M2 IFMA et MPE

TP 2 : méthodes statistiques élémentaires

1 Importation des données pour le TP

Télécharger le fichier auto-mpg.ods. Ce fichier renseigne des caractéristiques techniques de 398 voitures des années 70-80. Les variables renseignées sont :

consommation (miles per gallon)

nombre de cylindres

cylindrée du moteur (cu. inches)

puissance

poids (lbs.)

temps d'acceleration (sec.) de 0 à 60 mph

année du modèle

origine du véhicule (1 : American, 2 : European, 3 : Japanese)

modèle du véhicule.

Importer les données sous la forme d'un tableau (dataframe) que vous nommerez Auto, en respectant les consignes suivantes :

- Donner les noms suivants aux variables du tableau : mpg, cylinders, displacement, horsepower, weight, acceleration, modelyear, origin, modele.
- Le fichier comporte des données manquantes symbolisées par des points d'interrogations. Accéder à l'aide de la fonction `read.table()` et étudier le fonctionnement de l'option `na.strings` pour importer correctement les données manquantes.
- La variable origin est importée automatiquement comme une variable numérique, faites en une variable de facteurs et changer les niveaux pour indiquer directement la provenance géographique.
- Les premières lignes du tableau sont données avec la commande `head()`.

2 Statistiques univariées

Nous étudions les données du fichier `auto-mpg.ods`. Executer la commande

```
> attach(Auto)
```

2.1 Analyse d'une variable quantitative

Un boxplot (boîte à moustache ou diagramme en boîte) est un résumé graphique de la distribution d'une variable. La fonction R qui trace le boxplot est `boxplot()`. Pour obtenir, par exemple, le boxplot de la variable consommation :

```
> boxplot(mpg)
```

On verra plus loin que les boxplots sont surtout utiles pour comparer plusieurs distributions de données. Notez que les différentes caractéristiques affichées par le boxplot peuvent être obtenues en demandant le "summary" de la variable

```
> summary(mpg)
```

L'instruction

```
> plot(density(mpg))
```

permet de représenter une estimation de la densité de la variable mpg. Calculer la moyenne empirique m et l'écart type sdt de la variable consommation. Représenter sur un même graphique un histogramme (avec la fonction `hist`) de mpg, une estimation de la densité ainsi que la densité gaussienne estimée et ajouter une légende.

Représenter la fonction de répartition empirique des données à l'aide de l'instruction

```
> plot(ecdf(mpg))
```

Superposer sur ce graphique la fonction de répartition de la loi gaussienne de paramètres m et sdt.

Simuler dans un vecteur x un 1000-échantillon d'une loi gaussienne de paramètres m et sdt. Représenter le graphique quantile-quantile (qq-plot) des vecteurs mpg et x, commenter.

Quelques tests statistiques

Calculer la p-valeur du test de Student "$moyenne(mpg) = 23$" contre "$moyenne(mpg) \neq 23$" :

```
> t.test(mpg, mu = 23)
```

Donner un intervalle de confiance pour la moyenne à 86% (consulter l'aide de la fonction `t.test()`). Exécuter ensuite les commandes

```
>  shapiro.test(mpg)
>  ks.test(mpg,x)
```

pour effectuer un test de normalité de la variable consommation. Faut-il pour autant remettre en cause la validité du test de Student effectué auparavant ?

2.2 Analyse d'une variable catégorielle

La fonction `table()` renvoie le tableau des fréquences d'une variable catégorielle :

```
> table(origin)
```

Calculer les proportions de chacune des origines géographiques dans l'échantillon. Stocker le résultat des proportions dans un vecteur appelé `prop`. Utiliser la fonction `barplot()` pour afficher un diagramme en bâtons représentant les proportions des origines géographiques dans l'échantillon.

En utilisant l'aide de R, déterminer ce que renvoient les lignes de code ci-dessous :

```
> T = table(origin)
> prop.test(T[1]),n= sum(T),p=0.5)
```

3 Liens entre deux variables

3.1 Deux variables numériques

En utilisant la fonction `plot()`, représenter quelques nuages de points de paires de variables numériques du tableau `Auto`. Vous pourrez aussi représenter la matrice des nuages avec la fonction `pairs()`. Calculer les corrélations linéaires correspondant à ces croisements : il est possible de calculer la matrice des corrélation de toutes les variables numériques comme suit :

```
> cor(Auto[,1:7],use = "complete.obs")
```

Que se passe-t-il si l'on retire l'option `use = "complete.obs"` ? A quoi sert cette option ?

Représenter la consommation en fonction de l' année du modèle. Calculer les moyennes de consommation par année. Superposer ces moyennes au nuage initial.

Choisir un croisement de deux variables numériques et faire un "test de corrélation nulle" à l'aide de la fonction `cor.test()`. Si la p-value est très élevée, ceci signifie-t-il qu'il n'y a pas de correlation linéaire entre les deux variables ? De façon générale, l'absence de correlation linéaire entre deux variable numériques signifie-t-il l'absence de lien entre celles-ci ?

3.2 Une variable numérique et une variable catégorielle

On souhaite déterminer si la provenance géographique a une influence sur la consommation des véhicules. Comparer les boxplots des trois distributions de consommation.

Superposer sur un même graphique les trois densités estimées de la consommation par origine géographique, ajouter un titre et une légende.

Comparer les distributions deux à deux en utilisant des procédures `var.test()` et `t.test()` sur deux groupes. En utilisant une méthode de Bonferroni, discuter l'égalité des moyennes des trois distributions.

Créer une nouvelle variable group.year indiquant la période d'origine de la voiture : 70-73, 74-77 ou 78-82. Etudier le lien entre cette group.year et mpg.