

Capsule 11

Objectifs de la capsule

À la fin de cette capsule, vous serez en mesure de:

1. Comprendre la **démarche** de la **régression linéaire**,
2. Utiliser la fonction générique **lm()** utilisée pour les régressions linéaires dans R,
3. **Valider les résultats** de votre analyse,
4. Calculer des **intervalles de confiances sur la moyenne** et les **prédictions de valeurs individuelles**.

Les méthodes utilisées dans ce script sont avancées, mais chacune peut être facilement comprise en étudiant l'aide relative aux fonctions utilisées (voir capsule #3).

Capsule vidéo

<https://youtu.be/yER43DrP4C4>

Exercices

i Note

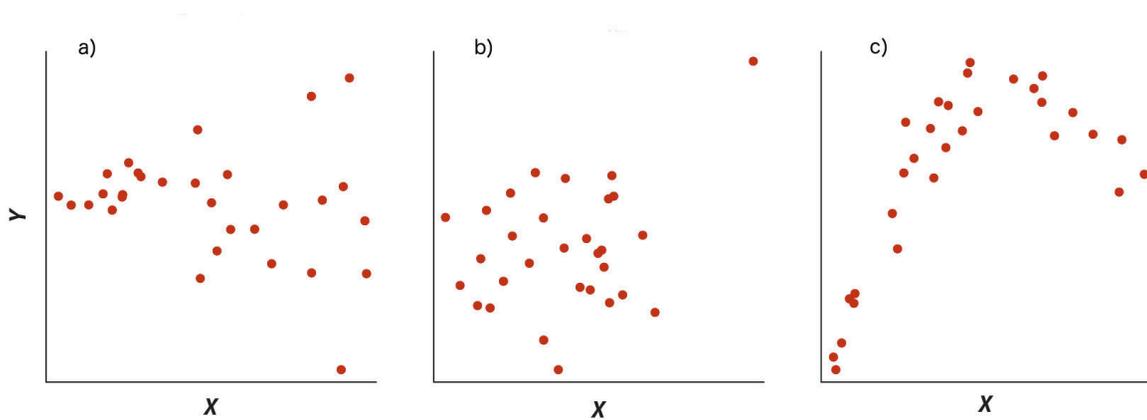
Veillez noter qu'il est possible d'avoir plus d'une bonne réponse par question. Vous pouvez reprendre chaque exercice grâce aux boutons "Start Over". Le bouton "Indice" est là pour être utilisé!

i Note

i Note

i Note

Associez à chaque graphique le problème principal que présentent les données et qui viole les **conditions d'application** de la régression linéaire simple :



i Note

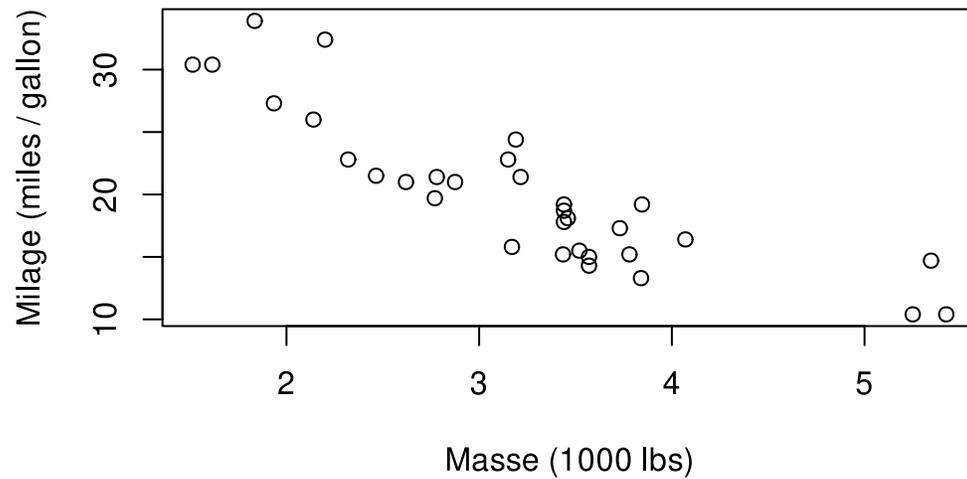
i Note

i Note

Voici un jeu de données. Il s'agit de variables mesurées sur 32 modèles d'automobiles et publiées dans le *Motor Trend US magazine* de 1974. Nous voulons prédire la consommation moyenne d'essence des modèles automobiles 73-74 selon leur masse.

```
plot(
  mtcars$mpg ~ mtcars$wt,
  xlab = "Masse (1000 lbs)",
  ylab = "Milage (miles / gallon)",
  main = "Consommation des automobiles"
)
```

Consommation des automobiles



i Note

Faites un histogramme des variables explicative `mtcars$wt` (la masse) et réponse `mtcars$mpg` (le milage) et décidez si leur distribution de fréquence semble suivre une loi Normale.

```
{webr}  
#| autorun: false
```

i Note

i Note

Faites la régression linéaire entre ces deux variables en utilisant la fonction `lm()`.

```
{webr}  
#| autorun: false
```

⚠ Solution

```
lm(mtcars$mpg ~ mtcars$wt)
```

```
Call:  
lm(formula = mtcars$mpg ~ mtcars$wt)
```

```
Coefficients:  
(Intercept)    mtcars$wt  
    37.285         -5.344
```

i Note

i Note

i Note

i Note

Finalement, on veut calculer l'intervalle de confiance à 95% des valeurs de milage attendues pour un véhicule de 1973-74 pesant 4750 lbs. En utilisant le modèle de régression suivant, complétez la fonction `predict()` avec la bonne option (remplacez les ...) pour calculer les limites de cet intervalle :

```
{webr}  
#| autorun: false  
mod <- lm(mpg ~ wt, data = mtcars)  
predict(mod, newdata = data.frame(wt = 4750 / 1000), ...)
```

💡 Indice

Utilisez le paramètre "interval" = ...

i Note

Matériel accompagnateur

Charger les données pour l'analyse

i Note

Tapez "Puromycin" dans l'aide de R pour en savoir plus (voir capsule #1).

Il s'agit d'un jeu de données fourni avec R et qui comprend des valeurs d'activité enzymatique mesurées pour différentes concentrations de substrat dans deux groupes de cellules, le premier traité avec un antibiotique inhibiteur de synthèse protéique (la puromycine), et le second un groupe contrôle non traité.

On va d'abord charger le jeu de données, puis le rendre accessible en premier plan dans la mémoire de R :

```
data("Puromycin")
```

Puromycin

```
   conc rate  state
1 0.02  76  treated
2 0.02  47  treated
3 0.06  97  treated
4 0.06 107  treated
5 0.11 123  treated
6 0.11 139  treated
7 0.22 159  treated
8 0.22 152  treated
9 0.56 191  treated
10 0.56 201  treated
11 1.10 207  treated
12 1.10 200  treated
13 0.02  67  untreated
14 0.02  51  untreated
15 0.06  84  untreated
16 0.06  86  untreated
```

17	0.11	98	untreated
18	0.11	115	untreated
19	0.22	131	untreated
20	0.22	124	untreated
21	0.56	144	untreated
22	0.56	158	untreated
23	1.10	160	untreated

La démarche de la régression linéaire

Une analyse de régression linéaire simple est utile lorsqu'il existe un **lien de causalité entre deux variables quantitatives** (numériques et mesurables) et que l'on veut **prédire** la valeur de la variable **réponse** (dépendante) selon la valeur de la variable **explicative** (indépendante).

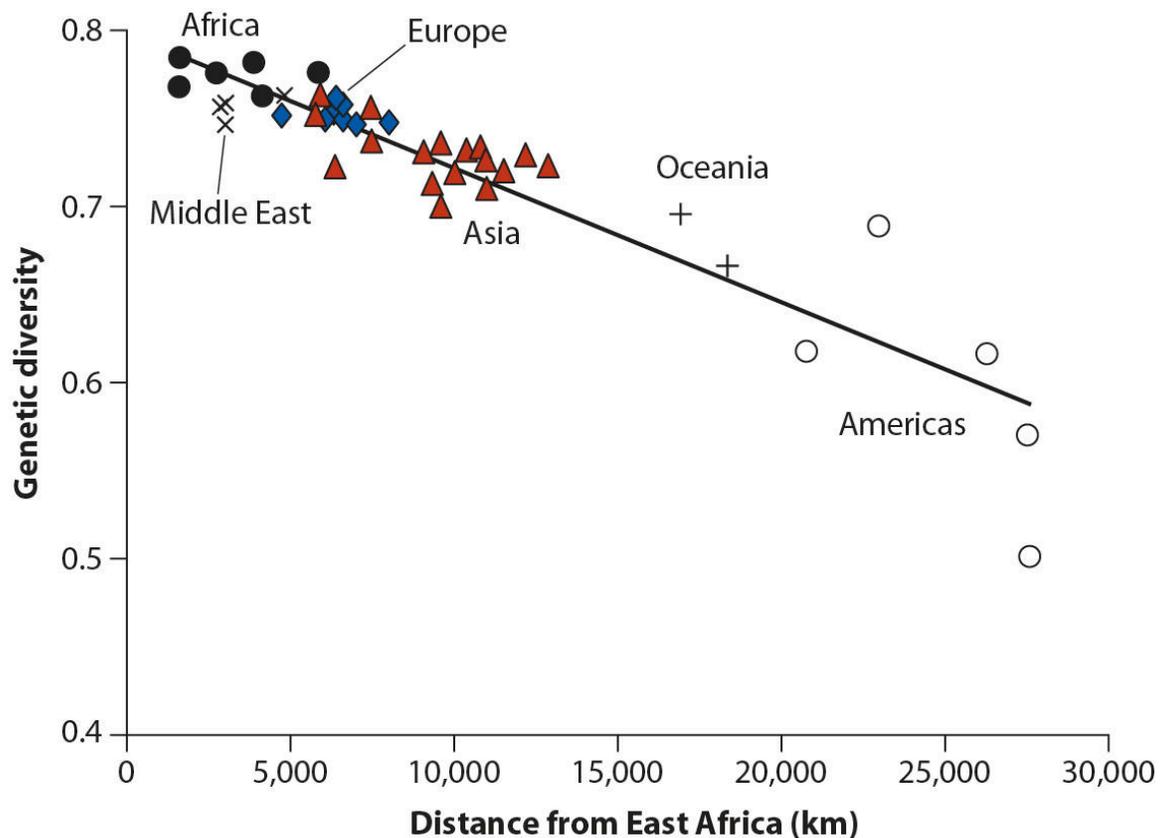


Figure 1: *Fig. 1. Relation entre la diversité génétique des populations humaines et leur distance par rapport à l'Afrique*

L'analyse de régression linéaire simple fait partie des méthodes statistiques (classiques) qui utilisent les propriétés mathématiques des distributions des variables, comme les tests paramétriques (voir capsule #8).

À ce titre, les données à analyser doivent respecter plusieurs conditions d'application :

1. La première est évidemment celle que l'on doit respecter (presque) toujours dans les analyses statistiques de base : les **échantillons doivent être aléatoires et indépendants**,
2. La deuxième est incluse dans le titre de la méthode : il doit exister une **relation linéaire entre les variables indépendante et dépendante**, ce qui peut se vérifier graphiquement. Cette condition est par ailleurs vérifiée si les conditions suivantes sont aussi vérifiées,
3. **L'homoscédasticité (homogénéité des variances) des résidus** de l'analyse. Les résidus sont simplement la différence entre la valeur observée et la valeur prédite par le modèle de régression linéaire. Ces résidus doivent être répartis de façon aléatoire et homogène de part et d'autre de la droite de régression et...
4. ...la distribution de ceux-ci doit respecter une distribution Normale de moyenne nulle, centrée sur la droite de régression. Il s'agit de la condition de **la normalité des résidus** de l'analyse.

La condition 2 peut bien sûr se vérifier *a priori* graphiquement, alors que les conditions 3 et 4 se vérifient *a posteriori* par des graphiques diagnostiques (voir capsule #9), mais aussi par des tests formels dont on verra un exemple ici.

! Important

L'analyse de régression linéaire utilise les propriétés mathématiques des données qui respectent ces conditions d'application pour déterminer les coefficients de la droite qui **minimise l'erreur** entre les valeurs observées de la variable réponse et les prédictions basées sur les valeurs de la variable explicative.

Nous pouvons illustrer ce principe avec le graphique ci-dessous :

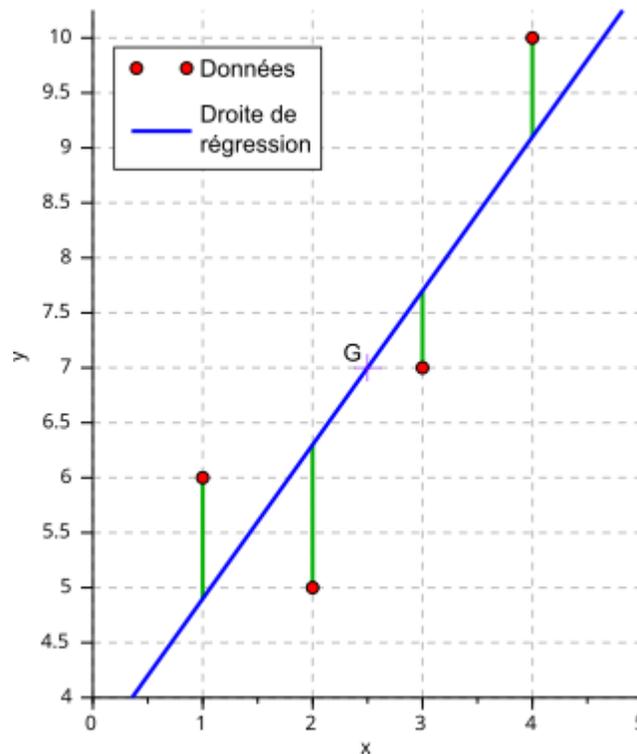


Figure 2: Fig. 2. Illustration de la droite des moindres carrés dont les coordonnées correspondent aux coefficients de la régression linéaire entre les variables explicative (x) et réponse (y). Image de Wikipedia.

Ici la **droite des moindres carrés** (en bleu) est celle qui **minimise les résidus** (en vert), c'est-à-dire la différence entre les observations (en rouge) et la valeur prédite par la droite de régression (en bleu). La méthode mathématique minimise la **somme des carrés des écarts (résidus)**, qu'on appelle aussi **l'erreur**.

Analyse de régression linéaire simple

Le but de cet exemple est de calculer et tester la significativité de la relation entre un taux de réaction enzymatique et la concentration de substrat.

- Vérification de la relation linéaire

On va d'abord vérifier l'allure de la relation entre les variables réponse (l'activité enzymatique) et explicative la concentration de substrat) :

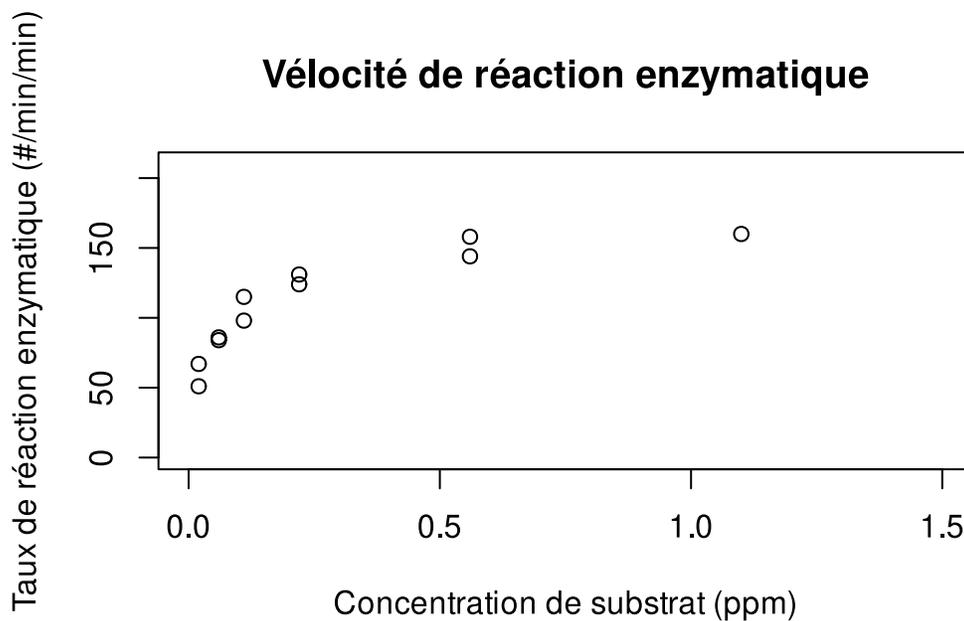
```
# Indices pour les deux groupes de traitement
un <- Puromycin$state == "untreated"
tr <- Puromycin$state == "treated"

rate1 <- Puromycin$rate[un]
concl <- Puromycin$conc[un]
```

```

# Graphique en nuage de point pour le groupe témoin (pas de puromycine)
plot(
  ratel ~ concl,
  xlim = c(0, 1.5),
  ylim = c(0, 210),
  xlab = "Concentration de substrat (ppm)",
  ylab = "Taux de réaction enzymatique (#/min/min)",
  main = "Vélocité de réaction enzymatique"
)

```

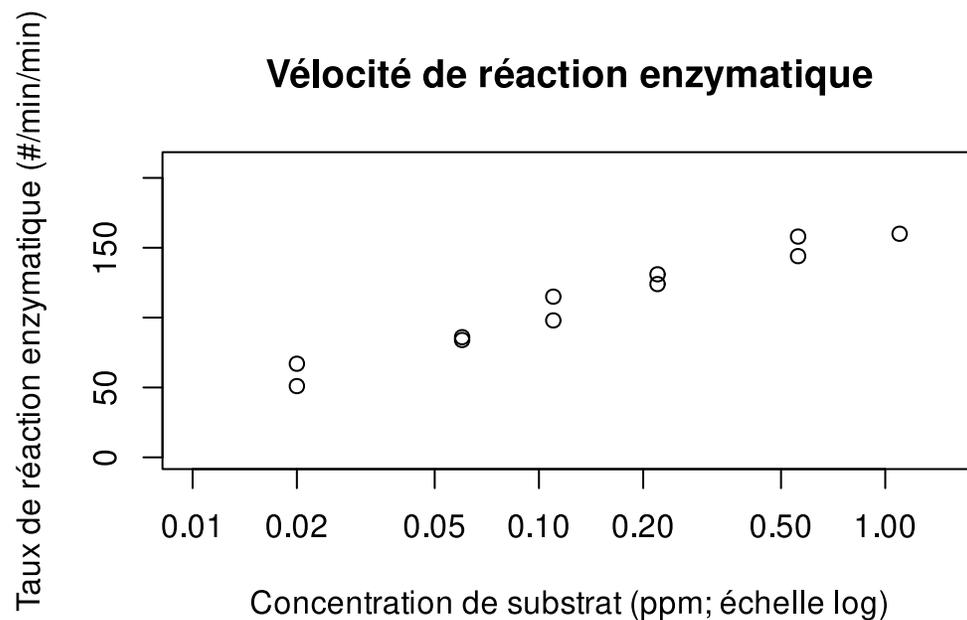


Il apparaît évident que la relation entre les variables n'est pas linéaire, mais notre connaissance de la cinétique enzymatique nous permet de comprendre qu'on pourrait obtenir une relation linéaire en transformant les valeurs de concentration en $\log()$!

```

# Graphique en nuage de point pour le groupe témoin (pas de puromycine)
plot(
  ratel ~ concl,
  xlim = c(0.01, 1.5),
  ylim = c(0, 210),
  xlab = "Concentration de substrat (ppm; échelle log)",
  ylab = "Taux de réaction enzymatique (#/min/min)",
  main = "Vélocité de réaction enzymatique",
  log = "x"
)

```



Ici l'utilisation de l'argument (option) `log = 'x'` permet de représenter automatiquement nos données selon une **échelle logarithmique sur l'axe des abscisses**.

On constate qu'on semble bien avoir une relation linéaire entre les concentrations de substrat transformées par un logarithme et les taux de réaction enzymatique.

! Important

Pour la suite de nos analyses, nous allons créer une variable transformée pour les valeurs de concentration avec laquelle nous pourrions travailler.

- Utilisation de la fonction `lm()`

L'analyse de régression linéaire fait partie de la grande famille des “**modèles linéaires**” en statistiques, d'où le nom de la fonction utilisée dans R : `lm()`

```
# Procéder à l'analyse de régression linéaire simple (une seule variable explicative)
# ATTENTION à bien utiliser les valeurs transformées pour les concentrations de substrat
logc1 <- log10(conc1)

regl <- lm(rate1 ~ logc1)

# Visualisation des résultats de l'analyse
summary(regl)
```

```

Call:
lm(formula = rate1 ~ logc1)

Residuals:
    Min       1Q   Median       3Q      Max
-8.034 -5.988 -2.677  7.616  9.969

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  164.588     4.266   38.59 2.62e-11 ***
logc1         62.129     4.156   14.95 1.16e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.575 on 9 degrees of freedom
Multiple R-squared:  0.9613,    Adjusted R-squared:  0.957
F-statistic: 223.5 on 1 and 9 DF,  p-value: 1.161e-07

```

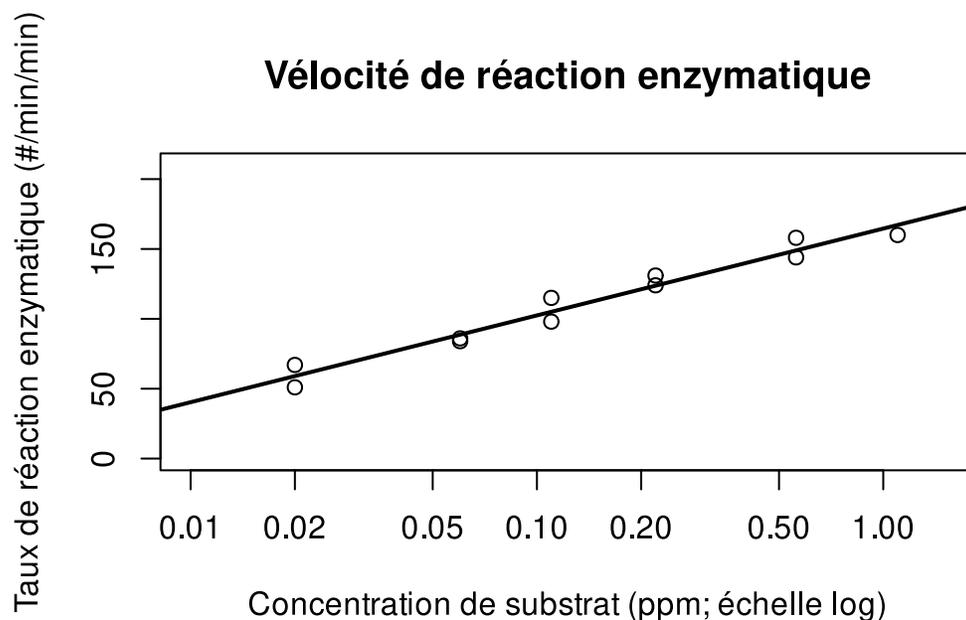
Les résultats de cette analyse sont très riches en informations :

1. Tout d'abord, comme pour tous les tests semblables dans R, on vous rappelle le modèle linéaire testé, c'est à dire ici le taux de réaction enzymatique en fonction du log de la concentration de substrat :`lm(formula = rate1~ logc[un])`
2. Ensuite on vous présente un résumé de la distribution des valeurs des résidus : Residuals
3. Puis viennent les informations relatives aux coefficients de la régression linéaire : Coefficients
 - La première colonne Estimate donne les valeurs estimées de l'ordonnée à l'origine (Intercept) et la pente `logc[un]` (toujours le nom de la variable explicative) de la droite de régression
 - La deuxième colonne Std. Error donne l'erreur standard associée à l'estimation de ces valeurs,
 - La troisième colonne t value donne la valeur de t utilisée pour tester les hypothèses nulles que ces valeurs ne sont pas significativement différentes de 0 à un seuil de confiance de 95%,
 - Finalement la dernière colonne $\text{Pr}(>|t|)$ donne la valeur de la p -value permettant de tirer une conclusion de ces tests. On rappelle que si la p -value est $<$ au seuil $\alpha = 0.05$, alors on peut rejeter l'hypothèse nulle et conclure que les valeurs sont significativement différentes de 0. La ligne immédiatement sous ce tableau n'est qu'une sorte de "légende" pour aider à interpréter la force de la significativité.
4. Deux valeurs sont principalement d'intérêt dans la dernière section : Adjusted R-squared et F-statistic
 - Adjusted R-squared est appelé le **coefficient de régression R^2** (à ne pas confondre avec les coefficients de l'équation linéaire !) et indique ici la proportion de variance expliquée par la variable... explicative. Ce n'est pas rare de voir une valeur de $R^2 > 0.9$ en biochimie, lorsqu'on produit des courbes d'étalonnage par exemple; C'est beaucoup plus rare en écologie !

- F-statistic est le résultat d'un test qui permet d'évaluer la significativité de la régression linéaire dans son ensemble, et pas seulement des paramètres de l'équation de la droite. Dans une régression linéaire simple, si la pente est significativement différente de 0, alors la p -value du test de F de la régression est < 0.05 et la régression est significative, comme dans ce cas-ci.

On peut visualiser très simplement notre droite de régression sur le graphique de nos données avec la fonction `abline()`:

```
# Ajout de la droite de régression suite à l'analyse
abline(regl, lwd = 2)
```

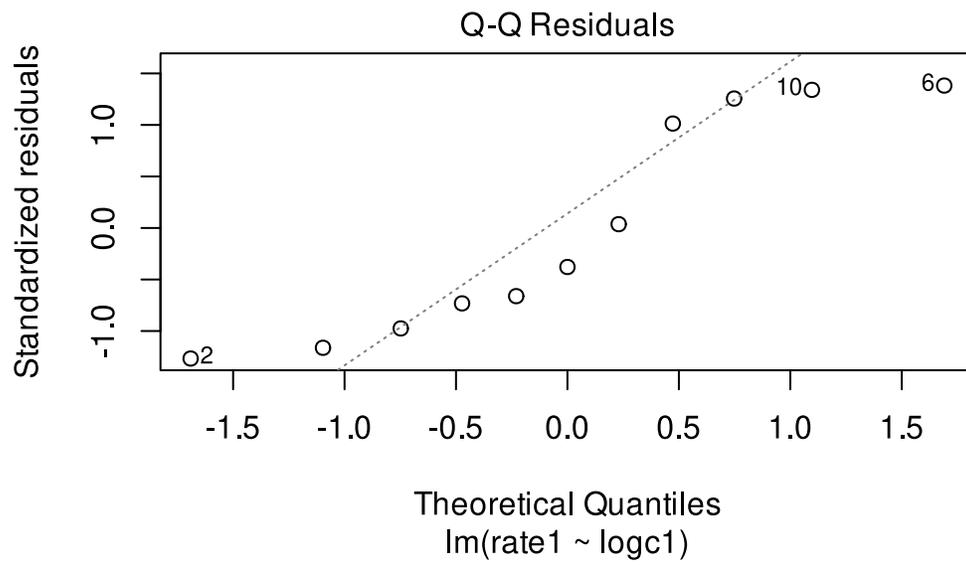
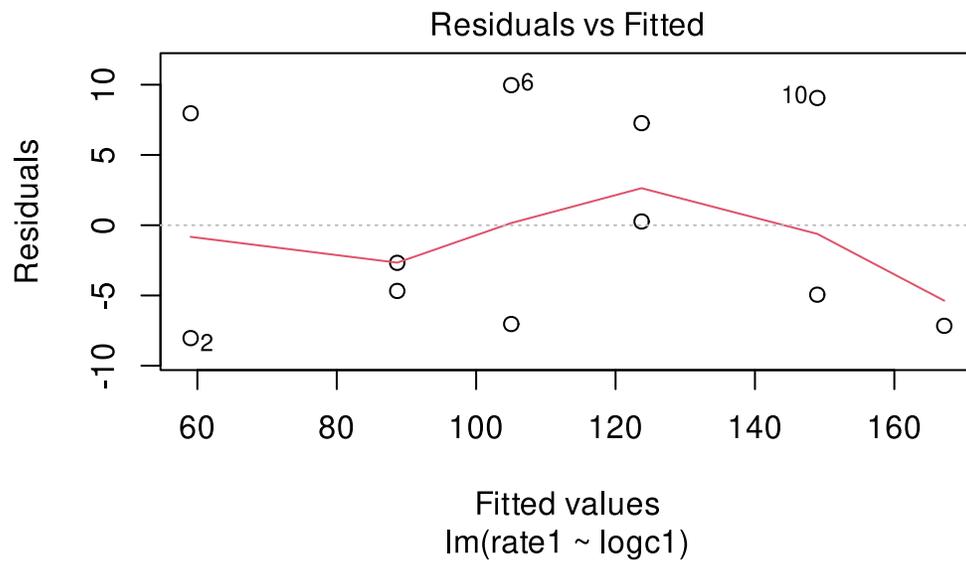


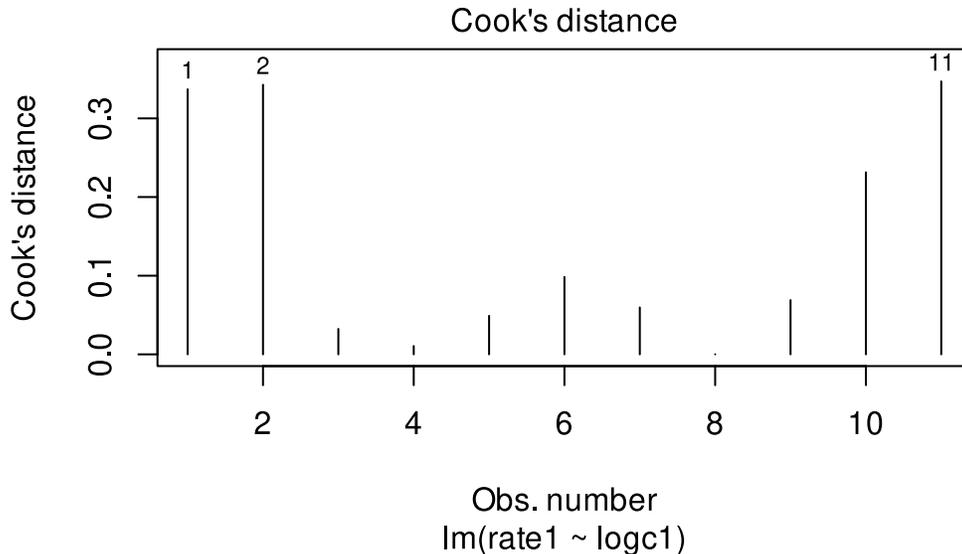
Vérifier *a posteriori* les conditions d'application de l'analyse de régression linéaire

On peut faire deux types de vérification (voir capsule #9) : visuelle et statistique

- Vérification **visuelle** des conditions d'application

```
plot(regl, which = c(1, 2, 4))
```





Nous utilisons les 3 graphiques diagnostiques produits automatiquement par la fonction `plot()`, comme on l'a vu dans la capsule #9.

1. Le premier nous montre que les résidus de la régression présentés en fonction de la valeur "prédite" par le modèle linéaire de régression sont apparemment **répartis symétriquement et aléatoirement de part et d'autre de l'horizon 0**.
2. Le deuxième semble montrer par contre que certaines valeurs de résidus **ne suivent pas tout à fait une distribution Normale** (pas alignés sur la droite).
3. Le troisième montre finalement qu'aucune observation ne produit une valeur de distance de Cook > 0.5 , ce qui confirme que notre analyse **n'est pas biaisée par une ou quelques valeurs extrêmes**. Une valeur > 0.5 est problématique.

Il apparaît donc de nos graphiques diagnostiques qu'en général nos données semblent respecter les conditions d'application, même si les conclusions ne sont pas toutes claires !

⚠ Warning

Le nombre de valeurs analysées est faible (11), donc les conclusions des graphiques diagnostiques sont nécessairement moins claires que pour des analyses avec de grands nombres de points.

- Vérification **statistique** des conditions d'application

La fonction `gvlma()` permet de vérifier en une fois si l'ensemble des conditions d'applications mathématiques sont remplies, grâce à plusieurs tests statistiques. Exemple :

```
library(gvlma)
gvlma(regl)
```

```
Call:
lm(formula = rate1 ~ logc1)
```

```
Coefficients:
(Intercept)      logc1
      164.59         62.13
```

```
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance = 0.05
```

```
Call:
gvlma(x = regl)
```

	Value	p-value	Decision
Global Stat	1.925e+00	0.7495	Assumptions acceptable.
Skewness	1.926e-01	0.6608	Assumptions acceptable.
Kurtosis	1.183e+00	0.2767	Assumptions acceptable.
Link Function	5.494e-01	0.4586	Assumptions acceptable.
Heteroscedasticity	6.932e-05	0.9934	Assumptions acceptable.

i Note

N'oubliez pas d'installer la librairie gvlma à l'aide de la fonction `install.packages()`.

Encore une fois, les résultats de cette analyse sont riches en informations :

1. D'abord on vous rappelle le modèle linéaire testé : Call
2. Ensuite les valeurs estimées des coefficients de la régression : Coefficients
3. Puis on clairement ce que cette fonction fait : ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
4. Finalement, on détaille les résultats des différents tests sur les résidus. Chacun doit être interprété de la même façon : une statistique de test calculée Value, suivie de la p -value associée p -value et de la décision suggérée (Assumptions acceptable. ou Assumptions NOT satisfied!)
 - Global Stat est un test sur la **conformité globale** de notre analyse aux conditions d'application, au cas où un des tests spécifiques ci-dessous ne serait pas concluant,
 - Skewness vérifie la **symétrie** de la distribution des résidus par rapport à 0,

- Kurtosis vérifie la **normalité** de la distribution des résidus ainsi que la présence de **valeurs extrêmes**,
- Link Function vérifie la **linéarité** de la relation entre les variables réponse et explicative,
- Heteroscedasticity vérifie **l'homogénéité de la variance** des résidus.

⚠ Warning

Tous ces tests graphiques et statistiques restent des **aides à la décision**. Vous devez **exercer votre jugement en tout temps** pour décider si vous pouvez vous fier aux résultats de vos analyses.

Prédictions grâce à la régression linéaire

Comme on l'a indiqué dès le début de cette capsule, la régression linéaire permet de faire des prédictions, c'est-à-dire d'estimer la valeur moyenne que devrait avoir une variable réponse pour toute valeur de la variable explicative.

Toutefois cela est vrai dans les limites observées de la variable explicative, autrement dit la régression permet de faire avec confiance de **l'interpolation**, et elle ne devrait être utilisée qu'avec grande prudence pour faire de **l'extrapolation**.

- Prédiction de la valeur moyenne

Il est très facile de faire une telle prédiction. En fait, il s'agit simplement d'utiliser l'équation de la droite de régression $y = a + bx$, avec a l'ordonnée à l'origine et b la pente.

On peut toutefois utiliser la fonction `predict()` si on veut faire plusieurs prédictions facilement, par exemple.

```
# Predire les taux de réaction enzymatique moyens pour des concentrations de
# substrat de 0.3, 0.35 et 0.4 ppm.
conx <- log10(c(0.3, 0.35, 0.4)) # Il ne faut pas oublier la transformation !
predict(regl, newdata = data.frame(logc1 = conx), interval = "confidence")
```

```
      fit      lwr      upr
1 132.1027 126.0071 138.1983
2 136.2620 129.8104 142.7136
3 139.8650 133.0728 146.6571
```

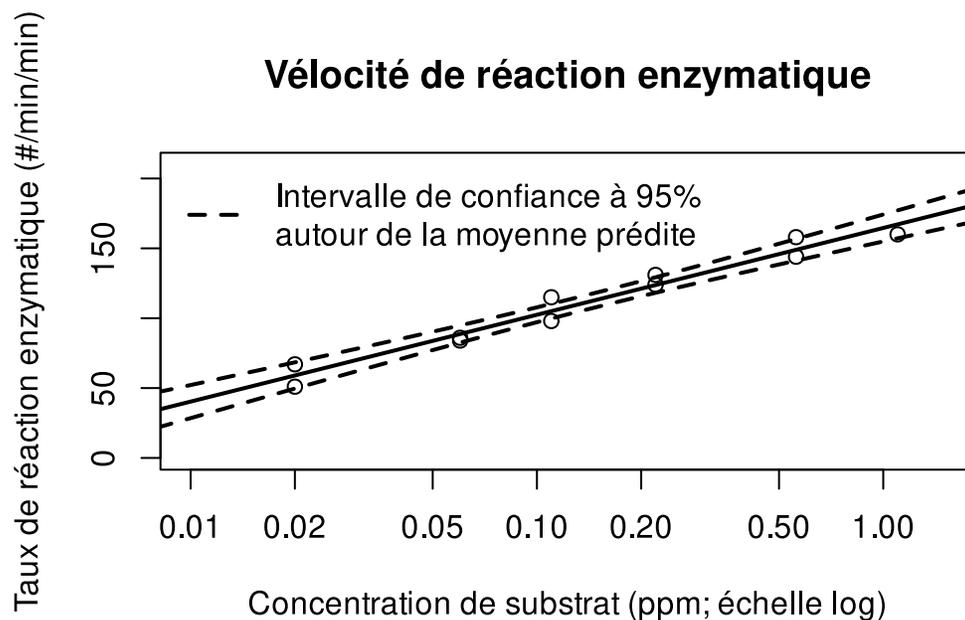
Les résultats de la fonction `predict()` donnent:

1. `fit` : la **valeur moyenne** prédite en x par la régression linéaire,
2. `lwr` : la **limite inférieure** de **l'intervalle de confiance à 95%** en x , selon la valeur de l'argument `interval =`
3. `upr` : la **limite supérieure** de l'intervalle de confiance à 95% en x , selon la valeur de cet argument.

L'ordre des lignes suit l'ordre des valeurs dans le vecteur x fourni à la fonction.

On remarque ici que la valeur de l'argument `interval = "confidence"`. Ceci permet de calculer **l'intervalle de confiance à 95% autour de la moyenne prédite**. Ça se traduit par ce genre de courbes :

```
# Ajout des intervalles de confiance
xint <- seq(-2.1, 1.1, 0.1)
pred1 <- predict(regl, newdata = data.frame(logcl = xint), interval =
"confidence")
lines(10^xint, pred1[, 2], lty = 2, lwd = 2)
lines(10^xint, pred1[, 3], lty = 2, lwd = 2)
legend(
  "topleft",
  legend = "Intervalle de confiance à 95%\nautour de la moyenne prédite",
  lty = 2,
  lwd = 2,
  bty = "n"
)
```



- Prédiction de l'étendue des valeurs individuelles

Il est de plus possible d'estimer l'intervalle à l'intérieur duquel 95% des valeurs individuelles de la variable réponse observée **devraient se retrouver** d'après l'analyse de régression.

Ceci peut être utile pour **détecter des valeurs individuelles extrêmes**, ou **prédire une plage de valeurs possibles de la variable réponse pour une valeur encore jamais observée de la variable explicative**, par exemple.

Il faut alors spécifier une autre valeur à l'argument `interval` : `interval = "prediction"`. Ceci donne :

```
# Ajout des intervalles de confiance
pred1 <- predict(regl, newdata = data.frame(logc1 = xint), interval =
"prediction")
lines(10^xint, pred1[, 2], lty = 2, lwd = 2, col = "red")
lines(10^xint, pred1[, 3], lty = 2, lwd = 2, col = "red")
legend("topleft", legend = "Intervalle de confiance à 95% de la\ndistribution
des valeurs individuelles\nd'activité enzymatique", lty = 2, lwd = 2, col =
"red", bty = "n")
```

