ordinal qualitative variables. Ensure ordering is the correct

# END PC1 x PC3



# START PC2 x PC3
#determine zoom level
#use the scale factor or not depending on the position of centroids
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL
GRAFIC
fm = round(max(abs(Psi[,2])))
#fm=20

X<-Phi[,eje2]
Y<-Phi[,eje3]

#represent numerical variables in background
plot(Psi[,eje2],Psi[,eje3],type="n",xlim=c(-1,1), ylim=c(-3,1))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="black")
text(X,Y,labels=etiq,col="black", cex=0.7)

#add centroids
c<-1
for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje2],dd[,k],mean)
```

```
  fdic2 = tapply(Psi[,eje3],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.6)
  c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=1,col=colors, cex=0.6)


#add ordinal qualitative variables. Ensure ordering is the correct

# END PC2 x PC3


dordi<-c(1)

levels(factor(dd[,dordi[1]]))
#reorder modalities: when required
dd[,dordi[1]] <- factor(dd[,dordi[1]], ordered=TRUE,  levels=
c("WorkingTypeUnknown","altres sit","temporal","fixe","autonom"))
levels(dd[,dordi[1]])

c<-1
col<-1
for(k in dordi){
  seguentColor<-colors[col]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  #connect modalities of qualitative variables
  lines(fdic1,fdic2,pch=16,col=seguentColor)
  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
  col<-col+1
}
```

```
legend("topleft",names(dd)[dordi],pch=1,col=colors[1:length(dordi)],
cex=0.6)

# END IMPORTANT1

#using our own colors palette
# search palettes in internet. One might be https://r-
charts.com/es/colores/

colors<-c("red", "blue", "darkgreen", "orange", "violet", "magenta",
"pink")

#represent numerical variables in background
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")
text(X,Y,labels=etiq,col="gray", cex=0.7)

#add centroids
c<-1
for(k in dcat){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.6)
  c<-c+1
}
legend("bottomleft",names(dd)[dcat],pch=19,col=colors, cex=0.6)
```

```r
#add ordinal qualitative variables. Ensure ordering is the correct

dordi<-c(8)


levels(factor(dd[,dordi[1]]))
#reorder modalities: when required
dd[,dordi[1]] <- factor(dd[,dordi[1]], ordered=TRUE,  levels=
c("WorkingTypeUnknown","altres sit","temporal","fixe","autonom"))
levels(dd[,dordi[1]])

c<-1
col<-length(dcat)+1
for(k in dordi){
  seguentColor<-colors[col]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  #connect modalities of qualitative variables
  lines(fdic1,fdic2,pch=16,col=seguentColor)
  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
  col<-col+1
}
legend("topleft",names(dd)[dordi],pch=19,col=colors[col:col+length(dor
di)-1], cex=0.6)


#Make two complementary factorial maps

colors<-c("red", "blue", "darkgreen", "orange", "violet", "magenta",
"pink")

#represent numerical variables in background
```

```
#plot(Psi[,eje1],Psi[,eje2],type="p",xlim=c(-1,1), ylim=c(-3,1),
col="lightgray")
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))
axis(side=1, pos= 0, labels = F, col="cyan")
axis(side=3, pos= 0, labels = F, col="cyan")
axis(side=2, pos= 0, labels = F, col="cyan")
axis(side=4, pos= 0, labels = F, col="cyan")

#add projections of numerical variables in background
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")
text(X,Y,labels=etiq,col="gray", cex=0.7)

#numerical variables of financial situation

seleccio<-c(1,2,3,4,5)
seleccio
seleccio
seleccio
seleccio
seleccio
dconMapa1<-dcon[,seleccio]

#referencia general comu a tots els mapes
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")
text(X,Y,labels=etiq,col="gray", cex=0.7)

#represent in the map1
XMapa1<-Phi[seleccio,eje1]
YMapa1<-Phi[seleccio,eje2]

arrows(ze, ze, XMapa1, YMapa1, length = 0.07,col="green")
text(XMapa1,YMapa1,labels=names(dconMapa1),col="green", cex=0.7)


#add centroids
dcatMapa1<-c(7)
```

```
c<-1
for(k in dcatMapa1){
  seguentColor<-colors[c]

  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  text(fdic1,fdic2,labels=levels(factor(dd[,k])),col=seguentColor,
cex=0.6)
  c<-c+1
}
legend("bottomleft",names(dd)[dcatMapa1],pch=19,col=colors, cex=0.6)


#add ordinal qualitative variables. Ensure ordering is the correct

dordi<-c(8)


levels(factor(dd[,dordi[1]]))
#reorder modalities: when required
dd[,dordi[1]] <- factor(dd[,dordi[1]], ordered=TRUE,  levels=
c("WorkingTypeUnknown","altres sit","temporal","fixe","autonom"))
levels(dd[,dordi[1]])

c<-1
col<-length(dcat)+1
for(k in dordi){
  seguentColor<-colors[col]
  fdic1 = tapply(Psi[,eje1],dd[,k],mean)
  fdic2 = tapply(Psi[,eje2],dd[,k],mean)

  #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))
  #connect modalities of qualitative variables
  lines(fdic1,fdic2,pch=16,col=seguentColor)
  text(fdic1,fdic2,labels=levels(dd[,k]),col=seguentColor, cex=0.6)
  c<-c+1
```

```
  col<-col+1
}
legend("topleft",names(dd)[dordi],pch=19,col=colors[col:col+length(dor
di)-1], cex=0.6)


# PROJECTION OF ILLUSTRATIVE qualitative variables on individuals' map
# PROJECCI? OF INDIVIDUALS DIFFERENTIATING THE Dictamen
# (we need a numeric Dictamen to color)

varcat=factor(dd[,1])
plot(Psi[,1],Psi[,2],col=varcat)
axis(side=1, pos= 0, labels = F, col="darkgray")
axis(side=3, pos= 0, labels = F, col="darkgray")
axis(side=2, pos= 0, labels = F, col="darkgray")
axis(side=4, pos= 0, labels = F, col="darkgray")
legend("bottomleft",levels(varcat),pch=1,col=c(1,2), cex=0.6)

# Overproject THE CDG OF  LEVELS OF varcat
fdic1 = tapply(Psi[,1],varcat,mean)
fdic2 = tapply(Psi[,2],varcat,mean)

text(fdic1,fdic2,labels=levels(factor(varcat)),col="cyan", cex=0.75)
```

11.4 Clustering + Profiling

```
#Retrieve the data saved AFTER the preprocessing practice..... this means data already cleaned

setwd("/home/alex/Pictures/")
dd <- read.table("BookingClean.csv",header=T, sep=";",
stringsAsFactors = TRUE);
names(dd)
dim(dd)
summary(dd)

#attach(dd)

#set a list of numerical variables
names(dd)

#hierarchical clustering

#euclidean distance si totes son numeriques
library(dplyr)

tipos <- sapply(dd, class)
varCat <- names(tipos)[which(tipos %in% c("factor", "character"))]
varCat <- c(varCat, "arr_wn", "can", "s_wend_n", "s_wday_n")

for(vC in varCat) {dd[, vC] <- as.factor(dd[, vC])}


#move to Gower mixed distance to deal
#simoultaneously with numerical and qualitative data

#library(dplyr)

# When wanting the plot without "is_cancelled" used in the report to see our k more clearly,
# remove "can" too
dd <- dd %>%
  select(-c(arr, arr_wn)
```

```
  )

library(cluster)

#dissimilarity matrix
#do not include in actives the identifier variables nor the potential
response variable
dissimMatrix <- daisy(dd, metric = "gower", stand = TRUE)
distMatrix <- dissimMatrix^2
h1 <- hclust(distMatrix, method = "ward.D2")  # NOTICE THE COST
#versions noves "ward.D" i abans de plot: par(mar=rep(2,4)) si se
quejara de los margenes del plot

plot(h1)
  abline(h = 2.55, col = "red")

k<-3 #number of clusters to change for ours

dd[, "cluster"] <- cutree(h1,k)

#class sizes
table(dd[, "cluster"])

#comparing with other partitions
#table(c1,c2)


#Profiling plots


names(dd)

#attach(dd)

#Dictamen  <- as.factor(Dictamen)
#levels(Dictamen) <- c(NA, "positiu","negatiu")
```

```
#Calcula els valor test de la variable Xnum per totes les modalitats
del factor P
ValorTestXnum <- function(Xnum,P){
  #freq dis of fac
  nk <- as.vector(table(P));
  n <- sum(nk);
  #mitjanes x grups
  xk <- tapply(Xnum,P,mean);
  #valors test
  txk <- (xk-mean(Xnum))/(sd(Xnum)*sqrt((n-nk)/(n*nk)));
  #p-values
  pxk <- pt(txk,n-1,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-
pxk[c]}}
  return (pxk)
}
```

```
ValorTestXquali <- function(P,Xquali){
  taula <- table(P,Xquali);
  n <- sum(taula);
  pk <- apply(taula,1,sum)/n;
  pj <- apply(taula,2,sum)/n;
  pf <- taula/(n*pk);
  pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2], byrow=TRUE);
  dpf <- pf - pjm;
  dvt <- sqrt(((1-pk)/(n*pk))%*%t(pj*(1-pj)));
  #i hi ha divisions iguals a 0 dona NA i no funciona
  zkj <- dpf
  zkj[dpf!=0]<-dpf[dpf!=0]/dvt[dpf!=0];
  pzkj <- pnorm(zkj,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){for (s in
1:length(levels(Xquali))){if (pzkj[c,s]> 0.5){pzkj[c,s]<-1-
pzkj[c,s]}}}
  return (list(rowpf=pf,vtest=zkj,pval=pzkj))
}
```

```
#source("file")
#dades contain the dataset
dades<-dd
#dades<-dd[filtro,]
#dades<-df
K<-dim(dades)[2]



#P must contain the class variable
#P<-dd[,3]
c2 <- dd[, "cluster"]
P<-c2
#P<-dd[,18]
nameP<-"classe"
#P<-df[,33]

nc<-length(levels(factor(P)))
nc
pvalk <- matrix(data=0,nrow=nc,ncol=K,
dimnames=list(levels(P),names(dades)))
nameP<-"Class"
n<-dim(dades)[1]

par(ask = TRUE)

for(k in 1:K){
  if (is.numeric(dades[,k])){
      print(paste("Anàlisi per classes de la Variable:",
names(dades)[k]))

      boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k],
"vs", nameP ), horizontal=TRUE)

      barplot(tapply(dades[[k]], P, mean),main=paste("Means of",
names(dades)[k], "by", nameP ))
```

```
      abline(h=mean(dades[[k]]))
      legend(0,mean(dades[[k]]),"global mean",bty="n")
      print("Estadístics per groups:")
      for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}
      o<-oneway.test(dades[,k]~P)
      print(paste("p-valueANOVA:", o$p.value))
      kw<-kruskal.test(dades[,k]~P)
      print(paste("p-value Kruskal-Wallis:", kw$p.value))
      pvalk[,k]<-ValorTestXnum(dades[,k], P)
      print("p-values ValorsTest: ")
      print(pvalk[,k])
  }else{
      if(class(dd[,k])=="Date"){
      print(summary(dd[,k]))
      print(sd(dd[,k]))
      #decide breaks: weeks, months, quarters...
      hist(dd[,k],breaks="weeks")
      }else{
      #qualitatives
      print(paste("Variable", names(dades)[k]))
      table<-table(P,dades[,k])
      #   print("Cross-table")
      #   print(table)
      rowperc<-prop.table(table,1)

      colperc<-prop.table(table,2)


      dades[,k]<-as.factor(dades[,k])


      marg <- table(as.factor(P))/n
      print(append("Categories=",levels(as.factor(dades[,k]))))

      #from next plots, select one of them according to your practical
case
      plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]))
```

```
      paleta<-rainbow(length(levels(dades[,k])))
      for(c in
1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }

      #with legend
      plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]))
      paleta<-rainbow(length(levels(dades[,k])))
      for(c in
1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }
      legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)

      #condicionades a classes
      print(append("Categories=",levels(dades[,k])))
      plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]))
      paleta<-rainbow(length(levels(dades[,k])))
      for(c in
1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }

      #with legend
      plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]))
      paleta<-rainbow(length(levels(dades[,k])))
      for(c in
1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }
      legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)

      #amb variable en eix d'abcisses
      marg <-table(dades[,k])/n
      print(append("Categories=",levels(dades[,k])))
      plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]), las=3)
      #x<-plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]), xaxt="n")
      #text(x=x+.25, y=-1, adj=1, levels(CountryName), xpd=TRUE,
srt=25, cex=0.7)
      paleta<-rainbow(length(levels(as.factor(P))))
```

```
      for(c in
1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }

      #with legend
      plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]), las=3)
      for(c in
1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}
      legend("topright", levels(as.factor(P)), col=paleta, lty=2,
cex=0.6)

      #condicionades a columna
      plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]), las=3)
      paleta<-rainbow(length(levels(as.factor(P))))
      for(c in
1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }

      #with legend
      plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg
by",names(dades)[k]), las=3)
      for(c in
1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}
      legend("topright", levels(as.factor(P)), col=paleta, lty=2,
cex=0.6)

      table<-table(dades[,k],P)
      print("Cross Table:")
      print(table)
      print("Distribucions condicionades a columnes:")
      print(colperc)

      #diagrames de barres apilades

      paleta<-rainbow(length(levels(dades[,k])))
      barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )

      barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )
```

```
      legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5,
col=paleta)

      #diagrames de barres adosades
      barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )

      barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)
      legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5,
col=paleta)

      print("Test Chi quadrat: ")
      print(chisq.test(dades[,k], as.factor(P)))

      print("valorsTest:")
      print( ValorTestXquali(P,dades[,k]))
      #calcular els pvalues de les quali
      }
  }
}#endfor

par(ask = FALSE)

#descriptors de les classes més significatius. Afegir info qualits
for (c in 1:length(levels(as.factor(P)))) {
  if(!is.na(levels(as.factor(P))[c])){
      print(paste("P.values per class:",levels(as.factor(P))[c]));
      print(sort(pvalk[c,]), digits=3)
  }
}
```