TP Statistique

Cédric Milinaire, Corentin Laharotte

4 avril 2020

Voici le plan de ce qui sera fait dans le TP.

0. Visualisation de chemins

Lecture du fichier des villes :

```
villes <- read.csv('DonneesGPSvilles.csv',header=TRUE,dec='.',sep=';',quote="\"")
str(villes)
```

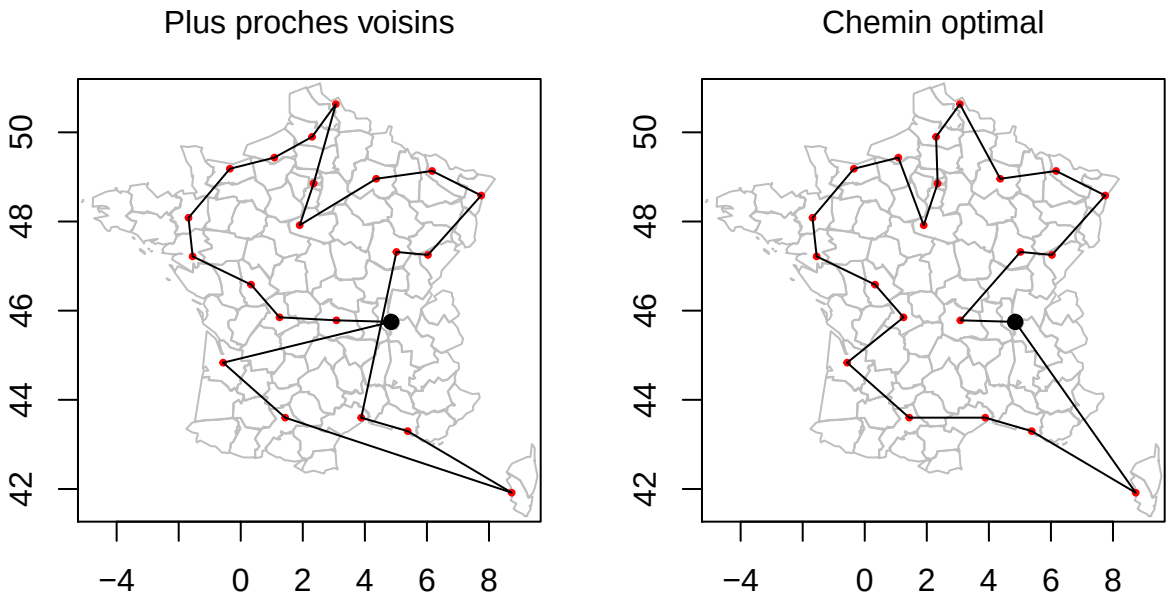
```
## 'data.frame':    22 obs. of  5 variables:
##  $ EU_circo : Factor w/ 7 levels "Centre","Est",..: 6 6 4 2 7 4 2 1 2 4 ...
##  $ region   : Factor w/ 22 levels "Alsace","Aquitaine",..: 22 9 19 10 2 4 8 3 5 17 ...
##  $ ville    : Factor w/ 22 levels "Ajaccio","Amiens",..: 11 1 2 3 4 5 6 7 8 9 ...
##  $ latitude : num  45.7 41.9 49.9 47.2 44.8 ...
##  $ longitude: num  4.847 8.733 2.3 6.033 -0.567 ...
```

Représentation des chemins par plus proches voisins et du chemin optimal :

```
coord <- cbind(villes$longitude,villes$latitude)
dist <- distanceGPS(coord)
voisins <- TSPnearest(dist)

pathOpt <- c(1,8,9,4,21,13,7,10,3,17,16,20,6,19,15,18,11,5,22,14,12,2)

par(mfrow=c(1,2),mar=c(1,1,2,1))
plotTrace(coord[voisins$chemin,], title='Plus proches voisins')
plotTrace(coord[pathOpt,], title='Chemin optimal')
```



Les longueurs des trajets (à vol d'oiseau) valent respectivement, pour la méthode des plus proches voisins :

```
## [1] 4303.568
```

et pour la méthode optimale :

```
## [1] 3793.06
```

Ceci illustre bien l'intérêt d'un algorithme de voyageur de commerce. Nous allons dans la suite étudier les performances de cet algorithme.

1. Comparaison d'algorithmes

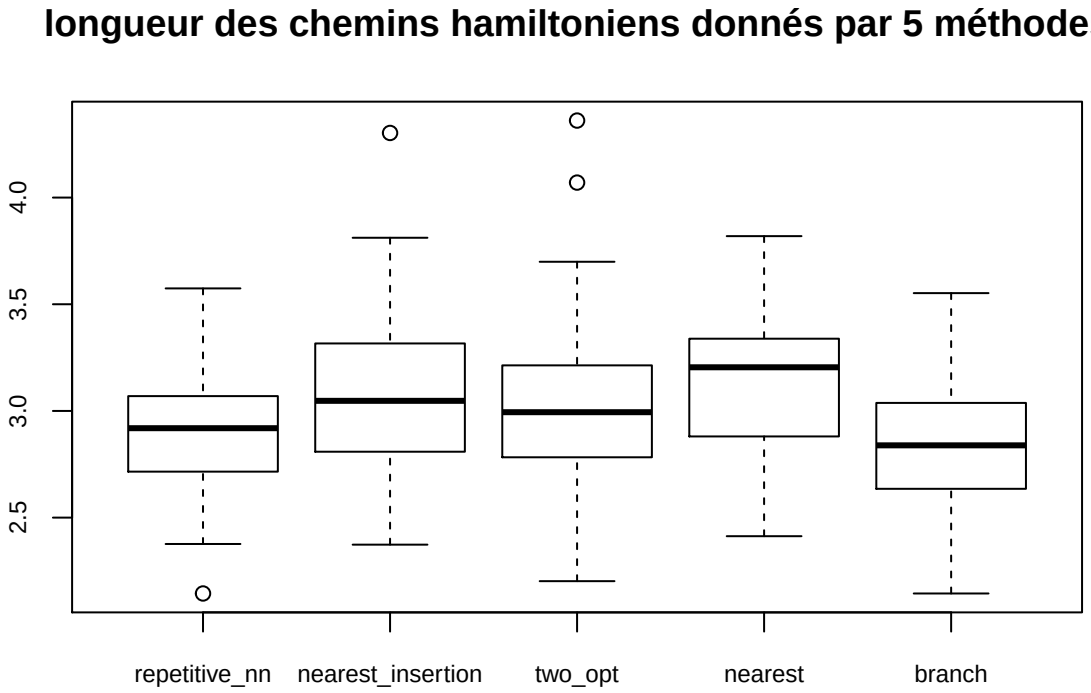
Dans cette partie, nous souhaitons comparer les méthodes repetitive_nn, nearest_insertion, two_opt, nearest, et branch. Pour cela, nous allons générer des graphes aléatoires de 10 sommets, et tester les longueurs des chemins calculés et le temps de calcul des différentes méthodes.

```
      n <- 10
sommets <- data.frame(x = runif(n), y = runif(n))
  couts <- distance(sommets)
```

1.1. Longueur des chemins

Dans un premier temps, nous allons comparer les longueurs des chemins hamiltoniens calculés par les 5 méthodes sur 50 réalisations de graphes aléatoires.

Représentation de la longueur des chemins hamiltoniens obtenus par différentes méthodes :



L'affichage sous forme de boxplot nous permet de remarquer que : * la méthode branch renvoie le plus souvent un chemin plus court que les autres méthodes * la méthode nearest renvoie le plus souvent un chemin plus long que les autres méthodes * la boîte de la méthode repetitive_nn est moins étendue que les boîtes obtenues par les autres méthodes, ce qui nous permet de constater que 50% des valeurs sont très proches de la valeur médiane * la boîte de la méthode nearest_insertion est plus étendue que les boîtes obtenues par les autres méthodes, ce qui nous permet de constater que 50% des valeurs sont assez étendues autour de la valeur médiane \

L'affichage obtenu est assez cohérent puisqu'aucune méthode n'a de valeur moyenne complètement absurde par rapport aux autre méthodes.

- test entre 'nearest' et 'branch'

On souhaite maintenant comparer les méthodes des plus proches voisins et Branch&Bound.\ On réalise donc un test sur l'espérance de chaque méthode.\ Notre hypothèse nulle (H0) est que la moyenne des chemins hamiltoniens obtenus avec la méthode des plus proches voisins est inférieure ou égale à la moyenne des chemins hamiltoniens obtenus avec la méthode Branch&Bound. Notre hypothèse alternative (H1) est que la moyenne des chemins hamiltoniens obtenus avec la méthode des plus proches voisins est supérieure à la moyenne des chemins hamiltoniens obtenus avec la méthode Branch&Bound.\ (H0) $m_{nn} - m_b <= 0 <=> m_{nn} <= m_b$ \ (H1) $m_{nn} - m_b > 0 <=> m_{nn} > m_b$ \

Nous allons ensuite tester si au seuil de 5% la moyenne des chemins hamiltoniens obtenus avec la méthode des plus proches voisins est inférieure ou égale à la moyenne des chemins hamiltoniens obtenus avec la méthode Branch&Bound. \ Pour cela, nous allons faire une comparaison d'échantillons gaussiens appariés.En effet, les deux méthodes étant basées sur les mêmes graphes, les résultats obtenus ne peuvent pas être considérés comme indépendant.\

On pose $a = 0.05$.

On obtient une p_{valeur} de :

```
## [1] 7.011422e-12
```

```
## [1] "p_valeur < a"
## [1] "On peut rejeter H0"
```

On observe que la p_{valeur} obtenue est strictement inférieure à a. \ On peut rejeter H0, et affirmer avec un risque de 5% que les chemins hamiltoniens obtenus avec la méthode des plus proches voisins sont en moyenne plus longs que ceux obtenus avec la méthode Branch&Bound.

- tests 2 à 2 On souhaite maintenant comparer 2 à 2 les longueurs moyennes des chemins hamiltoniens obtenus par les 5 méthodes vues précédemment.\ On réalise donc un test sur l'espérance de chaque méthode. \ Soit i, j deux méthodes différentes .Notre hypothèse nulle (H0) est que la moyenne des chemins hamiltoniens obtenus avec la méthode i est égale à la moyenne des chemins hamiltoniens obtenus avec la méthode j. Notre hypothèse alternative (H1) est que la moyenne des chemins hamiltoniens obtenus avec la méthode i est différente de la moyenne des chemins hamiltoniens obtenus avec la méthode j.\ (H0) $mi = mj$ \ (H1) $mi! = mj$ \

Nous avons lancé 10 tests simultanés, et obtenus les résulats suivants : \

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  results and methods
##
##                   branch  nearest nearest_insertion repetitive_nn
## nearest           0.00078 -       -                 -
## nearest_insertion 0.02272 0.94921 -                 -
## repetitive_nn     0.94921 0.01702 0.20157           -
## two_opt           0.09341 0.53849 0.94921           0.53849
##
## P value adjustment method: holm
```

Nous allons tester si au seuil de 5%, notre hypothèse H0 est vérifiée.\ \

Si on accepte un risque alpha=5%, on rejette notre hypothèse nulle (H0) si la p_{valeur} obtenue à l'indice [i,j] est inférieure à alpha.\ Donc, si la valeur à l'indice [i,j] est inférieure à alpha, nous pouvons affirmer que la moyenne des chemins hamiltoniens obtenus avec la méthode i est différente de celle obtenue avec la méthode j.\

En appliquant ce principe à nos résultats, nous pouvons dire que : - les méthodes nearest et branch ont des moyennes de chemins calculés différentes - les méthodes nearest_insertion et branch ont des moyennes de chemins calculés différentes - les méthodes nearest et repetitive_nn ont des moyennes de chemins calculés différentes

Pour les autres méthodes, nous ne pouvons pas rejeter l'hypothèse d'après laquelle la moyenne des chemins hamiltoniens obtenus avec la méthode i est égale à la moyenne des chemins hamiltoniens obtenus avec la méthode j.

1.2. Temps de calcul

Nous souhaitons maintenant comparer les temps d'éxécution des différentes méthodes de calcul de longueur de chemin hamiltonien sur 20 graphes de 10 sommets générés aléatoirement.\

Nous avons utilisé la focntion benchmark pour réaliser des statistiques d'exécution pour chaque méthode. Nous avons réalisé des tests sur les temps moyens d'exécution de chaque méthode : \ Soit i, j deux méthodes différentes .Notre hypothèse nulle (H0) est que le temps moyen d'exécution de la méthode i est égale au temps moyen d'exécution de la méthode j. Notre hypothèse alternative (H1) est que le temps moyen d'exécution de la méthode i est différent du temps moyen d'exécution de la méthode j.\ Le résultat de ces tests est représenté par une lettre dans la colonne X du tableau ci-dessous. Une même lettre est attribuée aux méthodes pour lequelles H0 n'est pas rejetée. Deux méthodes ayant des lettres différentes ont des temps d'exécution moyens différents.\

-en moyenne ils sont tous a peu pres equivalent en parecequ'on ne peut pas rejeter h0 pour tous sauf repitive nn -confirmer en regardant les chiffres repetitve nn prend en moyenne au moins 2x plus de temps que les

autres

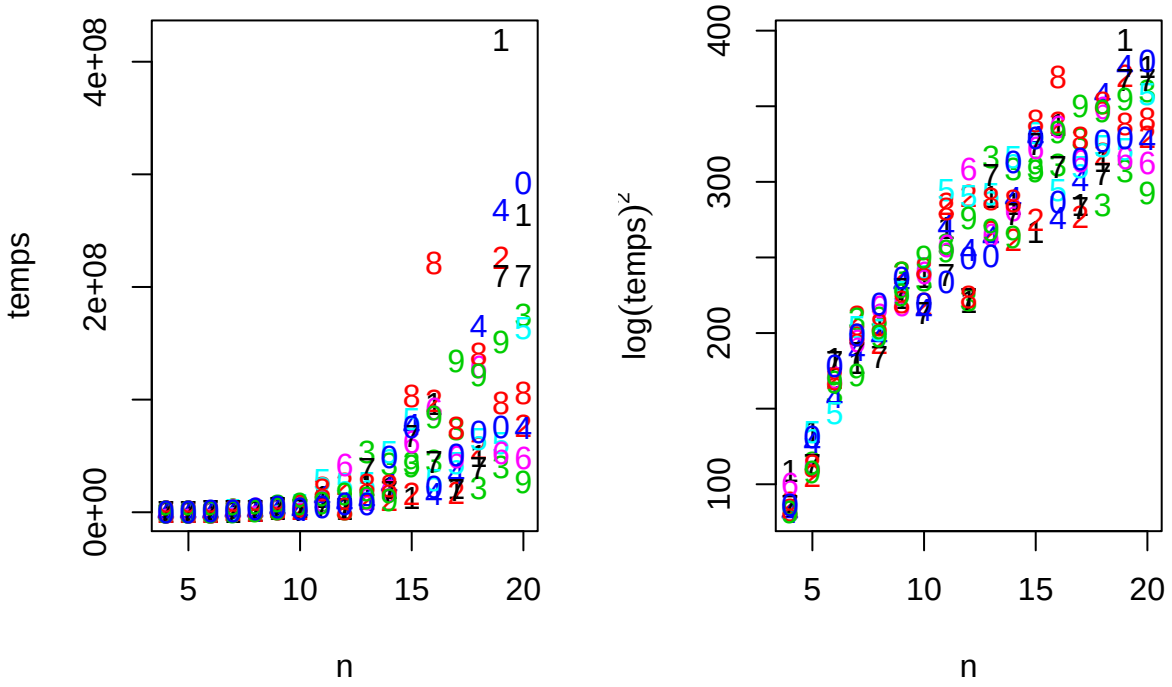
commentaire prof

Tout les membres d'un meme groupe n'ont pas de différence significative pour leurs moyenne et les groupes {a, b, c, d, . . . } sont rangés de manière croisante. Exemple - si variables X et variable Y sont dans le groupe a alors m X ' m Y où plutot qu'il n'a pas pu être mis en évidence une différence significative entre les deux. - si variables X et variable Y sont dans le groupe a et b alors m X 6 = m Y significativement. Et comme {a, b, c, d, . . . } sont rangés de manière croisante alors m a < m b donc m X < m Y

2. Etude de la complexité de l'algorithme Branch and Bound

2.1. Comportement par rapport au nombre de sommets : premier modèle

Récupération du temps sur 10 graphes pour différentes valeurs de n.



Les nombres représentés sont les numéros de colonnes de la valeur à la nième ligne !

Ajustement du modèle linéaire de $\log(temps)^2$ en fonction de n.

```
##
## Call:
## lm(formula = vect_temps ~ vect_dim)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -77.833 -19.419   4.071  20.775  57.969
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  74.8148     5.6623   13.21   <2e-16 ***
## vect_dim     14.7755     0.4369   33.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 27.9 on 168 degrees of freedom
## Multiple R-squared:  0.8719, Adjusted R-squared:  0.8712
## F-statistic:  1144 on 1 and 168 DF,  p-value: < 2.2e-16
```

COMMENTER

- on peut voire que log(tmps^2) en fonction de n suit une courbe lineaire (R squareed = 0.8705) du coups le temps est une fonction exponentielle de n i.e la complexite de temps est exponentielle.

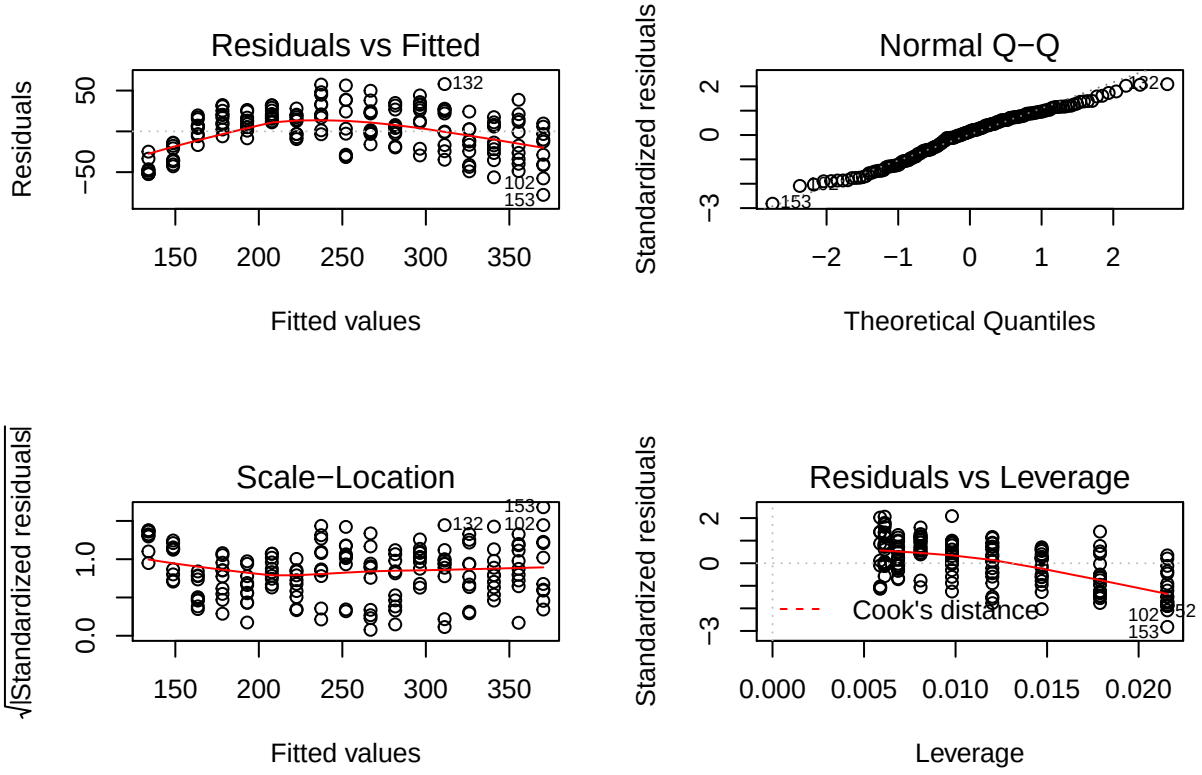
Analyse de la validité du modèle :

Le modèle nous renvoie une fonction de type: $Y = aX + b + \epsilon$. En effet nous avons les paramètres suivants: $a = 14.7$, $b = 68.6$. Il reste donc a savoir les coéfficients et donc le modèle sont pertinents. Nous allons tous d'abord analyser la pertinence des coefficients puis celle du modèle en géneral.

- L'analyse de a, permet d'établir un premier résultat quantifiant la significativité du modèle. En effet nous allons tester la significativité de a via le test statistique: $(H0) : a = 0$ contre $(H1) : a \neq 0$. La p-value de celui-ci ce retrouve dans le tableau summary(temps.lm) et est $2.2e-16$. Nous sommes donc capable d'affiirmer avec un risque de de moins que 0.1% (chiffre arbitraire plus grand que $2.2e-14$) que a est significatif.

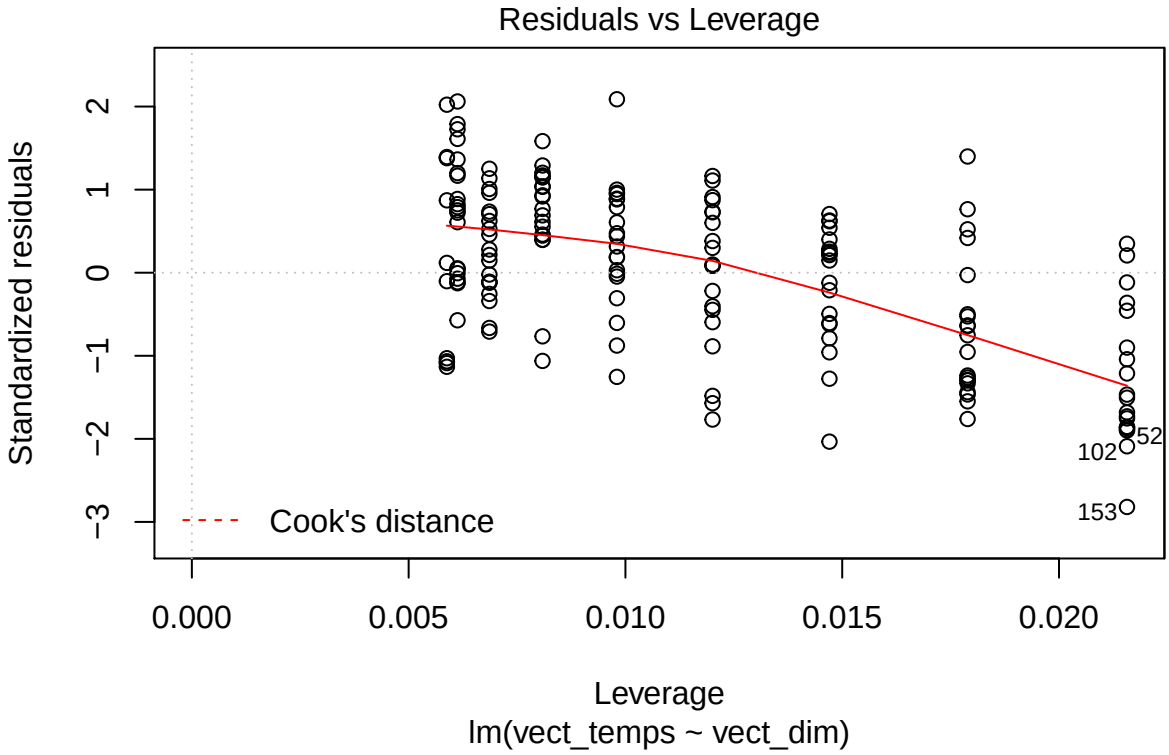
- L'analyse de b est la moins importante. Il nous indique seulement l'importance de l'intercept. Le test statistique est analogique à a. Ca p-value est aussi $2.2e-16$ Nous sommes donc capable d'affiirmer avec un risque de de moins que 0.1% (chiffre arbitraire plus grand que $2.2e-14$) que l'intercept est utile.

- Nous pouvons maintenant passé a l'analyse des résidus:
  - Pour ceci nous allons tous d'abord nous intéréssé a plusieurs graphique:



Residuals vs Fitted: La courbe n'est pas complètement horizontal. Il y'a donc un léger effet d'échelle. Normal Q-Q: les points sont proches de la bissectrice, la distribution des résidus est donc similaire à la distribution normale. Nous voyons une légère séparation au niveau des queues des distribution. *Scale Location: cette*

courbe représente la meme chose que la première seulement avec dles résisu normalisé. On remarque que la courbe est bien honrizontal que le léger effet d'échelle disparait. Pour la distance de cook nous avons préfére prendre le graphique suivant: Nous voyons qu'aucun résidu a une distance plus grande que 0.05 et que la plus part on une distance inferieure a 0.01. Ceci indique que le bon fit du modèle.



- Il est aussi possible d'éffectuer un test statistique sur les résidus. En effet s'ils suivent une loi normale ceci indique le bon fit du modèle.
- Définissons: ($H0$) *les résidus suivent une loi normale* ($H1$) les résidus ne suivent pas une loi normale
- Pour tester ceci nous pouvons efféctuer un test de shapiro.

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(temps.lm)
## W = 0.97704, p-value = 0.006429
```

```
## [1] "p-valeur < alpha"
## [1] "On peut rejeter H0"
```

On ne peut rejetter H0, donc nous pouvons affirmer que les résidusne suivent une loi normale. Ce qui indique un modèle pertinent.

2.2. Comportement par rapport au nombre de sommets : étude du comportement moyen

L'explication des résultats étants similaire a 2.1, nous allons simplement afficher nos résultats.\ Récupération du temps moyen.

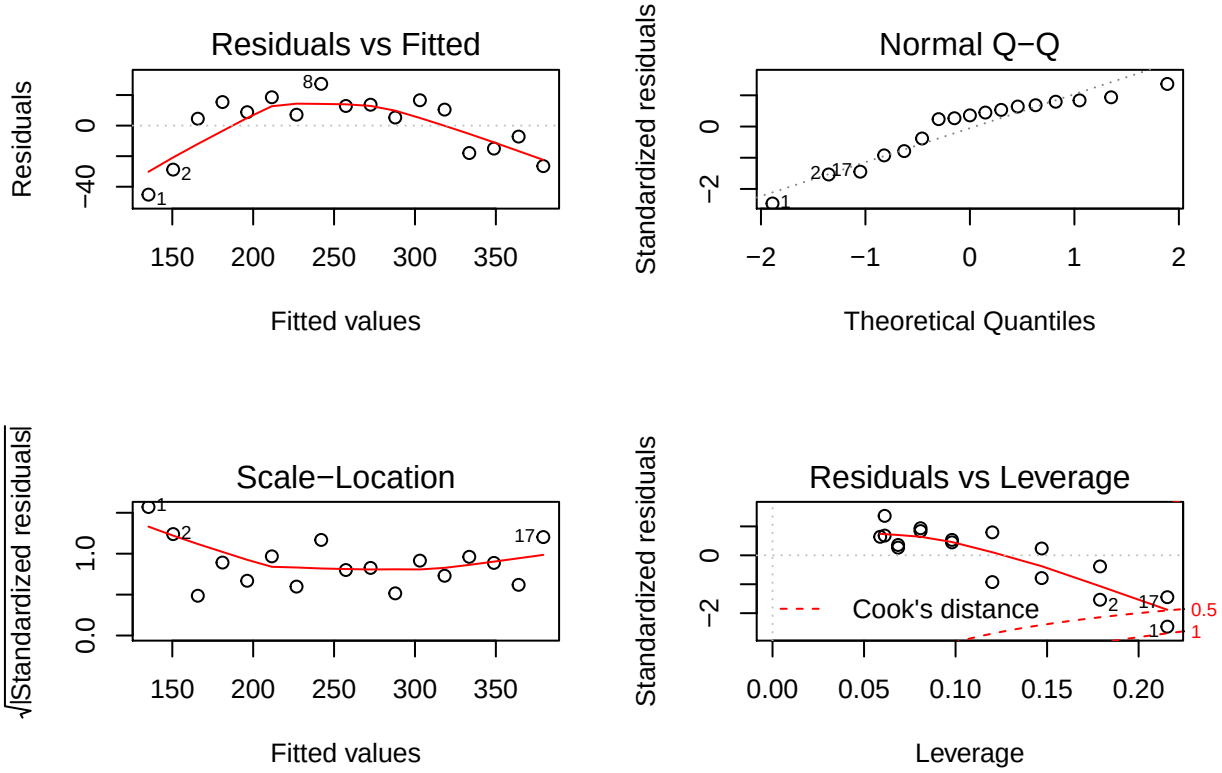
Ajustement du modèle linéaire de $\log(temps.moy)^2$ en fonction de n.

```
##
## Call:
## lm(formula = vect_temps_moy ~ vect_dim_moy)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.171 -15.037   7.131  13.673  27.332
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    74.264     13.270   5.596 5.10e-05 ***
## vect_dim_moy   15.253      1.024  14.898 2.14e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.68 on 15 degrees of freedom
## Multiple R-squared:  0.9367, Adjusted R-squared:  0.9325
## F-statistic:   222 on 1 and 15 DF,  p-value: 2.137e-10
```

Analyse de la validité du modèle :

- a pertinent $p - value = 2.3 * 10^{-9}$
- b pertinent $p - value = 0.000476$



Residuals vs Fitted: La courbe n'est pas du tou horizontal. Il y'a donc un important effet d'échelle. Normal Q-Q: Les distributions sont identiques. *Scale Location: L'effet d'échelle disparait. Le nuage de point est sans structure. Ce qui indique la qualité du modèle.

- Il est aussi possible d'éffectuer un test statistique sur les résidus. En effet s'ils suivent une loi normale ceci indique le bon fit du modèle.
- Définissons: $(H0)$ *les résidus suivent une loi normale* $(H1)$ les résidus ne suivent pas une loi normale
- Pour tester ceci nous pouvons efféctuer un test de shapiro.

(H0) les résidus suivent une loi normale (H1) les résidus ne suivent pas une loi normale

On prend un risque alpha=5%

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(temps.lm_moy)
## W = 0.90628, p-value = 0.08659
```

```
## [1] "p-valeur >= alpha"
## [1] "On ne peut pas rejeter H0"
```

On ne peut rejetter H0, donc nous pouvons affirmer que les résidus suivent une loi normale. Ce qui indique un modèle pertinent.

2.3. Comportement par rapport à la structure du graphe

Lecture du fichier 'DonneesTSP.csv'.

```
data.graph <- data.frame(read.csv('DonneesTSP.csv'))
data.graph$dim<-sqrt(data.graph$dim)
str(data.graph)
```

```
## 'data.frame':    70 obs. of  8 variables:
##  $ tps      : num  53692 144081 997803 2553322 6333009 ...
##  $ dim      : num  2 2.45 2.83 3.16 3.46 ...
##  $ mean.long: num  0.391 0.442 0.334 0.276 0.254 ...
##  $ mean.dist: num  0.665 0.592 0.537 0.506 0.502 ...
##  $ sd.dist  : num  0.276 0.259 0.246 0.238 0.227 ...
##  $ mean.deg : num  3 5 7 9 11 13 15 17 19 3 ...
##  $ sd.deg   : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ diameter : num  1 1 1 1 1 1 1 1 1 1 ...
```

- D'après nous les variables non pertinentes sont: diameter, mean.dist, sd.dist, mean.long. En effet tous ces variables s'intérèssent seulement aux couts des arretes. Ces derniers n'ont pas d'importance dans le temps de calcul (calculer le chemin avec une arrete de 5 ou dfe 1000 reviens a la meme chose).

Ajustement du modèle linéaire de $\log(temps.moy)^2$ en fonction de toutes les variables présentes. Modèle sans constante.

```
model.complete <- lm(log(tps)~., data = data.graph)
summary(model.complete)
```

```
##
## Call:
## lm(formula = log(tps) ~ ., data = data.graph)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78776 -0.15715  0.01542  0.17260  0.65036
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.4903426  0.5450715  11.907  < 2e-16 ***
## dim          3.4191719  0.2391476  14.297  < 2e-16 ***
## mean.long   -4.8152962  0.7294055  -6.602 1.05e-08 ***
## mean.dist   -0.0020048  0.0010633  -1.886  0.06404 .
## sd.dist      0.0048105  0.0006652   7.231 8.55e-10 ***
## mean.deg    -0.1367369  0.0425459  -3.214  0.00208 **
```

```
## sd.deg       0.1399515  0.0872430   1.604  0.11376
## diameter    -0.0646816  0.1566329  -0.413  0.68107
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2912 on 62 degrees of freedom
## Multiple R-squared:  0.986,  Adjusted R-squared:  0.9844
## F-statistic: 622.6 on 7 and 62 DF,  p-value: < 2.2e-16
```

```
(step(model.complete))
```

```
## Start:  AIC=-165.23
## log(tps) ~ dim + mean.long + mean.dist + sd.dist + mean.deg +
##     sd.deg + diameter
##
##             Df Sum of Sq     RSS      AIC
## - diameter   1    0.0145  5.2711 -167.038
## <none>                    5.2566 -165.230
## - sd.deg     1    0.2182  5.4748 -164.384
## - mean.dist  1    0.3014  5.5581 -163.327
## - mean.deg   1    0.8757  6.1324 -156.444
## - mean.long  1    3.6951  8.9517 -129.965
## - sd.dist    1    4.4335  9.6902 -124.417
## - dim        1   17.3311 22.5877  -65.176
##
## Step:  AIC=-167.04
## log(tps) ~ dim + mean.long + mean.dist + sd.dist + mean.deg +
##     sd.deg
##
##             Df Sum of Sq     RSS      AIC
## <none>                    5.2711 -167.038
## - sd.deg     1    0.2065  5.4776 -166.349
## - mean.dist  1    0.6554  5.9265 -160.835
## - mean.deg   1    0.9820  6.2531 -157.080
## - mean.long  1    3.8220  9.0931 -130.869
## - sd.dist    1    4.9133 10.1844 -122.935
## - dim        1   18.7788 24.0499  -62.785
```

```
##
## Call:
## lm(formula = log(tps) ~ dim + mean.long + mean.dist + sd.dist +
##     mean.deg + sd.deg, data = data.graph)
##
## Coefficients:
## (Intercept)          dim    mean.long    mean.dist      sd.dist     mean.deg
##    6.396008     3.444077    -4.854857    -0.002284     0.004883    -0.140823
##      sd.deg
##    0.126916
```

- La variable diameter à été supprimé du modèle. Il reste sd.di, mean.dist et mean.long. Qui nous apparaissent peut pertinente.

```
new_model <- lm(formula = log(tps) ~ dim + mean.long + mean.dist + sd.dist +
    mean.deg + sd.deg, data = data.graph)
summary(new_model)
```

```
##
## Call:
## lm(formula = log(tps) ~ dim + mean.long + mean.dist + sd.dist +
##     mean.deg + sd.deg, data = data.graph)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78246 -0.15445  0.00111  0.18027  0.63156
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.3960078  0.4916227  13.010  < 2e-16 ***
## dim          3.4440771  0.2298893  14.981  < 2e-16 ***
## mean.long   -4.8548566  0.7183110  -6.759 5.25e-09 ***
## mean.dist   -0.0022837  0.0008160  -2.799  0.00680 **
## sd.dist      0.0048833  0.0006372   7.663 1.39e-10 ***
## mean.deg    -0.1408227  0.0411061  -3.426  0.00108 **
## sd.deg       0.1269156  0.0807943   1.571  0.12123
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2893 on 63 degrees of freedom
## Multiple R-squared:  0.9859, Adjusted R-squared:  0.9846
## F-statistic:   736 on 6 and 63 DF,  p-value: < 2.2e-16
```

- D'après le test de fisher le modèle est pertinent ($p_{value} = 2.2 * 10^{-16}$)

```
new_model <- lm(formula = log(tps) ~ dim + mean.long + mean.dist + sd.dist +
    mean.deg + sd.deg, data = data.graph)
shapiroTest_aic<-shapiro.test(residuals(new_model))
print(shapiroTest_aic)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(new_model)
## W = 0.98094, p-value = 0.3641
```

```
## [1] "p-valeur >= alpha"
## [1] "On ne peut pas rejeter H0"
```

On ne peut pas rejetter H0, donc nous pouvons affirmer que les résidus suivent une loi normale. Ce qui indique un modèle pertinent.

Mise en œuvre d'une sélection de variables pour ne garder que les variables pertinentes.

Analyse de la validité du modèle :

- pertinence des coefficients et du modèle,
- étude des hypothèses sur les résidus.