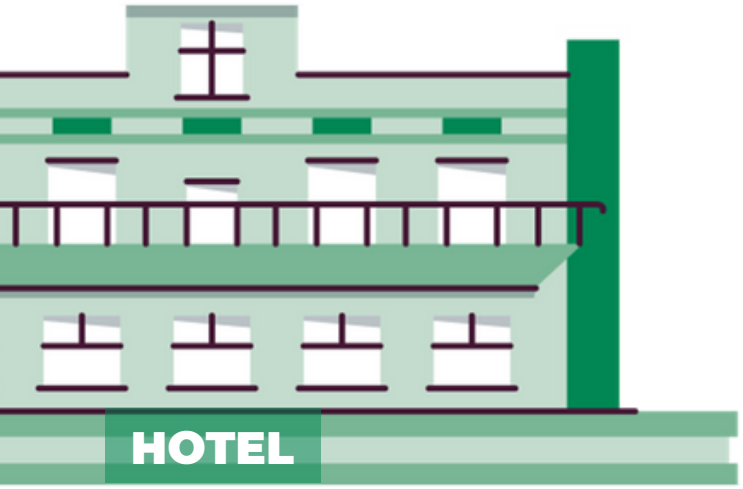
INSEA

PROJET ANALYSE DE DONNÉES

APPLICATION DE L'ACP SUR LES DONNÉES DES HOTELS



Réalisé par:

- EL GAOUT EL Mehdi
- ELAZZAOUI Mohamed

Encadré par :

- Mme. EL HANNOUN Wafae

2022-2023

Table des matières

Table des figures

1 Introduction

L'industrie de l'hôtellerie joue un rôle significatif dans l'économie du Maroc, avec le tourisme étant une des principales sources de revenu du pays. Ces dernières années, il y a eu une augmentation du nombre d'hôtels au Maroc, offrant aux voyageurs une large gamme d'options d'hébergement. Afin de mieux comprendre les facteurs qui influencent le succès de ces hôtels, il est important d'analyser les données sur les hôtels marocains et d'étudier les relations entre les variables qui leur sont associées.

Dans ce projet, nous utiliserons l'analyse en composantes principales (ACP) pour examiner les données sur les hôtels marocains. L'ACP est une technique statistique utilisée pour identifier la structure sous-jacente dans un jeu de données en réduisant le nombre de variables tout en capturant les informations les plus importantes. En appliquant l'ACP aux données sur les hôtels marocains, nous serons en mesure d'identifier les variables clés qui sont étroitement liées au succès de ces hôtels et de déterminer comment ces variables sont liées les unes aux autres.

2 Préparation des données

2.1 Introduction

La collecte et la préparation des données sont une étape cruciale dans tout projet de recherche, car elles impliquent la collecte et l'organisation des données qui seront utilisées pour l'analyse. Dans cette section, nous décrirons les méthodes que nous avons utilisées pour collecter et préparer les données pour notre étude sur les hôtels marocains.

2.2 Collection de données

Dans notre projet de recherche, nous n'avons pas été en mesure de trouver un jeu de données disponible qui répondait à nos besoins. En conséquence, nous avons décidé de collecter les données nous-mêmes en utilisant une technique appelée **Web Scraping** qui consiste à utiliser un programme ou un script pour envoyer des requêtes à un site Web et puis analyser la réponse afin d'extraire les données souhaitées.

Suite à nos recherches, nous avons trouvé le site "booking.com", qui est l'une des plus grandes agences de voyage en ligne et qui possède de nombreuses informations précieuses pour notre recherche. Ce site répertorie de nombreux hôtels dans le monde entier, et en particulier au Maroc, qui nous aideront à mener notre analyse.

FIGURE 2.1: le site web cible "www.booking.com" [1].

La prochaine étape consiste à examiner le site Web pour comprendre sa structure et les données que nous souhaitons extraire.

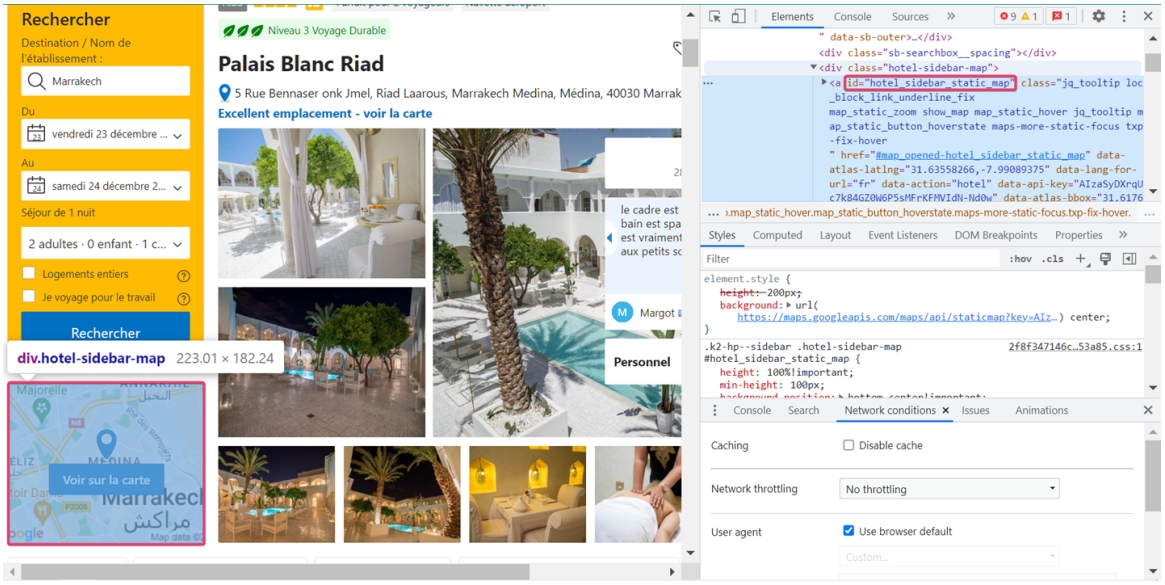


FIGURE 2.2: examination du site web

Voici les données que nous avons pu extraire :

- *Name* : Le nom de l'hôtel.
- *Coordinates* : Latitude et longitude de l'emplacement de l'hôtel sur la surface de la terre.
- *Reviews* : Évaluations de l'hôtel, par les clients.
- *Rating* : Évaluations de l'hôtel, souvent sur une échelle allant de 1 à 10.
- *Staff* : Personnel ou travailleurs de l'hôtel.

- $Facilities$: Installations ou ressources qui sont disponibles pour être utilisées à l'hôtel.
- $Cleanliness$: Propreté ou entretien de l'hôtel.
- $Comfort$: Confort ou convivialité de l'hôtel pour y séjourner.
- $Valueformoney$: Rapport qualité-prix ou justesse du prix de l'hôtel par rapport à la qualité ou à la quantité de ce qui est proposé.
- $Location$: Évaluations d'emplacement de l'hôtel.
- $Nightprice$: Prix par nuit.
- $WiFi$: Disponibilité d'un réseau sans fil qui permet aux appareils de se connecter à Internet à l'hôtel.

Le code utilisé pour extraire les données est découpé en plusieurs étapes :

Première étape

```
import requests
import pandas as pd
from bs4 import BeautifulSoup
```

Nous avons importé le module requests, qui nous a permis d'envoyer des requêtes HTTP en utilisant Python, ainsi que le module pandas pour la manipulation et l'analyse de données. Nous avons également importé le module re pour travailler avec les expressions régulières, et le module bs4 (Beautiful Soup 4)[3] pour l'analyse de documents HTML et XML.

Deuxième étape

```
#configuration de l'url du site web    partir duquel nous voulons
    r cup rer les donn es.
start_date = "2022-12-23"
end_date = "2022-12-24"
url = f"https://www.booking.com/searchresults.fr.html?ss=Maroc&ssne=Maroc&
    ssne_\
  untouched=Maroc&efdco=1&label=gen000nr-10
    CAIojAFCAm1hSA1YBGhNiAEBmAEzuAEFyAEe2A\
  ED6AEB-
    AEBiAIBqAIBuAK_1M6cBsACAdICJDdjNzllYWVmLWExZWMtNDM3Yy1iMDYyLTY4Y2EwMTVkNm
    \
```

```
  MyNNgCAeACAQ&sid=6b1171757eb9061e068c6ef46fd0af2f&aid=304142&lang=fr&sb=1&
   src_elem=\
  sb&src=searchresults&dest_id=143&dest_type=country&checkin={start_date}&
   checkout={end_date}\
  &group_adults=2&no_rooms=1&group_children=0&sb_travel_purpose=leisure"
#configuration des en-t tes de la requ te
headers = {
    "user-agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit
   /537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36"
}

#pagination
offsets=[0,25,50,75]
```

Ensuite, nous avons défini les valeurs de start_date et end_date sur des dates spécifiques de recherche, et avons utilisé ces variables pour construire une URL pour un site web (booking.com). Nous avons utilisé la syntaxe de chaîne f pour insérer les valeurs de start_date et end_date dans l'URL.

Nous avons également défini la valeur de headers, qui est un dictionnaire contenant les en-têtes HTTP qui sont envoyés avec la requête. Dans ce cas, nous avons défini l'en-tête "user-agent" sur une chaîne qui identifiait notre navigateur web et notre système d'exploitation. Enfin, nous avons défini la valeur de offsets sur une liste d'entiers que nous avons utilisés pour la pagination (c'est-à-dire pour récupérer plusieurs pages de données du site web).

Troisième étape

```
#envoi des requ tes
links=[]
for offset in offsets:
    response = requests.get(url+"&offset="+str(offset), headers=headers)
    #analyser le r sultat
    soup = BeautifulSoup(response.text, 'html.parser')
    #extraire les liens vers hotels
    for link in soup.find_all("a", class_="e13098a59f"):
        links.append(link['href'])

#suppression des donn es dupliqu es
links = list(set(links))
```

Ce code envoie des requêtes HTTP GET pour extraire les attributs href de chaque éléments hôtel de la réponse HTML. Ces valeurs sont stockées dans une liste, qui est convertie en ensemble pour supprimer les doublons.

Quatrième étape

```
hotels=[]
hotels_names=[]
for link in links:
    link_resp = requests.get(link, headers=headers)
    link_soup = BeautifulSoup(link_resp.text, 'html.parser')
    hotel={}
    hotel_name=link_soup.find("h2", class_="pp-header__title").text

    #extraction et formattage des donn es
    if hotel_name in hotels_names:
        continue
    hotels_names.append(hotel_name)
    hotel['name'] = hotel_name

    hotel_nb_photos = link_soup.find("span", class_="bh-photo-grid-thumb-
   more-inner-2")
    hotel['nb_photos'] = [int(s)+7 for s in hotel_nb_photos.text.split() if
   s.isdigit()][0]

    hotel_location = link_soup.find("a", id="hotel_sidebar_static_map")
    hotel_location = [float(s) for s in hotel_location['data-atlas-latlng'].
   split(',')]
    hotel["location_x"]= hotel_location[0]
    hotel["location_y"] = hotel_location[1]

    hotel_reviews = link_soup.find("div", class_="d8eab2cf7f c90c0a70d3
   db63693c62")
    hotel["reviews"] = [int(s) for s in hotel_reviews.text.split() if s.
   isdigit()][0]

    hotel["rating"] = float(link_soup.find("div", class_="b5cd09854e
   d10a6220b4").text.replace(",","."))
    hotel_cats_rating = link_soup.find_all("div", class_="ee746850b6
   b8eef6afe1")
```

```
    hotel["staff"] = float(hotel_cats_rating[0].text.replace(",","."))
    hotel["facilities"] = float(hotel_cats_rating[1].text.replace(",","."))
    hotel["cleanliness"] = float(hotel_cats_rating[2].text.replace(",","."))
    hotel["comfort"] = float(hotel_cats_rating[3].text.replace(",","."))
    hotel["money_value"] = float(hotel_cats_rating[4].text.replace(",","."))
    hotel["location"] = float(hotel_cats_rating[5].text.replace(",","."))
    hotel["wifi"] = float(hotel_cats_rating[6].text.replace(",","."))
    print(link_soup.find("span", class_="prco-valign-middle-helper"))
    hotel["night_price"] = float(link_soup.find("span", class_="prco-valign-
   middle-helper").text[4:].replace(" ",""))


    hotels.append(hotel)
```

À ce niveau, nous avons créé un tableau d'hôtels contenant les caractéristiques de chaque établissement. Ces caractéristiques ont été obtenues à l'aide de la fonction "$find$" de l'API $BeautifulSoup$, qui nous permet de spécifier soit le nom de classe soit l'identifiant d'élément HTML en tant que paramètre. Les données ont ensuite été nettoyées et mises en forme en supprimant tous les caractères inutiles et en les convertissant en types de données appropriés

Cinquième étape

```
df = pd.DataFrame(hotels)
df.to_csv('hotels.csv', index=False, header=True)
```

Finalement, un fichier nommé "hotels.csv" est créé en utilisant le $Dataframe$ de panda.

index	name	location_x	location_y	reviews	rating	staff	facilities	cleanliness	comfort	money_value	location	wifi	night_price
0	Riad Janate & SPA	31.63359	-7.986248	391	9.3	9.7	9.4	9.5	9.5	9.2	8.6	8.3	2842.0
1	La Ferme Ecolodge	30.60497169	-6.16689841	25	8.8	9.0	9.1	8.9	8.9	9.0	9.5	9.0	279.0
2	Riad Qodwa	31.52871748	-7.8693974	102	8.5	9.3	8.1	8.7	8.7	8.7	8.1	10.0	520.0
3	Airport Apartment Suite Casablanca FREE WIFI Modern Confort Calme	33.39254558	-7.55895853	60	9.8	9.9	9.9	9.8	9.9	9.8	9.5	7.5	652.0
4	Riad Raoud Rayhane	31.51387438	-9.77123111	247	9.0	9.3	9.2	9.4	9.4	9.0	9.6	10.0	656.0
5	Marrakech Ryads Parc All inclusive	31.68843559	-7.98714101	1	7.5	8.4	7.5	7.8	7.8	7.6	7.3	7.4	1535.0
6	Auberge Restaurant Telouet	31.28983317	-7.23663479	70	8.3	9.4	7.7	8.2	8.1	8.7	9.4	8.8	323.0
7	La Maison Jaune Dakhla Maison d'hôtes	23.7097408	-15.92297383	59	8.6	8.9	8.6	8.9	8.9	8.5	9.1	8.3	794.0
8	Riad Hayati	31.6220884	-7.9845188	46	9.5	9.9	9.4	9.8	9.6	9.2	9.9	6.7	1001.0
9	Hotel Leonor	35.22947204	-3.03163969	71	7.9	8.7	7.5	8.2	8.1	8.1	8.4	8.7	724.0
10	Auberge Kasbah Tiriguioute	31.099877	-6.557685	78	9.7	9.9	9.4	9.6	9.5	9.6	9.6	8.8	602.0
11	Kasbah Ait BenHadda	31.03852094	-6.57673359	346	9.3	9.8	9.1	9.3	9.1	9.4	8.9	9.0	524.0
12	RIAD LA SANTA	35.167445	-5.263292	238	8.8	9.4	8.6	9.0	9.0	9.0	9.6	7.5	802.0
13	Hyatt Place Taghazout Bay	30.52946788	-9.68710899	1	8.1	8.7	8.4	8.7	8.7	7.6	8.2	7.8	1802.0
14	Auberge Restaurant Atlas	31.5532455	-5.58526039	294	9.4	9.9	9.2	9.5	9.3	9.5	9.6	9.0	479.0
15	Royal Mirage Fes Hotel	34.03752884	-5.00397205	508	6.7	7.8	6.6	7.1	7.2	6.5	8.4	6.7	970.0
16	Villa Nour	31.45575744	-9.75165367	159	8.5	9.2	8.5	8.9	8.9	8.7	8.1	8.3	1163.0
17	Hotel Central Palace	31.62482589	-7.98835605	756	7.9	9.2	7.6	8.0	7.9	8.6	9.3	7.9	421.0
18	Riad Swaka	31.62151618	-7.98347712	256	9.0	9.7	8.8	9.2	9.1	8.9	9.2	8.8	1672.0
19	Hotel la renaissance tata	29.74264949	-7.97389009	15	8.1	9.3	7.7	8.5	8.3	8.2	8.5	9.3	401.0
20	Dawliz Rabat Art & Spa	34.02792637	-6.81295574	703	8.0	8.4	8.2	8.3	8.5	7.6	8.9	8.4	1715.0
21	HOTEL Bab Rimal	30.12872262	-6.86587572	63	8.3	8.9	8.3	8.1	8.5	7.9	8.7	6.7	892.0
22	HOTEL DES THERMES	33.388549	-6.094645	19	8.4	9.3	7.8	8.6	8.4	8.4	8.8	9.3	391.0
23	Kasbah Baha Baha	30.87128814	-5.86335182	94	9.2	9.3	9.1	9.3	9.3	9.3	9.2	5.0	446.0
24	Auberge la belle vue dades	31.50779391	-5.94239074	19	9.7	10.0	9.5	9.7	9.2	9.7	9.6	10.0	246.0

Show 25 per page 1 2 10 18

FIGURE 2.3: Le fichier htels.csv

3 Application d'ACP

3.1 Introduction

A l'époque, le mathématicien et statisticien Karl Pearson (Le père du test du χ^2) était principalement intéressé par la corrélation entre différentes variables et comment elle pouvait être utilisée pour prédire les valeurs d'une variable à partir d'une autre.[2]
En utilisant des techniques mathématiques avancées, Pearson a développé une méthode sous le nom **« Analyse en composantes principales »** pour représenter un grand nombre de variables sous forme de composantes principales, qui sont des variables linéairement non corrélées qui capturent la variabilité maximale des données d'origine. Cette méthode a permis de réduire la dimension des données de manière à les rendre plus simples et plus faciles à comprendre. Pearson a publié ses travaux sur l'ACP dans un article intitulé "On Lines and Planes of Closest Fit to Systems of Points in Space" en 1901, qui est considéré comme le premier article sur l'ACP. Sa méthode a rapidement été adoptée par d'autres scientifiques et est devenue un outil de base pour l'analyse de données dans de nombreux domaines.
Au fil des années, l'ACP a été développée et améliorée par de nombreux chercheurs, et elle est aujourd'hui largement utilisée dans de nombreux domaines pour explorer et comprendre de manière plus efficace de grands ensembles de données complexes.

3.2 Analyse en composantes principales :

Définition

L'ACP est un outil permettant d'analyser un ensemble de données complexes et de le réexprimer en termes plus simples.
La définition du manuel est : "une procédure statistique qui utilise une transformation orthogonale pour convertir un ensemble d'observations de variables éventuellement corrélées en un ensemble de valeurs de variables linéairement non corrélées appelées composantes principales".

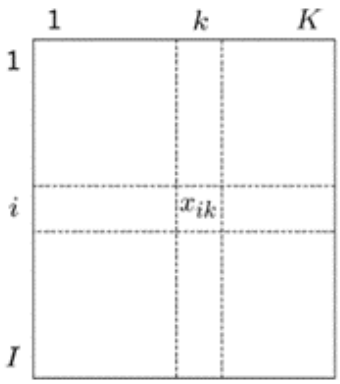
Comment faire l'ACP

La réalisation d'une Analyse en composantes principales nécessite les 4 étapes suivantes :

Standardiser de données

Tout d'abord, nous devons standardiser les données car cela garantit que toutes les fonctionnalités sont à la **même échelle**, ce qui est nécessaire pour que ACP fonctionne correctement

$$x_{ik} \rightarrow \frac{x_{ik}-\bar{X}_k}{S_k}$$



Avec :

x_{ik} : la valeur associée à l'individu i et la variable k

$\bar{X}_k = \frac{\sum_{i=1}^{n} x_{ik}}{n}$: La moyenne des valeurs dans la colonne (variable) k

$S_k = \sqrt{\frac{\sum_{i=1}^{n} (x_{ik}-\bar{X}_k)^2}{n-1}}$: L'écart-type des valeurs dans la colonne (variable) k

n : nombre de valeurs dans l'ensemble de données

Calculer la matrice de covariance

Pour séparer les variables fortement interdépendantes, on doit calculer la matrice de covariance. Une matrice de covariance est une matrice symétrique N x N qui contient les covariances de tous les ensembles de données possibles. Elle est donnée comme suit :

$$M = \begin{pmatrix} Cov(Y_1, Y_1) & \cdots & Cov(Y_1, Y_k) \\ \vdots & \ddots & \vdots \\ Cov(Y_k, Y_1) & \cdots & Cov(Y_k, Y_k) \end{pmatrix}$$

Avec :

$Y_1, \ldots, Y_k$: sont des variables

$Cov\left(Y_j, Y_p\right) = \overline{Y_j Y_p} - \overline{Y_j}\,\overline{Y_p}$: La covariance entre la variable Y_j et Y_p

Calculez le vecteur de caractéristiques

Pour déterminer le vecteur de caractéristiques, vous devez définir les valeurs propres et les vecteurs propres de la matrice de covariance.
Les vecteurs propres seront les composantes principales et les valeurs propres décriront la quantité de variance expliquée par chaque composante principale.
Tous simplement on choisit les composantes principales (vecteurs propres) qui sont associés à des variances (valeurs propres) grands où leurs sommes présente au moins 80%. On commence par les valeurs propres λ_i qui sont calculées par l'équation suivante :

$$|M - \lambda I| = 0$$

où M est la matrice de covariance et I la matrice Identité.

Ensuite, on calcule le vecteur propre ω_i associé à chaque valeur propre λ_i avec l'équation suivante :

$$M\lambda_i = M\omega_i$$

Enfin, nous trions en ordre décroissant les valeurs propres et leurs vecteurs propres correspondants, nos crions le vecteur de caractéristiques qui contient les premiers vecteurs propres qui la somme de leurs valeurs propres présente au moins 80% et nous ignorons le reste?

Multipliez les données normalisées par les vecteurs propres

L'objectif de l'ACP est de réexprimer l'ensemble de données d'origine, et maintenant nous sommes enfin prêts à franchir cette étape et à générer les "composantes principales" réelles. Nous multiplions simplement notre jeu de données standardisé à l'étape 1 par le vecteur de caractéristiques que nous avons générée à l'étape 3.

Exemple

Supposons que vous avez un ensemble de données qui contient les informations suivantes sur 3 personnes : le poids, la taille et l'âge.

Personne	**Poids**	**Taille**	**Age**
1	80	180	30
2	70	175	25
3	65	170	35

1. Standardisation de données :

Personne	Poids	Taille	Age
1	80	180	30
2	70	175	25
3	65	170	35
	$\bar{X}_k = 71.6$ $S_k = 7.6$	$\bar{X}_k = 175$ $S_k = 5$	$\bar{X}_k = 3$ $S_k = 5$

Personne	Poids	Taille	Age
1	1.09	1	0
2	-0.22	0	-1
3	-1.87	-1	1

2. Calculer la matrice de covariance :

$$M = \begin{pmatrix} var(Poids) & cov(Taille, Poids) & cov(Age, Poids) \\ cov(Poids, Taille) & var(Taille) & cov(Age, Taille) \\ cov(Poids, Age) & cov(Taille, Age) & var(Age) \end{pmatrix} = \begin{pmatrix} 1 & 0.98 & -0.32 \\ 0.98 & 1 & -0.5 \\ -0.32 & -0.5 & 1 \end{pmatrix}$$

3. Calculez le vecteur de caractéristiques :

$$\lambda_1 = 2.25 = \begin{pmatrix} -0.62 \\ -0.65 \\ 0.42 \end{pmatrix}, \lambda_2 = 0.74 = \begin{pmatrix} 0.40 \\ 0.19 \\ 0.89 \end{pmatrix}, \lambda_3 = 0.0004 = \begin{pmatrix} 0.66 \\ -0.72 \\ -0.14 \end{pmatrix}$$

Dans ce cas, la première composante principale (c'est-à-dire le premier vecteur propre) explique le plus de variance dans les données (75.24%) et la deuxième composante principale explique la variance suivante (24.75%). La troisième composante principale explique le moins de variance (0.01%) et peut être ignoré.

Le vecteur de caractéristiques sera : $\begin{pmatrix} -0.62 & -0.40 \\ -0.65 & 0.19 \\ 0.42 & 0.89 \end{pmatrix}$

4. Multipliez les données normalisées par les vecteurs propres

$$DonneesStandardise * VecteurCaracteristiques =$$

$$\begin{pmatrix} 1.09 & 1 & 0 \\ -0.22 & 0 & -1 \\ -1.87 & -1 & 1 \end{pmatrix} * \begin{pmatrix} -0.62 & -0.40 \\ -0.65 & 0.19 \\ 0.42 & 0.89 \end{pmatrix} = \begin{pmatrix} -3.94 & 0.86 \\ -0.83 & -1.34 \\ 3.16 & 0.69 \end{pmatrix}$$

Les données finales :

Personne	CP1	CP2
1	-3.94	0.86
2	-0.83	-1.34
3	3.16	0.69

Nous pouvons ensuite tracer les données transformées sur un nuage de points, avec la CP1 sur l'axe des x et la CP2 sur l'axe des y.

3.3 Implémentation

Dans ce chapitre, nous allons appliquer ce que nous avons appris dans la dernière partie de manière automatique en utilisant les langages de programmation R et Python, sur l'ensemble des données que nous avons collectées au premier chapitre.

3.3.1 Implémentation avec R

Chargement des données

Le code ci-dessous chargera simplement les données de notre ordinateur sous format csv :

```
setwd("/Chemin du dossier contenant le fichier csv/")
hotels.data <- read.csv(file="hotels.csv",header = 1, sep =",",row.names=1)
```

- **header** : *TRUE* si les noms des variables existent dans la première ligne
- **sep** : comment les colonnes dans le fichier csv sont séparées
- **row.names** : spécifier la colonne où trouvent les noms des individus, ils doivent être uniques

Nous pouvons voir son contenu avec la fonction $View(hotels.data)$:

	name	location_x	location_y	reviews	rating	staff	facilities	cleanliness	comfort	money_value	location	wifi	night_price
1	Dar Illi	31.51583	-7.489758	85	9.4	9.6	9.6	9.7	9.8	9.1	9.6	8.0	2478
2	La Claire Fontaine	31.63617	-7.990884	75	9.7	9.8	9.8	9.9	9.8	9.5	9.3	10.0	1634
3	Nubia Luxury Camp Erg Chegaga	29.87327	-6.138761	6	8.5	9.6	7.1	8.3	7.9	5.8	9.6	9.6	5104
4	Maison d'Hôtes Irocha	31.12625	-7.324855	157	9.0	9.5	9.0	9.3	9.1	9.0	8.8	7.5	374
5	Kasbah Hotel Camping Jurassique	32.15385	-4.375351	263	8.6	9.5	8.1	8.5	8.4	9.1	9.3	6.7	246
6	Luxury Tented Camp	31.08017	-4.013361	58	9.6	9.7	9.7	9.9	9.7	9.4	9.8	8.8	1229
7	Desert Berber Fire-Camp	31.11249	-4.004012	333	9.4	9.6	9.5	9.5	9.5	9.5	9.7	9.6	308
8	Camp Sahara Holidays	29.82780	-5.728240	151	9.7	9.8	9.6	9.8	9.7	9.9	9.8	9.9	334
9	Riad Kasbah des Roches	31.55003	-5.909586	66	8.7	9.9	8.0	9.0	8.6	9.1	9.2	9.9	389
10	Riad Africa	31.62034	-7.985569	325	8.2	9.3	8.2	8.5	8.5	8.4	8.8	8.5	1081
11	Aftas Trip	29.61125	-10.019711	202	8.8	9.5	8.7	9.0	9.1	8.6	8.8	6.5	751
12	RIAD MAROSKO	31.51237	-9.767399	199	8.8	9.6	8.9	9.2	9.2	9.0	9.3	9.6	1001
13	Toubkal Garden	31.14278	-7.902496	388	9.0	9.5	8.8	9.1	9.1	9.2	9.3	7.5	423
14	Auberge La Fibule Du Dades	31.51525	-5.933191	94	9.3	9.8	8.9	9.2	9.1	9.6	9.1	9.2	379
15	Kasba TanTan	28.49858	-11.333918	29	7.7	9.1	7.5	8.2	8.1	7.5	8.8	10.0	586
16	SOUKTANA	30.53127	-7.923879	52	7.2	8.3	6.8	7.1	7.0	7.5	7.7	6.3	167
17	RIAD DLIM TAMRAGHT	30.51635	-9.676674	31	9.2	9.9	9.3	9.6	9.4	9.4	7.7	9.9	585
18	Suite Azur Hotel	31.51471	-9.769387	509	9.0	9.4	9.0	9.5	9.5	8.8	9.4	8.1	3203
19	Hôtel Chark	34.22720	-3.344839	11	8.8	9.8	8.9	9.3	9.1	8.4	9.8	9.8	890
20	Authentique berber Camp	29.87927	-5.681945	53	9.9	10.0	9.4	9.7	9.7	10.0	10.0	10.0	400
21	Mhamid Camp Excursions	29.82805	-5.727206	66	9.5	9.4	9.5	9.4	9.4	9.5	9.4	9.3	300
22	Riad Rihana Dades	31.44900	-5.975961	92	9.2	9.9	9.2	9.4	9.2	9.3	9.5	9.9	721
23	Palm's Hotel Club	31.40968	-4.248997	60	8.0	8.6	7.5	7.9	7.9	8.1	8.4	5.0	979
24	Riad Sephora	31.57864	-5.586534	552	8.2	9.2	7.7	8.2	8.1	8.6	9.3	8.0	264
25	Riad Elias & Spa	31.62885	-7.983273	183	8.6	9.4	8.6	9.1	9.1	8.9	9.0	10.0	1261
26	Magnifique Penthouse 140m2 de terrasse vue mer	33.52339	-7.842189	17	9.4	9.4	9.3	9.4	9.5	9.0	9.7	9.4	1335
27	Kasbah Mohayut	31.13110	-4.017037	402	9.2	9.4	9.1	9.2	9.2	9.3	9.5	7.5	601
28	Puerta Azul	35.16437	-5.262893	761	9.4	9.8	9.2	9.5	9.4	9.4	9.0	8.9	901
29	Gite Elkhorbat	31.49324	-5.088215	88	9.1	9.4	8.8	9.3	9.1	9.3	9.0	5.0	656
30	Hôtel les truites	33.73523	-5.007234	24	7.8	9.0	7.9	9.0	8.8	8.1	8.5	9.0	350

Application d'ACP

Quelques sources disent qu'avant d'appliquer l'ACP, il est nécessaire de vérifier si la moyenne de la matrice de corrélation est supérieure à 0.3 ou inférieur à -0.3 pour que les résultats soient bons Pour obtenir la matrice de corrélation :

```
Core_matrix = cor(hotels.data[3:14])
```

```
> cor(hotels.data[2:13])
              location_x  location_y     reviews      rating       staff facilities cleanliness    comfort
location_x   1.000000000  0.44364501  0.19155111 -0.01728862 -0.04394342 0.06984155  0.08084079 0.07385913
location_y   0.443645011  1.00000000 -0.05478799  0.26625980  0.22514980 0.19207875  0.19234136 0.14334050
reviews      0.191551105 -0.05478799  1.00000000  0.11381130  0.05891738 0.15366090  0.13005342 0.17854848
rating      -0.017288624  0.26625980  0.11381130  1.00000000  0.86984831 0.93492182  0.92410784 0.90136522
staff       -0.043943418  0.22514980  0.05891738  0.86984831  1.00000000 0.74508792  0.76479854 0.70232960
facilities   0.069841552  0.19207875  0.15366090  0.93492182  0.74508792 1.00000000  0.95605743 0.96188708
cleanliness  0.080840785  0.19234136  0.13005342  0.92410784  0.76479854 0.95605743  1.00000000 0.96653141
comfort      0.073859127  0.14334050  0.17854848  0.90136522  0.70232960 0.96188708  0.96653141 1.00000000
money_value -0.018893807  0.22121034  0.16216537  0.87896537  0.77959122 0.83961907  0.81164256 0.80353270
location    -0.038215857  0.31220794  0.03848067  0.73061146  0.59231058 0.65934347  0.64182743 0.65871882
wifi         0.075315144  0.15407902 -0.08752784  0.31604192  0.38438076 0.24516997  0.28173129 0.24709522
night_price -0.002301399 -0.19489951  0.04423740  0.12547816  0.07892185 0.11399913  0.15027200 0.17168071
            money_value    location        wifi  night_price
location_x  -0.01889381 -0.03821586  0.07531514 -0.002301399
location_y   0.22121034  0.31220794  0.15407902 -0.194899511
reviews      0.16216537  0.03848067 -0.08752784  0.044237399
rating       0.87896537  0.73061146  0.31604192  0.125478159
staff        0.77959122  0.59231058  0.38438076  0.078921854
facilities   0.83961907  0.65934347  0.24516997  0.113999133
cleanliness  0.81164256  0.64182743  0.28173129  0.150272003
comfort      0.80353270  0.65871882  0.24709522  0.171680706
money_value  1.00000000  0.59169258  0.24250470 -0.157722187
location     0.59169258  1.00000000  0.30948591  0.153191122
wifi         0.24250470  0.30948591  1.00000000  0.108027903
night_price -0.15772219  0.15319112  0.10802790  1.000000000
>
```

Et on calcule sa moyenne avec la fonction mean

```
mean(Core_matrix)
Output: 0.3965109
```

Le résultat est égal à 0.4> 0.3, alors on peut calculer l'ACP sur ces données

Maintenant, on applique l'ACP avec une fonction prédéfinie dans la bibliothèque *FactoMiner*.

```
install.packages("FactoMiner")
library("FactoMiner")
hotels.acp <- PCA(hotels.data[2:13])
```

FIGURE 3.1: La commande PCA

La fonction PCA prend plusieurs paramètres :

- **Ncp** : nombre de CP conservées dans les résultats (par défaut 5)
- **Scale.unit** : un booléen, si *TRUE* (valeur définie par défaut) alors les données seront normalisées

- **ind.sup** : un vecteur indiquant les indices des individus supplémentaires
- **quanti.sup** : un vecteur indiquant les indices des variables supplémentaires quantitatives
- **quali.sup** : un vecteur indiquant les indices des variables supplémentaires qualitatives

On peut afficher le résultat de l'interprétation de l'ACP utilisant la fonction summary(hôtels.acp)

```
Call:
PCA(X = hotels.data[c(2:13)], ncp = 4, graph = F)


Eigenvalues
                       Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7   Dim.8   Dim.9  Dim.10  Dim.11
Variance               6.255   1.627   1.258   0.876   0.753   0.537   0.294   0.207   0.093   0.049   0.030
% of var.             52.122  13.558  10.480   7.303   6.275   4.474   2.450   1.723   0.777   0.411   0.248
Cumulative % of var.  52.122  65.680  76.160  83.463  89.737  94.212  96.661  98.384  99.161  99.571  99.819
                      Dim.12
Variance               0.022
% of var.              0.181
Cumulative % of var. 100.000

Individuals (the 10 first)
               Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
1          |  4.045 |  1.971  0.144  0.237 | -0.700  0.070  0.030 |  3.027  1.694  0.560 |
2          |  2.034 |  0.905  0.030  0.198 | -0.172  0.004  0.007 | -1.315  0.320  0.418 |
3          |  1.974 | -0.369  0.005  0.035 | -0.524  0.039  0.071 | -0.701  0.091  0.126 |
4          |  3.749 |  3.237  0.390  0.746 |  0.384  0.021  0.010 | -0.352  0.023  0.009 |
5          |  2.605 |  1.770  0.116  0.462 | -0.921  0.121  0.125 |  0.169  0.005  0.004 |
6          |  4.297 | -3.636  0.492  0.716 | -0.827  0.098  0.037 |  0.395  0.029  0.008 |
7          |  2.089 | -0.744  0.021  0.127 | -0.300  0.013  0.021 | -1.359  0.342  0.424 |
8          |  6.182 |  0.022  0.000  0.000 | -6.039  5.212  0.954 | -0.273  0.014  0.002 |
9          |  3.314 |  2.593  0.250  0.612 | -0.479  0.033  0.021 | -0.075  0.001  0.001 |
10         |  3.531 | -2.105  0.165  0.356 |  2.395  0.820  0.460 | -0.645  0.077  0.033 |

Variables (the 10 first)
              Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3    ctr   cos2
location_x  | -0.152  0.370  0.023 |  0.894 49.092  0.799 |  0.173  2.367  0.030 |
location_y  |  0.149  0.353  0.022 |  0.867 46.254  0.753 | -0.195  3.038  0.038 |
reviews     | -0.125  0.251  0.016 |  0.212  2.767  0.045 |  0.665 35.183  0.442 |
rating      |  0.984 15.469  0.968 | -0.010  0.006  0.000 | -0.014  0.016  0.000 |
staff       |  0.905 13.109  0.820 | -0.026  0.040  0.001 | -0.100  0.789  0.010 |
facilities  |  0.957 14.636  0.915 |  0.025  0.038  0.001 |  0.112  1.006  0.013 |
cleanliness |  0.959 14.707  0.920 |  0.011  0.008  0.000 |  0.098  0.770  0.010 |
comfort     |  0.954 14.543  0.910 |  0.000  0.000  0.000 |  0.144  1.644  0.021 |
money_value |  0.911 13.272  0.830 |  0.019  0.021  0.000 | -0.192  2.931  0.037 |
location    |  0.720  8.283  0.518 |  0.065  0.259  0.004 | -0.042  0.142  0.002 |
```

Choisir le nombre de composante principale

Comme nous voyons dans le résultat de la fonction « *summary* » sur l'ACP, ils nous donnent tous les valeurs propres et leurs vecteurs propres, ça va nous aider à choisir le nombre de composantes principales qu'on a besoin pour exprimer nos données.

```
Eigenvalues
                       Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7   Dim.8   Dim.9  Dim.10  Dim.11
Variance               6.255   1.627   1.258   0.876   0.753   0.537   0.294   0.207   0.093   0.049   0.030
% of var.             52.122  13.558  10.480   7.303   6.275   4.474   2.450   1.723   0.777   0.411   0.248
Cumulative % of var.  52.122  65.680  76.160  83.463  89.737  94.212  96.661  98.384  99.161  99.571  99.819
                      Dim.12
Variance               0.022
% of var.              0.181
Cumulative % of var. 100.000
```

Nous sommes besoin d'au moins 80% d'informations après la réduction des dimensions. Alors on commence à calculer le pourcentage « % of var » cumulé commençant de la première dimen-

sion « Dim.1 » jusqu'avoir une valeur supérieure ou égale à 80%. Enfin, on prend l'ordre de la dimension final et on le donne à la fonction PCA pour calculer l'ACP d'une façon correcte. Dans notre cas, on a achevé 83.32% à la 4éme dimension, alors on relance l'ACP en spécifiant le nombre de composantes en 4. Le code est comme suit :

3.3.2 Implémentation avec Python

Dans cette section, nous avons également implémenté l'ACP en utilisant python pour comprendre comment elle fonctionne, nous avons suivi les mêmes étapes que celles mentionnées précédemment.

Le script avec python

Voici le code commenté que nous avons écrit :

```
import numpy as np

class PCA():
    """
    Classe qui impl mente une analyse en composantes principales (PCA) sur
   un ensemble de donn es.
    """

    def __init__(self, X, num_components=5):
        """
        Initialise l'objet PCA avec un ensemble de donn es et le nombre de
   composantes principales    conserver.

        Parameters:
        X (numpy array) : un tableau numpy contenant les donn es
   analyser
        num_components (int) : le nombre de composantes principales
   conserver (par d faut 5)
```

```
        """

        # Normalisation des donn es
        mean = np.mean(X, axis=0)  # moyenne de chaque colonne
        scale = np.std(X, axis=0)  #  cart -type de chaque colonne
        X = (X - mean) / scale  # normalisation des donn es en soustractant
 la moyenne et en divisant par l' cart -type

        # Calcul de la matrice de covariance
        self.covariance_matrix = np.cov(X, rowvar=False)

        # Calcul des vecteurs et valeurs propres de la matrice de covariance
        eigenvalues, eigenvectors = np.linalg.eig(self.covariance_matrix)

        # Tri des valeurs propres et vecteurs propres par ordre d croissant
        sorted_indices = np.argsort(eigenvalues)[::-1]
        self.eigenvalues = eigenvalues[sorted_indices]
        eigenvectors = eigenvectors[:, sorted_indices]

        # Si le nombre de composantes n'est pas sp cifi  , on utilise tous
les vecteurs propres
        if num_components is None:
            num_components = X.shape[1]

        # S lection des num_components vecteurs propres les plus importants
        eigenvectors = eigenvectors[:, :num_components]

        # Calcul du pourcentage de variance expliqu e par chaque valeur
propre
        self.explained_variance_percent = [i/np.sum(eigenvalues)*100 for i
in eigenvalues[:num_components]]

        # Calcul de la variance cumul e expliqu e par chaque valeur propre
        self.cumu_explained_variance = np.cumsum(self.
explained_variance_percent)

        # Projection des donn es sur l'espace des composantes principales
        # X_transformed = np.dot(X, eigenvectors)

    def get_all(self):
```

```
        """
        Affiche les informations sur les valeurs propres et leur importance
    en termes de variance expliqu e.
        """
        print("         eigenvalue | percentage of variance | cumulative
    percentage of variance")
        for i in enumerate(zip(self.eigen
```

La méthode __init__ est le constructeur de la classe, c'est-à-dire qu'elle est appelée lorsqu'on crée un objet de cette classe. Elle prend en entrée les données à analyser (X) et le nombre de composantes principales à conserver (num_components). Par défaut, le nombre de composantes principales est défini à 5.

La méthode commence par normaliser les données en soustrayant la moyenne de chaque colonne et en divisant par l'écart-type de chaque colonne. Cette étape est importante, car elle permet de mettre toutes les variables sur la même échelle, ce qui est nécessaire pour l'ACP.

Ensuite, la matrice de covariance des données est calculée. La matrice de covariance est une mesure de la relation entre les différentes variables d'un ensemble de données. Elle est utilisée dans la PCA pour trouver les directions de plus grande variance dans les données.

Après avoir calculé la matrice de covariance, le code calcule les valeurs propres et les vecteurs propres de cette matrice en utilisant la fonction $np.linalg.eig$. Les valeurs propres et les vecteurs propres sont des propriétés mathématiques qui permettent de décomposer la matrice de covariance en une forme diagonale. Les valeurs propres sont les coefficients sur la diagonale de cette forme diagonale et les vecteurs propres sont les vecteurs qui permettent de passer de la base originale des données à la base des composantes principales.

Le code trie ensuite les valeurs propres et les vecteurs propres par ordre décroissant en utilisant la fonction $np.argsort$. Cela permet de sélectionner les $num\_components$ vecteurs propres les plus importants.

Le pourcentage de variance expliquée par chaque valeur propre est ensuite calculé en utilisant une liste compréhension. La variance cumulée expliquée par chaque valeur propre est calculée en utilisant la fonction $np.cumsum$.

Enfin, la méthode get_all affiche les informations sur les valeurs propres et leur importance.

3.3.3 Comparaison des résultats

Voici une comparaison entre l'application PCA avec $FactomineR$ et notre implémentation avec python.

```
> res.pca$eig
        eigenvalue percentage of variance cumulative percentage of variance
comp 1  6.25468580             52.1223817                           52.12238
comp 2  1.62692107             13.5576756                           65.68006
comp 3  1.25756564             10.4797137                           76.15977
comp 4  0.87633189              7.3027658                           83.46254
comp 5  0.75297908              6.2748257                           89.73736
comp 6  0.53690999              4.4742499                           94.21161
comp 7  0.29398070              2.4498392                           96.66145
comp 8  0.20672786              1.7227322                           98.38418
comp 9  0.09320591              0.7767159                           99.16090
comp 10 0.04926347              0.4105289                           99.57143
comp 11 0.02975857              0.2479881                           99.81942
comp 12 0.02167001              0.1805834                          100.00000
```

FIGURE 3.2: Résultat du PCA avec la commande PCA du package FactomineR

```
res = PCA(X, num_components=5)
res.get_eigenvalues()
```

```
          eigenvalue | percentage of variance | cumulative percentage of variance
Comp 1 | 6.2692654853393055 | 52.12238165136752  | 52.12238165136752
Comp 2 | 1.6307134235122247 | 13.55767555594467  | 65.68005720731219
Comp 3 | 1.2604970330370064 | 10.479713704900702  | 76.15977091221289
Comp 4 | 0.8783746235956146 | 7.302765765940291  | 83.46253667815319
Comp 5 | 0.7547342758930516 | 6.274825665855027  | 89.73736234400822
```

FIGURE 3.3: Résultat du PCA avec Python

Après avoir effectué la comparaison, nous avons constaté que les résultats étaient presque identiques. Cela indique que notre mise en œuvre a réussi et que le résultat produit par notre méthode correspond étroitement aux résultats attendus.

L'implémentation nous a vraiment aidés à comprendre de manière pratique comment fonctionne le PCA.

4 Visualisation et Interprétation

4.1 Introduction

Dans ce chapitre, nous allons explorer la relation entre différents facteurs qui influencent les hôtels marocains à l'aide de techniques de visualisation et d'interprétation. Nous utiliserons une variété de techniques graphiques.

En visualisant et en interprétant les données, nous pourrons identifier les principaux facteurs qui ont le plus grand impact sur les hôtels marocains et obtenir des insights sur la manière dont ces facteurs interagissent entre eux.

4.2 Quantité d'informations expliquée par chaque CP

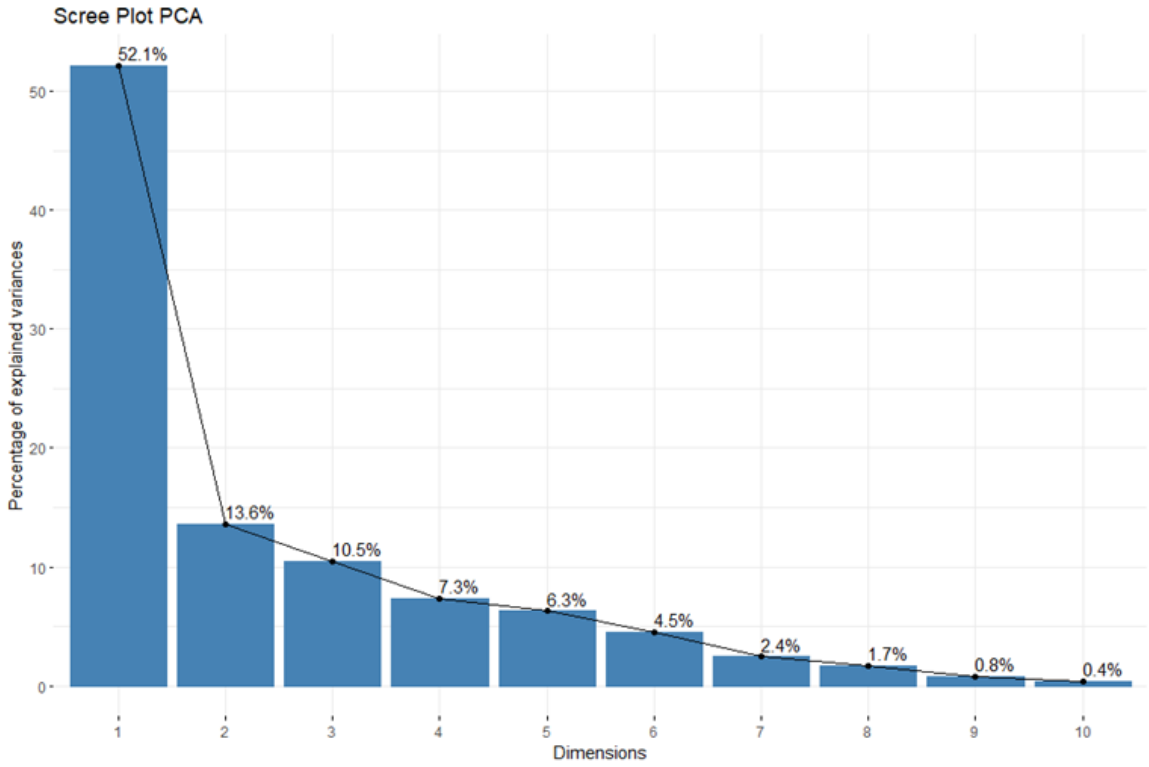


FIGURE 4.1: Screeplot

```
fviz_screeplot(hotels.acp, type="lines", addlabels=T, main="Scree Plot PCA")
```

fviz_screeplot est une fonction de la bibliothèque « *factoextra* » de R qui permet de créer un graphique en escalier, à partir des résultats de l'analyse des composantes principales (ACP).

Scree plot est un outil couramment utilisé pour visualiser et interpréter les résultats de l'ACP. Il représente graphiquement la variance expliquée par chaque composant principal, en ordonnant les composants principaux par ordre décroissant de variance expliquée. Il peut être utilisé pour déterminer le nombre de composants principaux à conserver pour l'analyse, en observant le point où la variance expliquée commence à diminuer de manière significative.

Pour les données que nous avons utilisées, nous notons que le premier facteur est prépondérant : il explique à lui seul 52.12% de la variabilité totale des données. Il convient de noter que dans un tel cas, la variabilité liée aux autres composantes peut être dénuée de sens, en dépit d'un pourcentage élevé.

4.3 Contribution des variables/individues dans chaque CP

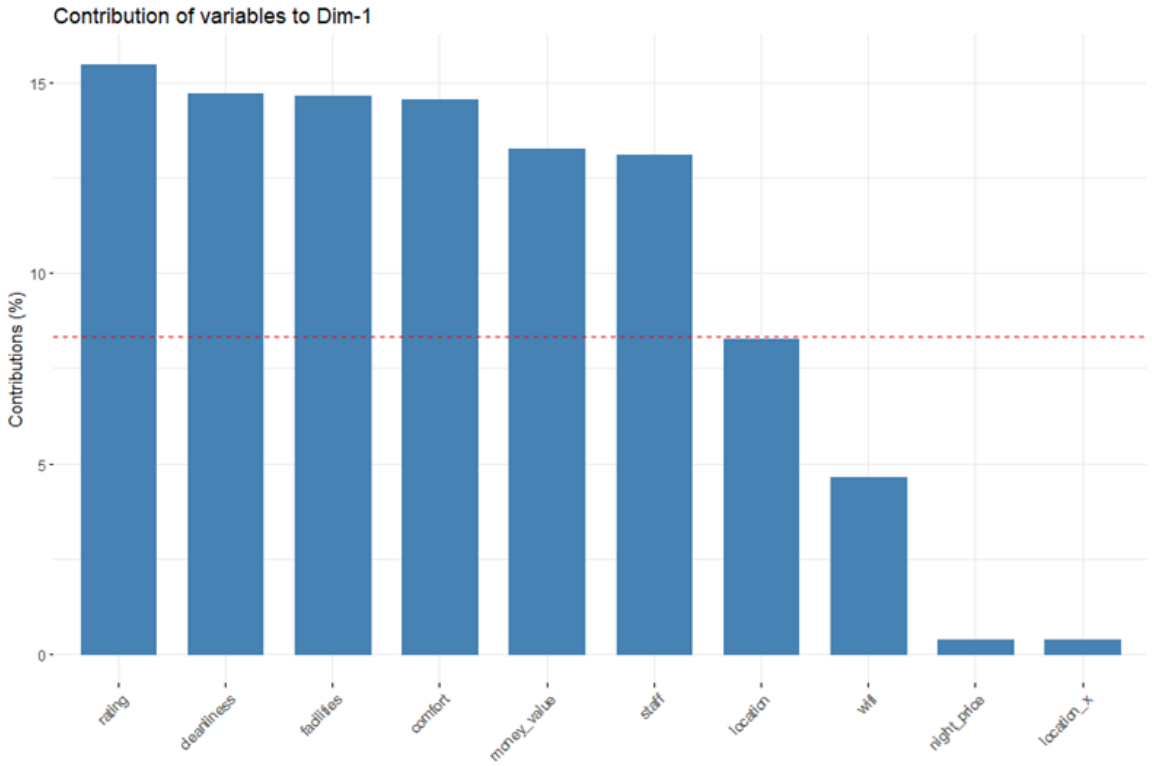


FIGURE 4.2: Contribution des variables à la dimension 1

```r
fviz_contrib(hotels.acp, choice = "var", axes = 1, top = 10)
```

$fviz\_contrib$ avec le paramètre *choice="var"* est une fonction du package factoextra en R génère un graphique en barres qui montre les contributions de chaque variable dans chaque composante principale, ce qui peut être utile pour comprendre les caractéristiques qui déterminent la variance dans les données. Vous pouvez également utiliser l'option $choice = "ind"$ pour visualiser les contributions des individus plutôt que les contributions des variables. Et vous pouvez spécifier la CP en utilisant le paramètre axes.

Nous pouvons résumer les contributions sur tous les axes sur un seul graphique avec la fonction $corrplot$:

```
corrplot(hotels.acp$var$contrib, is.corr=F,cl.ratio = 0.5)
```

En analysant le graphique, on constate que les principaux contributeurs au première CP sont $rating, staff, facilities, cleanliness, comfort, money\_value$ et $location$. Pour le deuxième axe, il y a les deux variables $location\_x$ et $location\_y$.

Enfin, dans le troisième et le quatrième axe ; les variables $reviews$ et $night\_price$ contribuent de manière significative par rapport au reste.

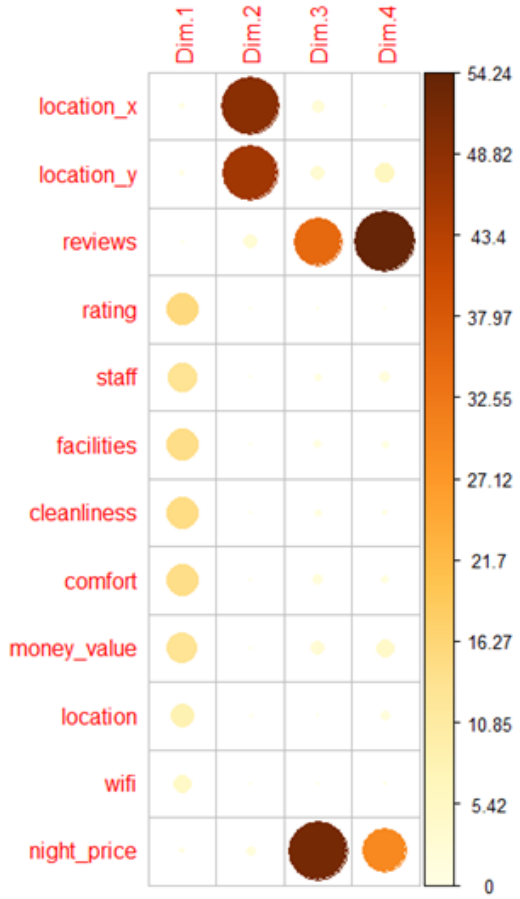


FIGURE 4.3: Corrélogramme

4.4 Interprétation par Biplot

Dans un $biplot$, les variables sont représentées par des vecteurs et les individus par des points. La longueur et la direction des vecteurs représentent la force et la direction des relations entre les

variables, tandis que la position des points par rapport aux vecteurs représente les relations entre les individus et les variables.

Pour afficher ce graphique on utilise la commande suivante :

```
fviz_pca_biplot(res.pca,
                geom.ind = "point",
                axes=c(1, 2),
                pointshape = 21, pointsize = 1,
                # les variables
                alpha.var ="contrib", col.var = "cos2",
                gradient.cols = c("#00AFBB","#E7B800","#FC4E07"),
                repel = TRUE
```

FIGURE 4.4: Commande pour afficher le *biplot*

Cette commande nous permet également d'obtenir des informations sur la contribution des variables à la formation des composants principales et sur leur qualité de représentation.

Le résultat :

FIGURE 4.5: Biplot de la dimension 1 et 2

En constate d'abord que certains variables ont une faible qualité de représentation comme le *night_price* et *reviews* donc il est difficile d'interpréter ces relations.

La dimension 1 oppose des individus caractérisés par une coordonnée fortement positive sur l'axe (à droite du graphe) à des individus caractérisés par une coordonnée fortement négative sur l'axe (à gauche du graphe).

1. **Le groupe 1** (caractérisés par une coordonnée positive sur l'axe) partage : de fortes valeurs pour les variables $rating$, $cleanliness$, $facilities$, $comfort$, $staff$, $money\_value$, $location$, $wifi$. Et de faibles valeurs pour les variables $location\_x$ et $reviews$.

2. **Le deuxième groupe** partage : de faibles valeurs pour les variables $cleanliness$, $rating$, $comfort$, $staff$, $facilities$, $money\_value$, $wifi$, $location\_y$.

Notons aussi que les variables $rating$, $facilities$, $cleanliness$ et $comfort$ sont extrêmement corrélées à cette dimension.

La dimension 2 oppose des individus caractérisés par une coordonnée fortement positive sur l'axe (en haut du graphe) à des individus caractérisés par une coordonnée fortement négative sur l'axe (en bas du graphe).

1. **Le groupe 1** (caractérisés par une coordonnée positive sur l'axe) partage : de fortes valeurs pour les variables $location\_x$, $location\_y$ et $reviews$.

2. **Le groupe 2** (caractérisés par une coordonnées négative sur l'axe) partage : de faibles valeurs pour les variables $cleanliness$, $rating$, $comfort$, $staff$, $facilities$, $money\_value$, $location$, $wifi$, $location\_y$ et $night\_price$ (de la plus extrême à la moins extrême).

3. **Le groupe 3** (caractérisés par une coordonnées négative sur l'axe) partage : de fortes valeurs pour les variables $rating$, $cleanliness$, $facilities$, $comfort$, $staff$, $money\_value$, $location$, $wifi$ et $night\_price$ (de la plus extrême à la moins extrême). de faibles valeurs pour les variables $location\_x$ et $reviews$ (de la plus extrême à la moins extrême).

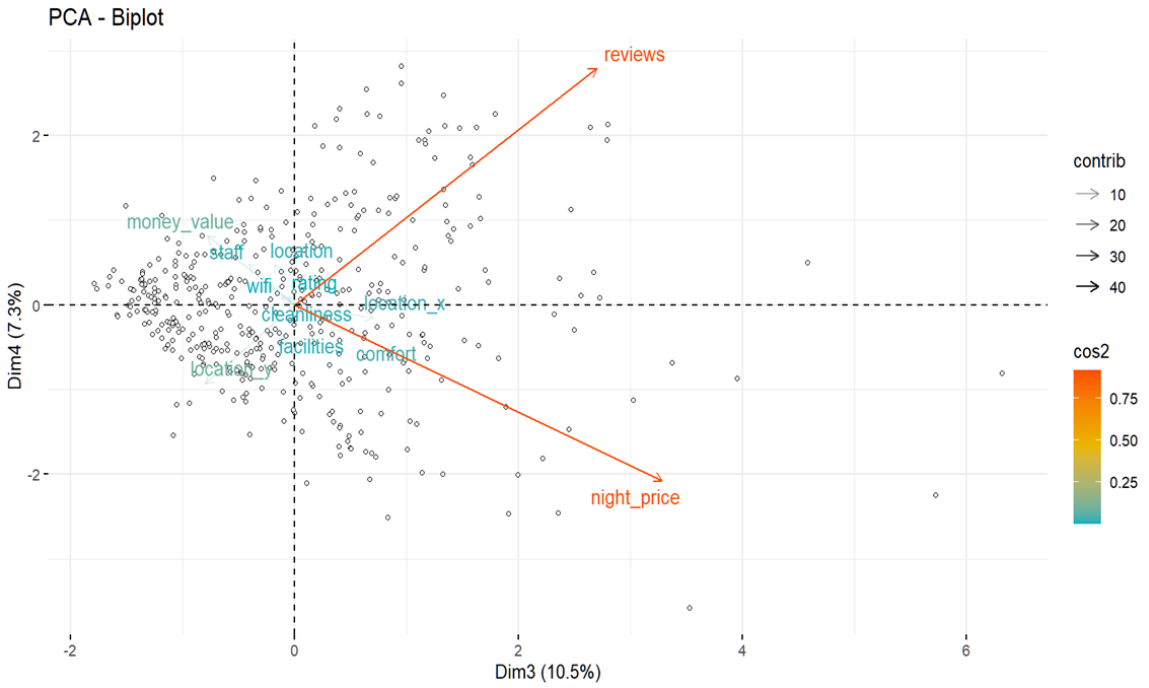


FIGURE 4.6: Biplot de la dimension 3 et 4

Pour les variables $night\_price$ et $reviews$, même si elles sont bien représentées sur le plan formé par les dimensions 3 et 4, elles ne sont presque pas corrélées entre elles, donc dans ce cas nous avons une idée des facteurs qui influencent la variable $night\_price$ qui est le plus intéressant.

4.5 Quels sont les facteurs qui influencent le prix des hôtels?

Comme le prix par nuit des hôtels est un facteur très intéressant, Les résultats de l'analyse PCA n'expliquent pas clairement les raisons des prix élevés des hôtels, car la variable $price\_night$ n'est pas bien représentée et ne présente pas de corrélation avec les autres variables dans le $biplot$. Cela ne signifie pas que la méthode n'est pas utile ou qu'une analyse plus approfondie ne serait pas utile. Cela signifie simplement que, dans ce cas particulier, la méthode n'a pas fourni une réponse claire à la question étudiée. Il est possible que d'autres techniques ou approches soient plus efficaces pour comprendre les facteurs qui influencent les prix des hôtels, ou qu'un ensemble de données plus complet ou un contexte supplémentaire soit nécessaire pour comprendre pleinement cette question.

5 CONCLUSION

En général, l'analyse en composantes principales (ACP) est un outil efficace analyser des données. Elle permet d'identifier les variables et les observations les plus importantes dans le jeu de données. Dans ce travail, nous avons étudié les étapes nécessaires pour utiliser l'ACP, y compris la collecte de données, l'implémentation avec R et Python, la visualisation et l'interprétation des résultats. Grâce à la visualisation et l'interprétation, nous avons pu mieux comprendre les données et en tirer des conclusions significatives. L'ACP est une technique précieuse qui peut être appliquée à de nombreux ensembles de données et a de nombreuses applications dans les domaines de la statistique, de l'apprentissage automatique et de la science des données.

Bibliographie

[1] BOOKING.COM. (s. d.). *The largest selection of hotels, homes, and vacation rentals*. https://www.booking.com/

[2] INC, T. E. (s. d.). *Step-By-Step Guide to Principal Component Analysis With Example*. https://www.turing.com/kb/guide-to-principal-component-analysis

[3] RICHARDSON, L. (2007). Beautiful soup documentation. *April*.