

TP Statistique

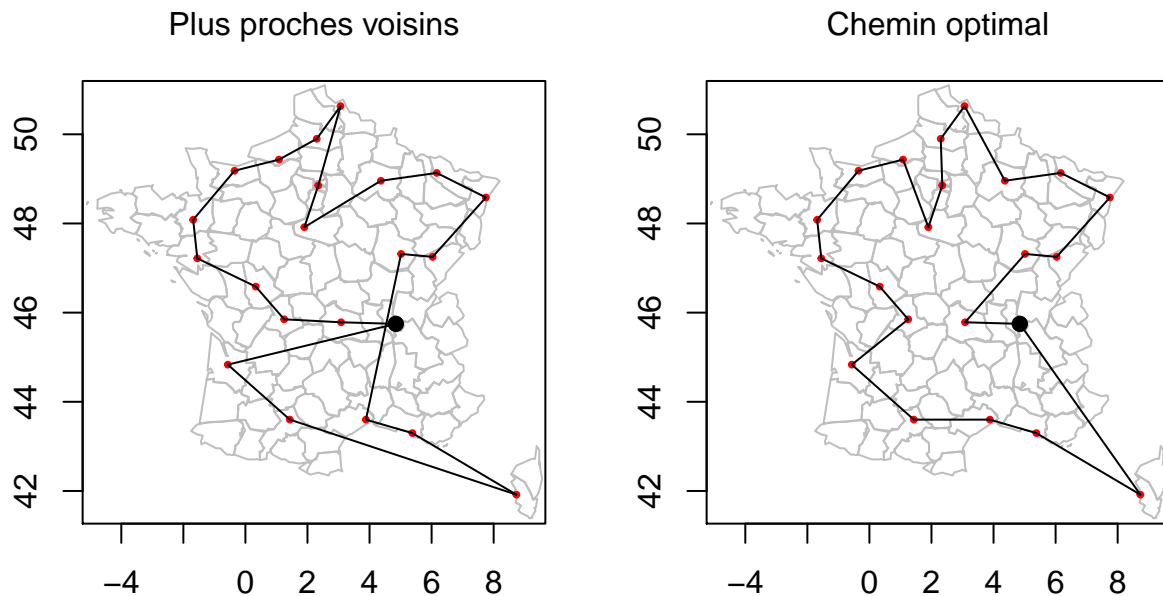
Emmy LERANDY et Louis UNG

31 mars 2020

0. Visualisation de chemins

On nous fournit dans ce TP des données de villes françaises stockées dans un fichier CSV et des algorithmes de calcul du chemin optimal.

Voici ci-dessous une représentation des chemins par la méthode des plus proches voisins et la méthode du 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
```

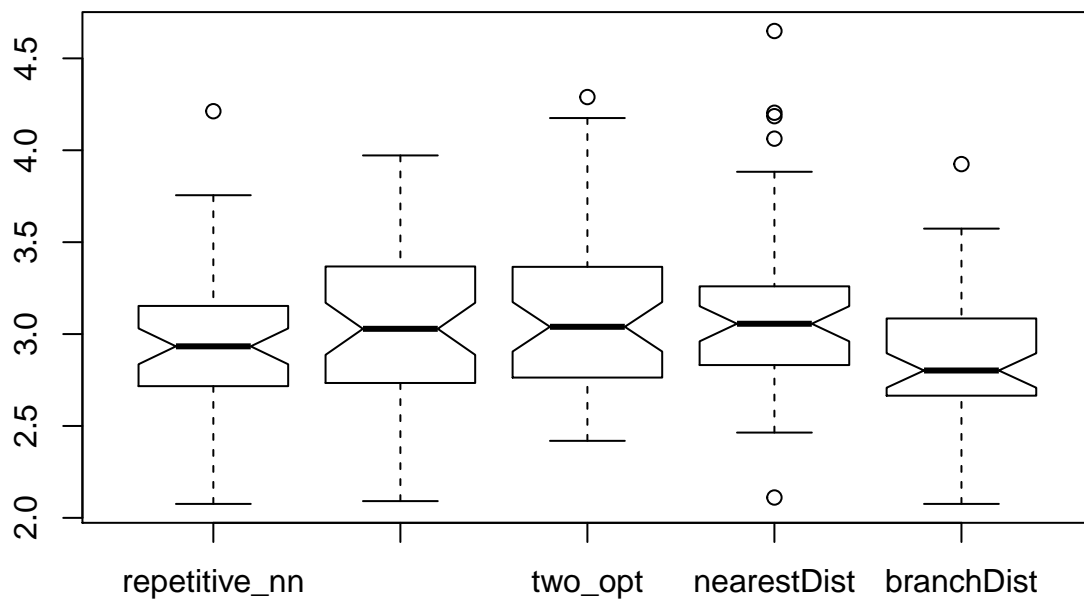
Ceci illustre bien l'intérêt d'un algorithme de voyageur de commerce. Nous allons dans la suite étudier les performances de cet algorithme.

1. Comparaison d'algorithmes

1.1. Longueur des chemins

Dans cette partie, il s'agit de comparer 5 différentes méthodes de calcul du chemin optimal : `repetitive_nn`, `nearest_insertion`, `two_opt`, `nearest` et `branch`. Les longueurs des chemins hamiltoniens générées par chaque algorithme est calculée sur 50 réalisations de graphes de 10 sommets générées aléatoirement.

Boxplots des 5 méthodes



Commentaire : On constate que Branch&Bound a la plus petite médiane, et a donc calculé le meilleur chemin optimal. En ce qui concerne les autres algorithmes, comme le résultat n'est pas stable entre chaque exécution du code, il est difficile d'en déduire d'autres informations supplémentaires.

Test entre nearest et branch

Nous cherchons à tester l'espérance des algorithmes nearest et branch. Soient mnn et mb les espérances respectives des algorithmes des plus proches voisins et de Branch&Bound. Nous supposons alors les hypothèses suivantes : $(H_0) mnn - mb \leq 0$ contre $(H_1) mnn - mb > 0$

```
##
## One Sample t-test
##
## data: nearestDist - branchDist
## t = 7.2937, df = 49, p-value = 2.33e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.1788553 0.3148939
## sample estimates:
## mean of x
## 0.2468746
```

Commentaire : L'espérance obtenue appartient à l'intervalle de confiance à 95%, nous ne rejetons donc pas l'hypothèse H_0 , l'espérance de l'algorithme Branch&Bound est donc inférieure à celui des plus proches voisins.

Tests 2 à 2

```
##
## Pairwise comparisons using t tests with pooled SD
##
## data: results and methods
##
##               Branch&Bound nearest nearest_insertion repetitive_nn
## nearest          0.030      -      -
## nearest_insertion 0.131      1.000 -
## repetitive_nn     1.000      0.411 0.945
## two_opt           0.051      1.000 1.000 0.551
##
## P value adjustment method: holm
```

Les résultats obtenus du tests de la correction multiple de Bonferroni sont présents ci-dessus.

Commentaire : Nous observons que la difference entre Branch&Bound et nearest_insertion, Branch&Bound et repetitive_nn, nearest_insertion et repetitive_nn, nearest et two_opt est supérieur ou égal 10^{-5} . Nous pouvons donc conclure pour ces tests que l'hypothèse H_0 est rejetée, ils n'ont donc pas la même espérance 2 à 2. A l'inverse, pour les autres pairs de tests, les résultats sont tous inférieurs ou égaux à 10^{-12} , valeur qu'on peut considérer suffisamment négligeable pour être approximée à 0 Nous pouvons donc pour ces pairs valider l'hypothèse H_0 , ils ont donc la même espérance 2 à 2.

En résumé, les pairs de tests validant l'hypothèse H_0 sont: nearest et Branch&Bound / nearest et insertion / nearest et repetitive_nn / Branch&Bound et two_opt / nearest_insertion et two_opt / repetitive_nn et two_opt

1.2. Temps de calcul

Dans cette partie, nous allons faire une comparaison des temps d'exécution à l'aide du package microbenchmark.

Microbenchmark

```
microbenchmark(TSPsolve(couts, 'repetitive_nn'), TSPsolve(couts, 'nearest_insertion'),
TSPsolve(couts, 'two_opt'), TSPnearest(couts)$longueur, TSPbranch(couts), times=400,
setup={
  n <- 10
  sommets <- data.frame(x = runif(n), y = runif(n))
  couts <- distance(sommets)
})
```

```
## Unit: microseconds
##              expr      min       lq      mean  median
##   TSPsolve(couts, "repetitive_nn") 4572.1 4888.05 5470.33650 5048.70
##   TSPsolve(couts, "nearest_insertion") 724.9 779.60 918.36850 830.95
##           TSPsolve(couts, "two_opt") 408.4 439.95 528.76575 473.75
##           TSPnearest(couts)$longueur   7.7   9.90 13.77175 12.90
##           TSPbranch(couts) 302.5 1919.20 3391.16100 2918.25
##           uq      max neval   cld
## 5337.90 63498.6 400    e
## 904.10 4353.1 400    c
## 543.05 3497.6 400    b
## 14.60 132.3 400    a
## 4206.45 13832.8 400    d
```

Nous avons décidé d'augmenter le nombre de graphes étudiés car les résultats qu'on obtenait étaient trop instables avec 20 graphes. Après plusieurs essais, sur 400 graphes, nous obtenons dans la majorité des cas pour la dernière colonne le classement suivant : e pour repetitive_nn, c pour nearest_insertion, b pour two_opt, a pour nearest et d pour Branch&Bound.

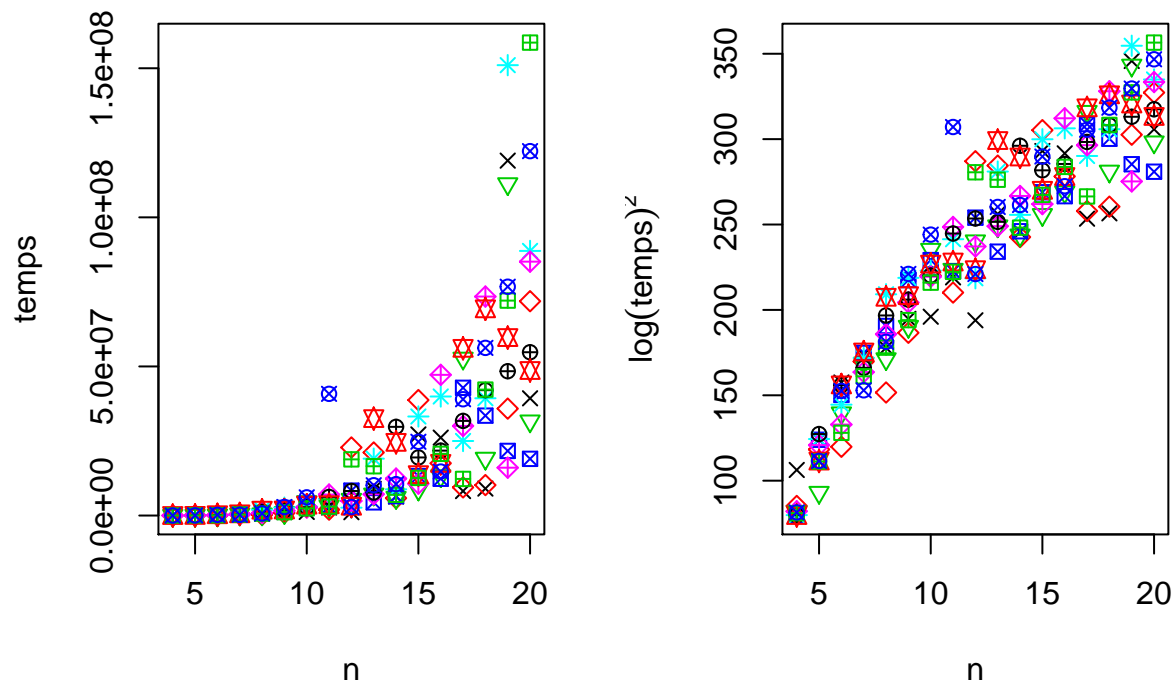
Commentaire: Le plus rapide en temps d'exécution parmi les 5 méthodes d'après microbenchmark est donc la méthode nearest, suivie de nearest_insertion et ensuite de nearest_insertion, puis par Branch&Bound et enfin repetitive_nn qui est la méthode la plus lente.

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

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

Dans un premier temps nous allons construire un modèle de régression linéaire simple du temps d'exécution de Branch&Bound en fonction du nombre de sommets n .

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



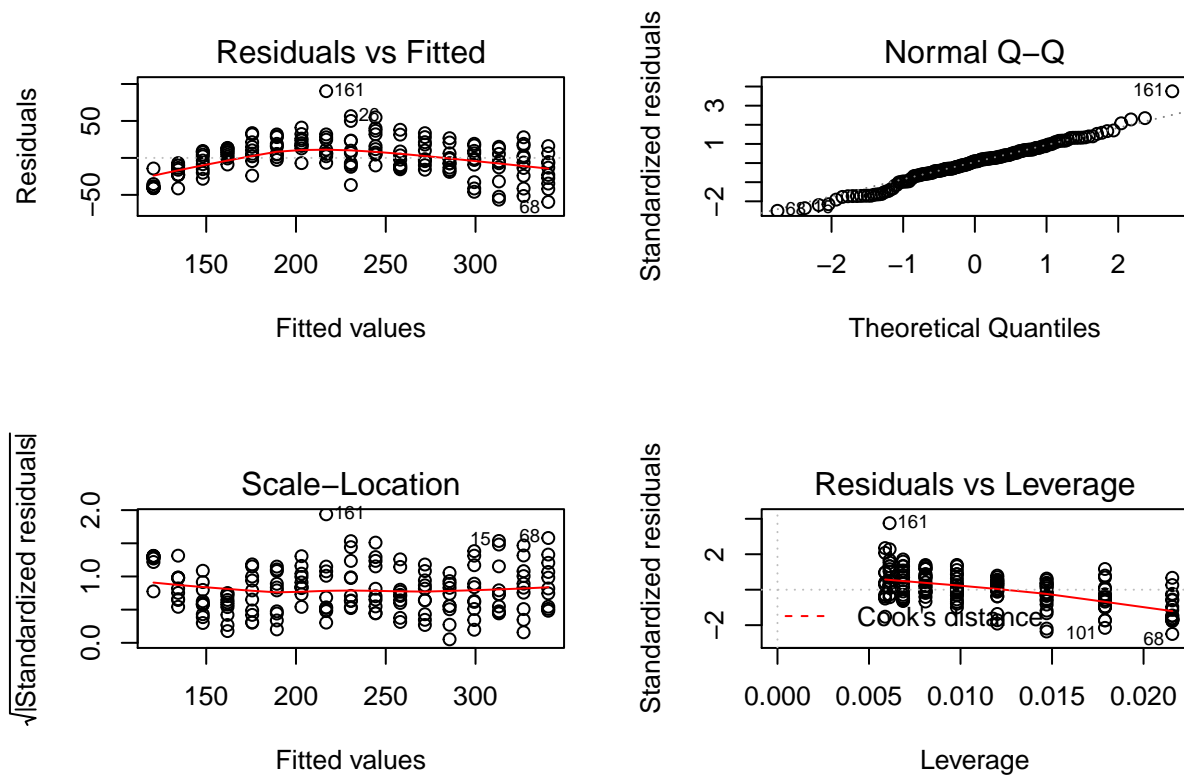
Après ajustement, on observe qu'on obtient bien une courbe qui ressemble plus à une droite.

Test de Fisher

```
##
## Call:
## lm(formula = vect_temps ~ vect_dim)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -59.593 -13.207   2.176  15.745  90.284
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  65.6482     4.8976   13.40  <2e-16 ***
## vect_dim     13.7430     0.3779   36.37  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.14 on 168 degrees of freedom
## Multiple R-squared:  0.8873, Adjusted R-squared:  0.8866
## F-statistic: 1323 on 1 and 168 DF,  p-value: < 2.2e-16
```

Commentaire : On observe qu'on obtient un coefficient R^2 ajusté valant 0.8758, on est donc proche de 1. Or plus ce coefficient est proche de 1, plus la courbe obtenue sera proche d'une droite. Nous avons donc une relation de proportionnalité entre le logarithme au carré du temps d'exécution et le nombre de sommets. Par ailleurs, on obtient une p-value inférieure à 10^{-16} , donc très négligeable et approximable à la valeur 0, on ne rejette donc pas l'hypothèse H_0 et le test de Fisher est validé. On en conclut que ce modèle est donc pertinent.

Etude graphique



Commentaire : Dans les graphes Residuals vs Fitted et Scale-Location, on observe que les points ne sont pas homogènes et une droite non-horizontale, on n'a donc pas de linéarité. Dans le graphe Normal Q-Q, on obtient une droite très semblable à une diagonale, la distribution des résidus peut donc être assimilée à une distribution normale. Dans le graphe Residuals vs Leverage, on est dans les bornes de la distance de Cook donc la qualité de l'échantillon de données est correcte.

Test de Shapiro-Wilk

```
shapiro.test(residuals(temps.lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(temps.lm)
## W = 0.9842, p-value = 0.05097
```

Commentaire : On obtient une p-value grande, qui est largement supérieure au risque alpha qui est de 0.05. Ainsi, on ne rejette pas l'hypothèse H_0 , les résidus du modèle suivent donc possiblement une distribution normale.

Conclusion : le modèle est valide.

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

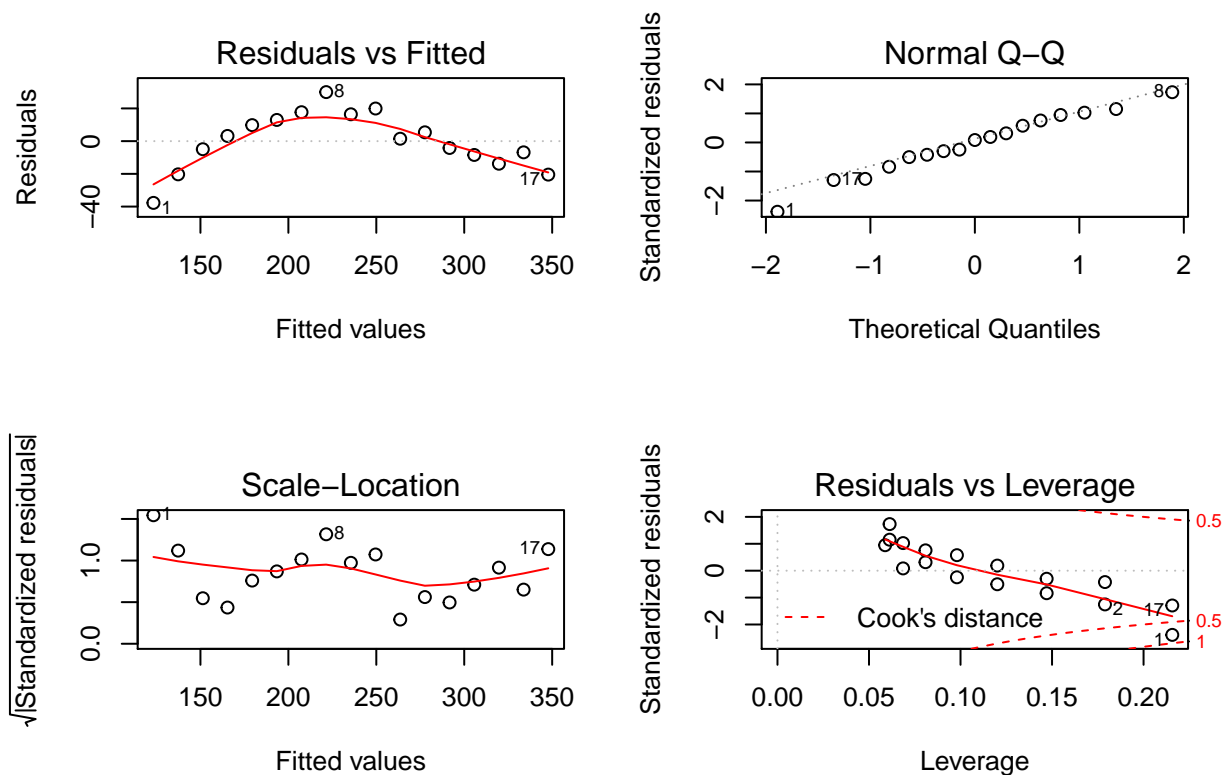
Dans un second temps nous allons construire cette fois un modèle de régression linéaire simple basé sur le temps moyen d'exécution de Branch&Bound en fonction du nombre de sommets n .

Ajustement du modèle de régression linéaire simple gaussien de $\log(\text{temps.moy})^2$ en fonction de seqn

```
##
## Call:
## lm(formula = vect_temps.moy ~ seqn)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -37.748  -8.454   1.460  12.955  29.961
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  67.1574    11.4647   5.858 3.15e-05 ***
## seqn         14.0317     0.8845  15.864 8.79e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.87 on 15 degrees of freedom
## Multiple R-squared:  0.9437, Adjusted R-squared:  0.94
## F-statistic: 251.7 on 1 and 15 DF,  p-value: 8.787e-11
```

Commentaire : On observe qu'on obtient un coefficient R^2 ajusté valant 0.9385, on est donc proche de 1. Or plus ce coefficient est proche de 1, plus la courbe obtenue sera proche d'une droite. Nous avons donc une relation de proportionnalité entre le logarithme au carré du temps d'exécution et le nombre de sommets. Par ailleurs, on obtient une p-value inférieure à 10^{-10} , donc très négligeable et approximable à la valeur 0, on ne rejette donc pas l'hypothèse H_0 et le test de Fisher est validé. On en conclut que la statistique $F = 0$, donc la modèle n'apporte aucune information utile.

Etude graphique



Dans les graphes Residuals vs Fitted et Scale-Location, on observe que les points ne sont pas homogènes et une droite non-horizontale, on n'a donc pas de linéarité dans ce cas aussi. Dans le graphe Normal Q-Q, on obtient des points qui sont éparpillés autour de la diagonale, mais si on cherche à relier ces points on obtient une courbe ressemblant à une diagonale, on en déduit donc que la distribution des résidus peut être assimilée à une distribution normale. Dans le graphe Residuals vs Leverage, on observe un outlier (le numéro 1), la qualité de l'échantillon en est donc impactée. Il faudrait supprimer cet outlier pour améliorer la qualité de l'échantillon des données et augmenter leur pertinence. Cependant, l'écart est relativement faible et nous pouvons donc quand même affirmer que nous avons un échantillon de qualité acceptable.

Test de Shapiro-Wilk

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(temps.moy.lm)  
## W = 0.98359, p-value = 0.9827
```

Commentaire : On obtient à l'issue du test une p-value de valeur élevée. Cette p-value est largement supérieure au risque alpha qui est de 5%, nous ne rejetons pas l'hypothèse H_0 . Les résidus du modèle peuvent donc suivre une distribution normale.

Conclusion : le modèle est valide

2.3. Comportement par rapport à la structure du graphe

Construction du modèle de régression

Nous allons construire le modèle de régression de $\log(\text{tps})^2$ par rapport à $\sqrt{\text{dim}}$ et toutes les autres variables, exceptées la variable tps. Une fois le modèle construit, nous mettons en oeuvre la sélection de variables à l'aide de la fonction step. On obtient le résultat ci-dessous :

```
## Start: AIC=272.28
## vect_tps ~ sqrt(dim) + mean.long + mean.dist + sd.dist + mean.deg +
##      sd.deg + diameter
##
##           Df Sum of Sq    RSS    AIC
## - mean.dist  1      75.2  2798.5 272.18
## <none>                        2723.2 272.28
## - diameter   1     230.7  2953.9 275.97
## - sd.deg     1     269.1  2992.3 276.87
## - mean.deg   1     552.5  3275.7 283.21
## - mean.long  1    2122.7  4846.0 310.62
## - sd.dist    1    2483.9  5207.2 315.65
## - sqrt(dim)  1   14671.8 17395.0 400.08
##
## Step: AIC=272.18
## vect_tps ~ sqrt(dim) + mean.long + sd.dist + mean.deg + sd.deg +
##      diameter
##
##           Df Sum of Sq    RSS    AIC
## <none>                        2798 272.18
## - sd.deg     1      299  3098 277.29
## - mean.deg   1      714  3512 286.08
## - diameter   1      718  3516 286.17
## - mean.long  1     2068  4867 308.92
## - sd.dist    1     3116  5914 322.56
## - sqrt(dim)  1    44208 47007 467.67
##
## Call:
## lm(formula = vect_tps ~ sqrt(dim) + mean.long + sd.dist + mean.deg +
##      sd.deg + diameter, data = dataset)
##
## Coefficients:
## (Intercept)      sqrt(dim)    mean.long      sd.dist    mean.deg
##      -10.118       93.493     -113.470        0.102       -2.447
##      sd.deg      diameter
##       5.159       -11.133
```

Résultat de la sélection de variables

Nous observons qu'après la mise en oeuvre de la sélection de variables, la variable mean.dist a été exclue du modèle par l'algorithme de la fonction step(). Par ailleurs, nous pouvons constater qu'individuellement toutes les variables conservées ne sont pas autant pertinentes. En effet, on cherche à maximiser l'AIC et la variable sqrt(dim) semble ainsi être largement plus pertinente que les autres.

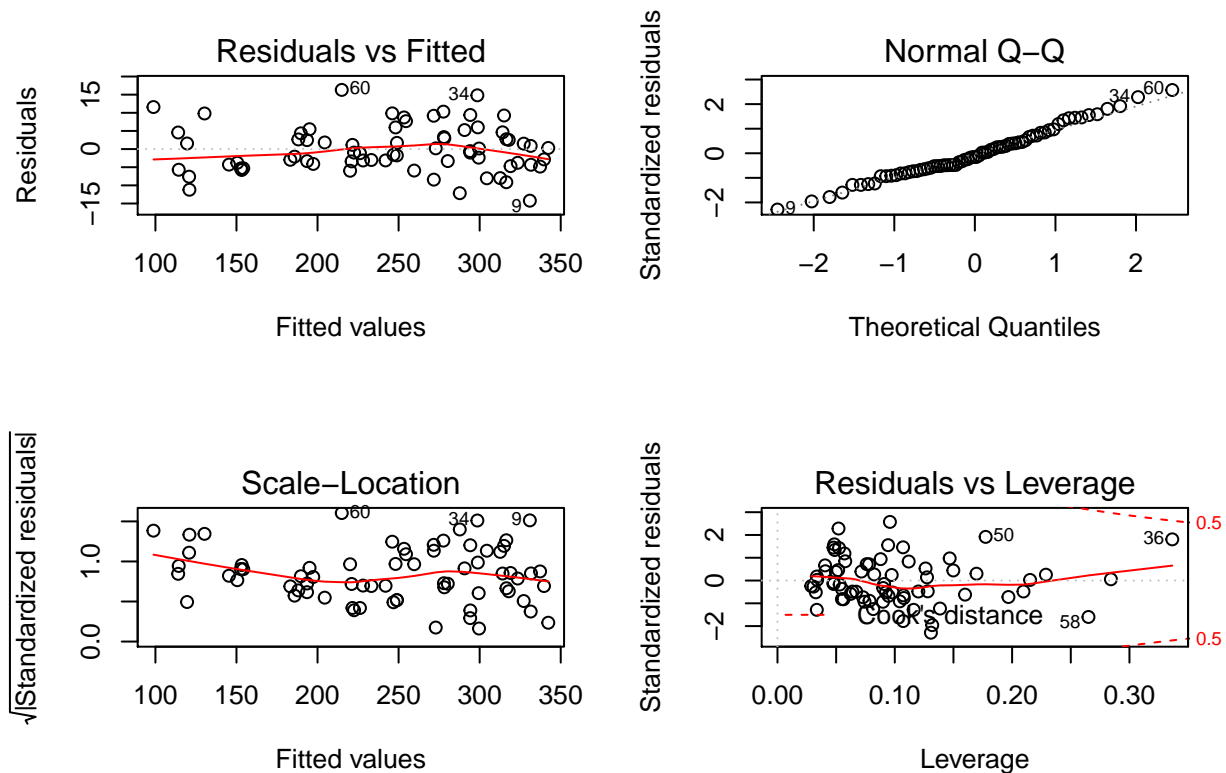
Test de Fisher

Nous effectuons le test de Fisher avec le nouveau modèle sans la variable mean.dist, comme elle a été déterminée comme non pertinente précédemment.

```
##
## Call:
## lm(formula = vect_tps ~ ., data = datasetAfterAIC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2246  -4.2312  -0.9755   4.0830  16.3081
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -10.11788    8.50435  -1.190  0.238617
## sqrtDIM      93.49324    2.96359  31.547 < 2e-16 ***
## mean.long    -113.47013   16.62867  -6.824  4.05e-09 ***
## sd.dist       0.10196    0.01217   8.375  7.90e-12 ***
## mean.deg      -2.44693    0.61053  -4.008  0.000165 ***
## sd.deg        5.15950    1.98806   2.595  0.011745 *
## diameter     -11.13317    2.76963  -4.020  0.000158 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.665 on 63 degrees of freedom
## Multiple R-squared:  0.9913, Adjusted R-squared:  0.9905
## F-statistic: 1199 on 6 and 63 DF,  p-value: < 2.2e-16
```

Commentaire : On observe qu'on obtient un coefficient R^2 ajusté valant 0.9905, on est donc très proche de 1. Nous avons donc une relation de proportionnalité entre le logarithme au carré du temps moyen d'exécution de l'algorithme Branch&Bound et le nombre de sommets. Par ailleurs, on obtient une p-value inférieure à 10^{-16} , donc très négligeable et approximable à la valeur 0, on ne rejette donc pas l'hypothèse H_0 et le test de Fisher est validé. On en conclut que ce modèle est donc pertinent.

Etude des résidus



Commentaire : Dans les graphes Residuals vs Fitted et Scale-Location, on observe que les points ne sont pas homogènes et une droite non-horizontale, on n'a donc pas de linéarité dans ce cas aussi. Dans le graphe Normal Q-Q, on obtient une courbe ressemblant à une diagonale, on en déduit donc que la distribution des résidus peut être assimilée à une distribution normale. Dans le graphe Residuals vs Leverage, on observe que tous les points sont compris dans la zone délimitée par les pointillés rouges, calculée à partir de la distance de Cook. Nous n'avons donc pas de valeurs aberrantes présentes dans cet échantillon, nous pouvons donc en déduire que nous avons un échantillon de bonne qualité.

Test de Shapiro-Wilk

```
##  
## Shapiro-Wilk normality test  
##  
## data: residuals(tps.lm_aic)  
## W = 0.98303, p-value = 0.4621
```

Commentaire : On obtient une p-value grande, qui est largement supérieure au risque alpha qui est de 0.05. Ainsi, on ne rejette pas l'hypothèse H_0 , les résidus du modèle suivent donc possiblement une distribution normale.

Conclusion : le modèle est valide.