www.universite-paris-saclay.fr

ANNEE UNIVERSITAIRE 2015/2016

Master II

Mention : Innovation, Entreprise, Société (IES)

Spécialité : Innovation, marché et science des données (IMSD)

Mise en place d'un modèle prédictif du nombre de personnes dans le parc de Disneyland Paris

Projet réalisé par :

Asma Ben Amor/Hager Oueslati
Amadou Dioulde Balde/Mourad Karoui
Marcel Elia Rahari

Sous la direction de : M.Vincent Chalmel

Table des matières

Introduction

Disneyland Paris a développé un Système de Management de la Sécurité, visant à assurer la sécurité des visiteurs, des personnes, du personnel, du patrimoine et de l'environnement. Ce système repose sur plusieurs étapes : le diagnostic initial et permanent, résultat de l'identification et de l'évaluation de risques inhérents à une infrastructure ou à une activité. Le plan d'action d'amélioration par la prévention et par la protection. La mesure des performances et le contrôle, réalisés sous formes d'audits, diligentés par les équipes de Disneyland Paris et par des experts agréés. L'amélioration continue par la révision permanente de diagnostics et des plans d'action selon les enseignements tirés de l'analyse des performances et des contrôles. L'efficacité de ce Système de Management de la Sécurité et de la satisfaction provient de son adaptabilité et de son objectif d'amélioration constante.

De ce fait, et afin de garantir la satisfaction des clients, le service des data scientistes au sein de Disneyland Paris nous a proposé d'établir un modèle de prévision des réservations hôtelières de Disneyland Paris, ce qui nous a ramenés à établir une deuxième problématique qui n'est que la prédiction du nombre de personnes dans le parc de Disney Land Paris en fonction de plusieurs critères comme le nombre de personnes dans les hôtels, la météo…

Afin de pouvoir construire ce modèle, on a eu recours à un historique journalier datant de 2013. Ce modèle permettra à Disneyland Paris d'optimiser son système de support sur son site et de prévoir certaines actions pour garantir la satisfaction des visiteurs.

I. Traitements préliminaires des données

Nous disposons de plusieurs fichiers de données contenants des informations sur la situation des hôtels (nombre de réservations, d'admissions, party mix[1], durée de séjour…) et les parcs (fréquentation, horaires d'ouverture…) de Disneyland Paris pour l'année 2015.

Ces fichiers sont à l'état brut, ils contiennent beaucoup de doublons, d'informations à filtrer. Nous avons d'abord filtré nos données en amont avec SAS en supprimant toutes les répétitions, puis nous les avons agrégées pour ne garder qu'une ligne par jour, avec l'option tableau croisé dynamique d'Excel, avant de les joindre à d'autres bases en vue de les enrichir.

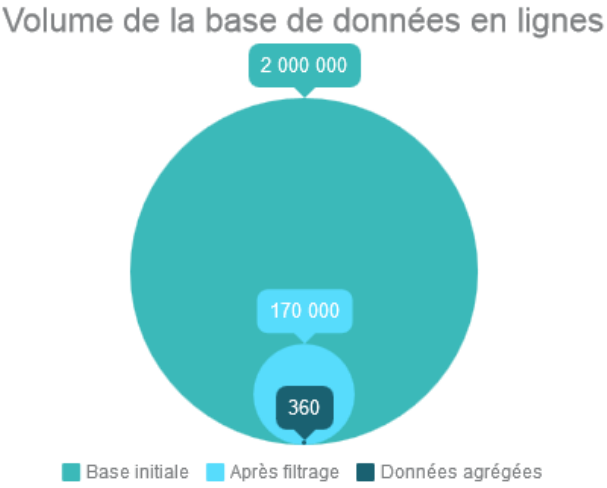


Figure 1:Evolution de la taille de la base

1. Traitement des jours de fermeture des parcs

L'année 2015 a la particularité d'avoir connu la fermeture des deux parcs de Disneyland Paris du 14 au 17 novembre suite aux attentats survenus à Paris, les valeurs d'attendance pour ces jours sont donc nulles et pourraient conduire à des erreurs dans notre modèle. Pour y remédier, nous avons décidé de considérer ces valeurs comme manquantes et de les imputer. Nous avons donc remplacé les attendances du 14 et 15 novembre par la moyenne du weekend précédent, car ces jours correspondent à un weekend, et celles du 16 et 17 novembre par la moyenne du 13 et 18 novembre.

[1] Nombre d'adultes et d'enfants

2. Filtre des variables qui ne varient pas trop

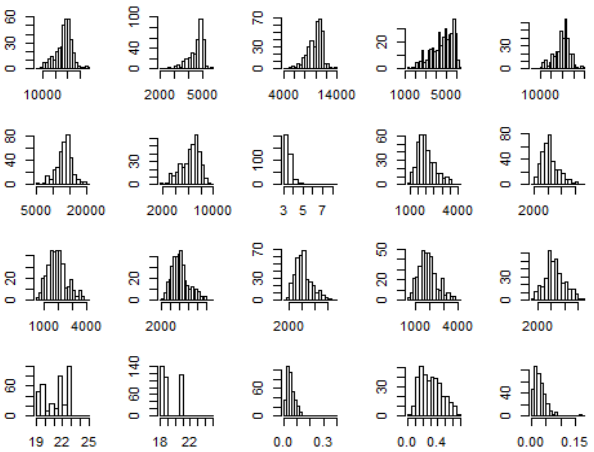
Les variables ayant une faible variance ont en général un pouvoir prédictif très faible et posent problème à la plupart des modèles. D'un côté elles induisent des soucis de multicolinérité avec la constante du modèle. De l'autre, dans une logique de validation croisée, on risque de trouver que dans certains, voire tous, de nos échantillons ces variables sont constantes. Il faut donc supprimer ce genre de variables des données avant l'estimation. Pour ce faire, nous utilisons la fonction **nearZeroVar** du package *caret* qui donne toutes les variables à variance très faible de la base. Après traitement, sept variables ont été supprimées de la base.

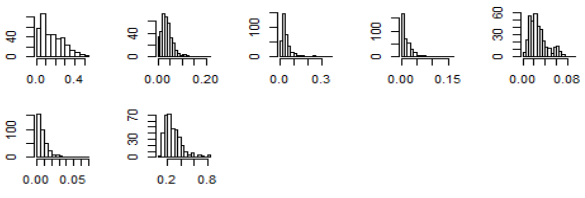
```
dim(donnees)

## [1] 365  59

nzv <- nearZeroVar(donnees)

donneesFiltrees <- donnees[, -nzv]

dim(donneesFiltrees)

## [1] 365  52
```

3. Traitement des données numériques

Nous avons d'abord extrait toutes nos variables quantitatives. Puis nous avons tracé les histogrammes de ces dernières pour avoir une idée de leurs répartitions.

```
num <- sapply(donneesFiltrees, is.numeric)
donnum <- donneesFiltrees[ ,num]

par(mar = rep(2, 4))
hist(donnum)
```



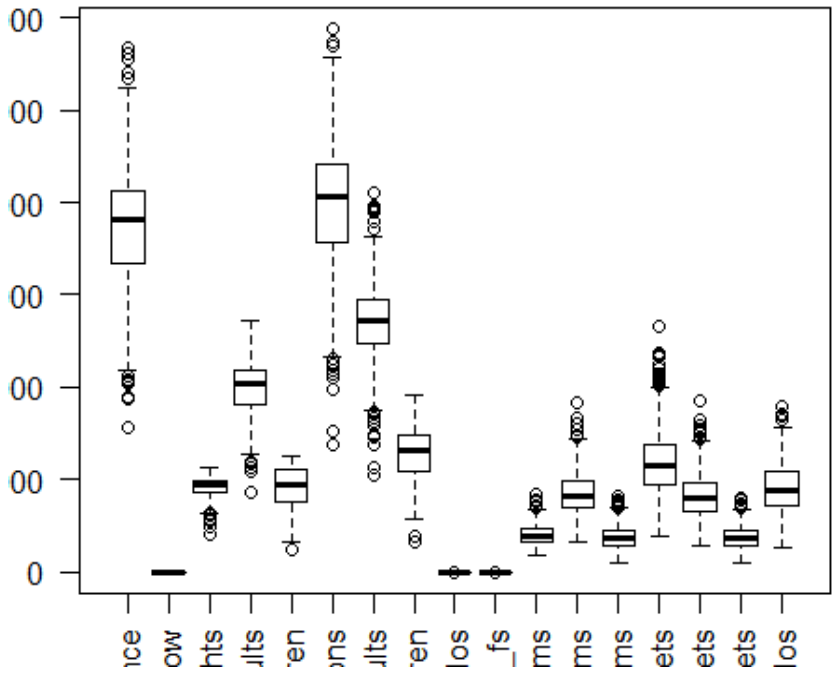


Nous remarquons que les distributions des variables sont assez diverses, il serait sage d'envisager des transformations de variables comme le logarithme par exemple.

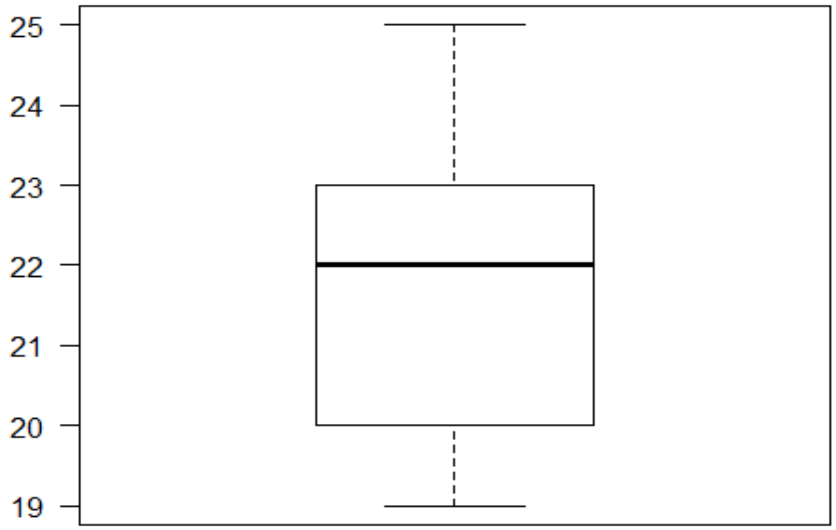
3.1. Valeurs extrêmes

Pour détecter nos observations, nous nous sommes servis de box plots et d'une ACP. Nous remarquons que la plupart de nos variables contiennent des valeurs extrêmes. Quand nous les avons analysés de plus près, nous nous sommes rendu compte que ces valeurs correspondent à des jours de grande fréquentation du parc (fin d'année, vacances scolaires…). C'est donc des valeurs normales. Nous avons décidé de les garder.

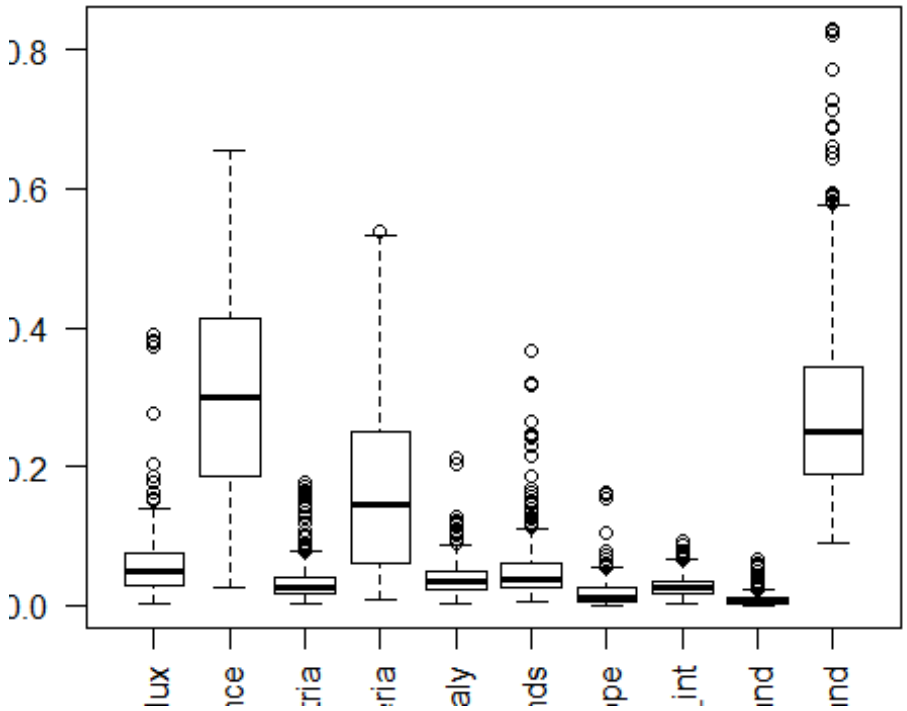
```
boxplot(donnum[, 1:17], las = 2)
```



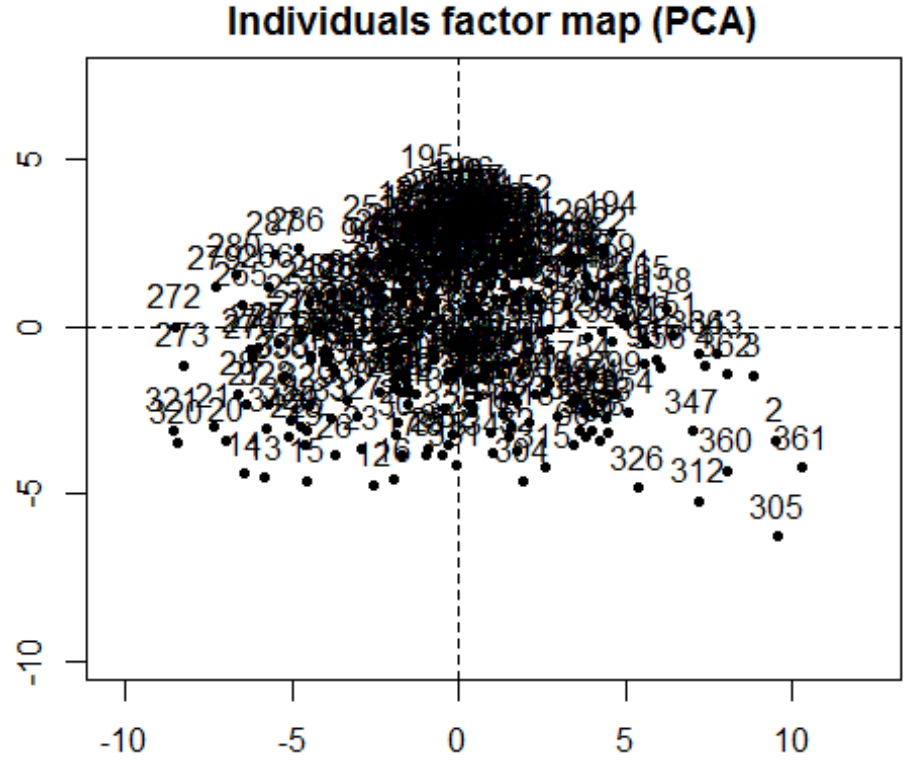

```
boxplot(donnum[, 19], las = 2)
```



```r
boxplot(donnum[, 28:37], las = 2)
```

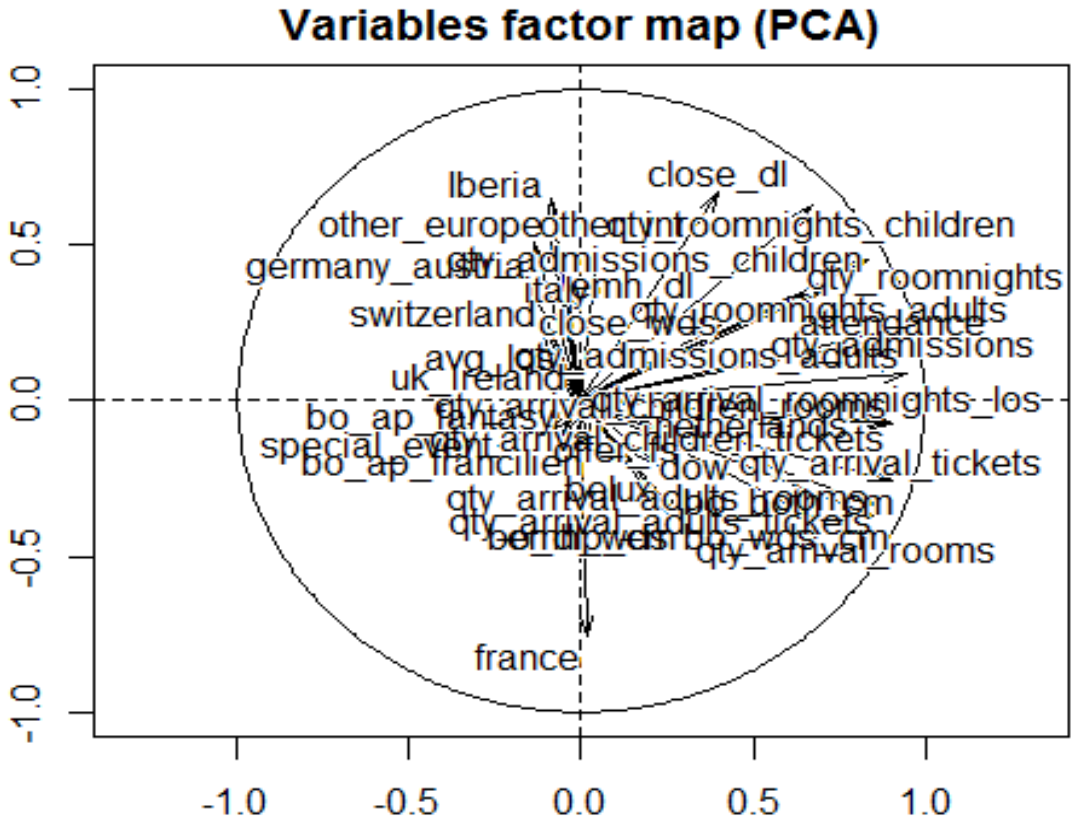



```r
#sapply(donnum, Boxplot)

acp <- PCA(donnum)
```

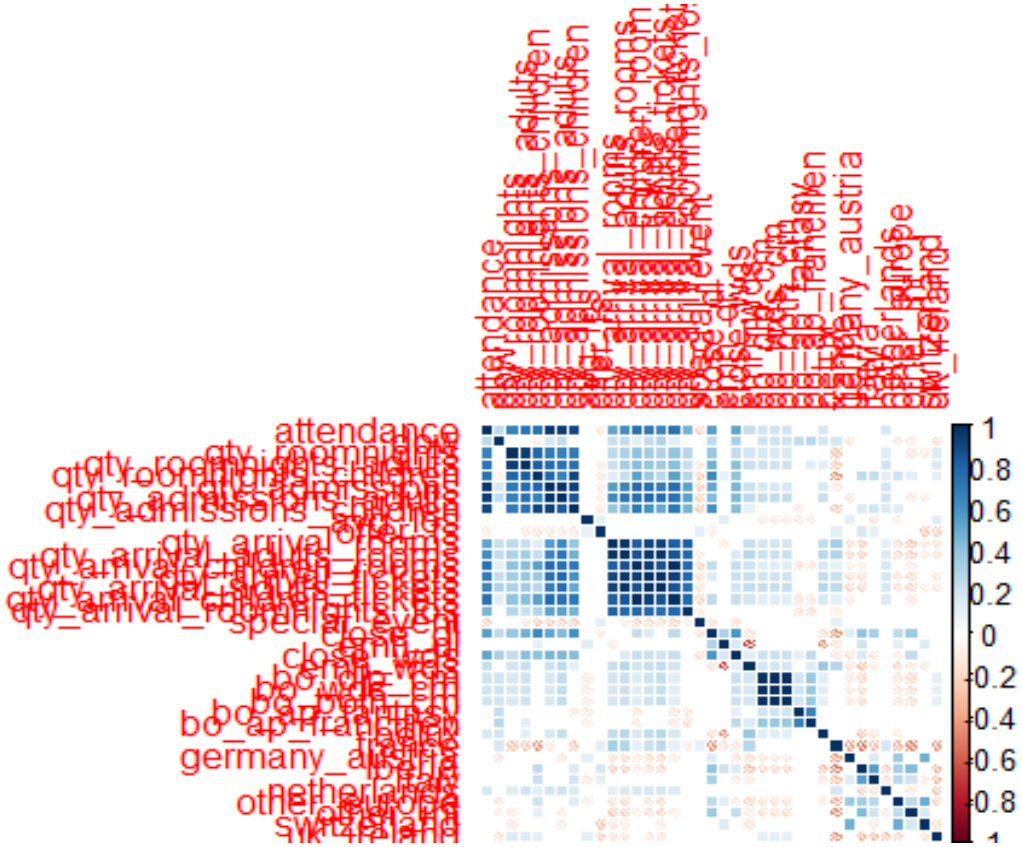


3.2. Corrélations

Nous constatons, sur le graphique du plan factoriel, l'existence de fortes corrélations entre nos variables. Pour avoir une idée plus claire, nous avons représenté notre matrice de corrélation avec le code couleur dégradé : rouge pour les corrélations négatives et bleu pour les positives. Nous avons mis en plan les corrélations non significatives. Nos résultats sont rassurants : les variables comme le nombre de nuitées, le nombre d'admissions, d'enfants, d'adultes… ont des impacts très positifs sur l'attendance. Cependant, nous avons aussi remarqué l'existence de fortes liaisons entre certaines de nos variables, ce qui nous penser à étudier l'existence de combinaisons linéaires entre elles.




```
M <- rcorr(as.matrix(donnum))

corrplot(M$r, method = "shade", p.mat = M$P, insig = "blank" )
```



3.3. Détection des combinaisons linéaires

En utilisant la fonction **findLinearCombos** nous détectons quatre combinaisons linéaires entre nos variables. Par exemple le nombre d'admissions total est la somme du nombre d'admission des adultes et de celui des enfants. Nous avons donc supprimé les colonnes redondantes.

```
comboInfo <- findLinearCombos(donnum)
comboInfo
```

```
## $linearCombos
## $linearCombos[[1]]
## [1] 8 6 7
##
## $linearCombos[[2]]
## [1] 16 14 15
##
## $linearCombos[[3]]
## [1] 24 23
##
## $linearCombos[[4]]
## [1] 25 23
##
##
## $remove
## [1]  8 16 24 25
```

```
donnumfiltrees <- donnum[, -comboInfo$remove]
```

4. Reconstitution de la base

Nous avons reconstitué notre base de données en rajoutant nos variables qualitatives à nos variables quantitatives filtrées.

```
bdd <- cbind(donfact, donnumfiltrees)
```

5. Échantillonnage

En séries chronologiques, il n'est pas logique de faire un échantillonnage aléatoire car il existe un ordre aux données, celui du temps. Comme le but de nombre modèle est d'établir des prévisions sur une semaine, nous avons décidé de faire l'apprentissage sur trois semaines (21 jours) et le tester sur une semaine (7 jours) consécutivement.

```
trainIndex <- createTimeSlices(bdd$attendance, initialWindow = 21, horizon
= 7, fixedWindow = TRUE, skip = 21)

trainSlices <- trainIndex[[1]]
testSlices <- trainIndex[[2]]

myTimeControl <- trainControl(method = "timeslice", initialWindow = 21, hor

izon = 7, fixedWindow = TRUE)
```

6. Modélisation

En raison de la forte colinéarité entre nos variables, nous avons choisi d'utiliser un modèle de régression PLS à l'aide du package caret.

Nous avons réussi à avoir un résultat qui semble bon mais on est toujours à la recherche de comment exploiter nos résultats et si c'est le cas, comment les améliorer.

```
> attach(disney)
> trainIndex <- createTimeSlices(disney$attendance, initialWindow = 21,
+                                horizon = 7, fixedWindow = TRUE, skip = 21)
>
> trainSlices <- trainIndex[[1]]
> plsFit <- train(attendance ~ .,
+                 data = disney[trainSlices[[1]],],
+                 method = "pls",
+                 preProc = c("center", "scale"))
```

```
> plsFit
Partial Least Squares

 21 samples
165 predictors

Pre-processing: centered (605), scaled (605)
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 21, 21, 21, 21, 21, 21, ...
Resampling results across tuning parameters:

  ncomp  RMSE      Rsquared   RMSE SD   Rsquared SD
  1      2574.925  0.8446476  659.5111  0.09875778
  2      1999.927  0.8929395  658.7467  0.08552966
  3      1713.012  0.9113898  671.1701  0.07214628

RMSE was used to select the optimal model using  the smallest value.
The final value used for the model was ncomp = 3.
```

Pour l'instant, nous avons réalisé le test avec plusieurs autre méthodes de régression, nous préfèrons présenter les résultats dans la deuxième phase du projet afin d'avoir une synthèse logique et cohérente.

II. Prochaines étapes

Dans un premier temps, nous avons réussi à avoir une vision globale de notre jeu de données, notre analyse statistique nous a permis d'ajuster notre champ de recherche par rapport aux méthodes employées pour prédire le nombre de personnes dans le parc.

Dans un deuxième temps, nous réalisons une comparaison entre les différentes méthodes de régression appliquées à notre jeu de données afin d'aboutir au modèle optimal pour notre cas de figure.