Examen R-Stats Geeft

Votre nom

2026-02-26

Contents

Déroulement de l'examen

Vous travaillerez de 9h15 à 12h30 au plus tard. Vous aurez accès au cours, à l'internet... La seule exigence est que vous n'utilisiez pas d'aide extérieure.

Vous devrez répondre aux questions dans ce fichier en ajoutant du texte et du code.

> Utilisez ce format pour le texte de vos réponses (sautez une ligne avant) et ajoutez le code dans des bouts de code standard:

```
# Code
```

> Commencez chaque paragraphe par > et tricotez souvent.

Vous devez répondre aussi clairement que possible, en rédigeant vos réponses. Votre code doit être facile à lire, avec des commentaires quand c'est utile, formaté correctement (attention aux espaces, aux indentations, à la clarté des noms des objets, etc.).

A la fin de votre travail, envoyez le fichier examen.Rmd à eric.marcon@agroparistech.fr

Les données nécessaires se trouvent dans le sous-dossier `data`.

L'examen comporte plusieurs questions à traiter :

- Une régression du nombre d'espèces en fonction de la surface pour tester la relation d'Arrhenius,
- Une ACP sur le climat de villes du monde entier,
- Un exercice de programmation avec R estimer le nombre d'espèces d'une communauté.

Relation d'Arrhenius

La relation d'Arrhenius (1921) prévoit que le nombre d'espèces d'un écosystème augmente avec sa surface à la puissance z selon l'équation

$$S(A) = cA^z$$

où $S(A)$ est le nombre d'espèces observées sur la surface A, c est une constante qui dépend des écosystèmes et du taxon considéré, et z est le paramètre d'intérêt. La valeur de z a fait l'objet d'une abondance littérature : la valeur théorique est $0,26$ (Preston, 1962) pour des îles de taille variable.

Les données disponibles (Johnson et Simberloff, 1974) sont le nombre d'espèces de plantes vasculaires (`species`) pour 42 îles britanniques en fonction de différents prédicteurs, incluant la surface de l'île en km^2 (`area`). Elles se trouvent dans le fichier `britain_species.csv` (attention, c'est un fichier américain).

Lecture (et affichage) des données :

```
library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.2.0     v readr     2.2.0
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.2     v tibble    3.3.1
## v lubridate 1.9.5     v tidyr     1.3.2
## v purrr     1.2.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
(britain_species <- read_csv("data/britain_species.csv"))
```

```
## Rows: 42 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): island
## dbl (6): area, elevation, soil_types, latitude, dist_britain, species
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 42 x 7
##    island       area elevation soil_types latitude dist_britain species
##    <chr>       <dbl>     <dbl>      <dbl>    <dbl>        <dbl>   <dbl>
##  1 Ailsa         0.8       340          1     55.3         14        75
##  2 Anglesey    712.        127          3     53.3          0.2     855
##  3 Arran       429.        874          4     55.6          5.2     577
##  4 Barra        18.4       384          2     57           77.4     409
##  5 Bressay      31.1       226          1     60.1        202.      177
##  6 Britain  229850.       1343         16     54.3          0      1666
##  7 Canna        12.7       210          1     57.1         40.6     300
##  8 Coll         74.1       103          3     56.6         14.5     443
##  9 Colonsay     44.8       143          1     56.1         31.1     482
## 10 Eigg         29         393          1     56.9         12.3     453
## # i 32 more rows
```

Modèle non logarithmique

Estimez le modèle de base, dans lequel la variable explicative (créez-la) est $A^{0.26}$. Attention : son ordonnée à l'origine est obligatoirement nulle.

Effectuez les vérifications des hypothèses, faites des figures, discutez.

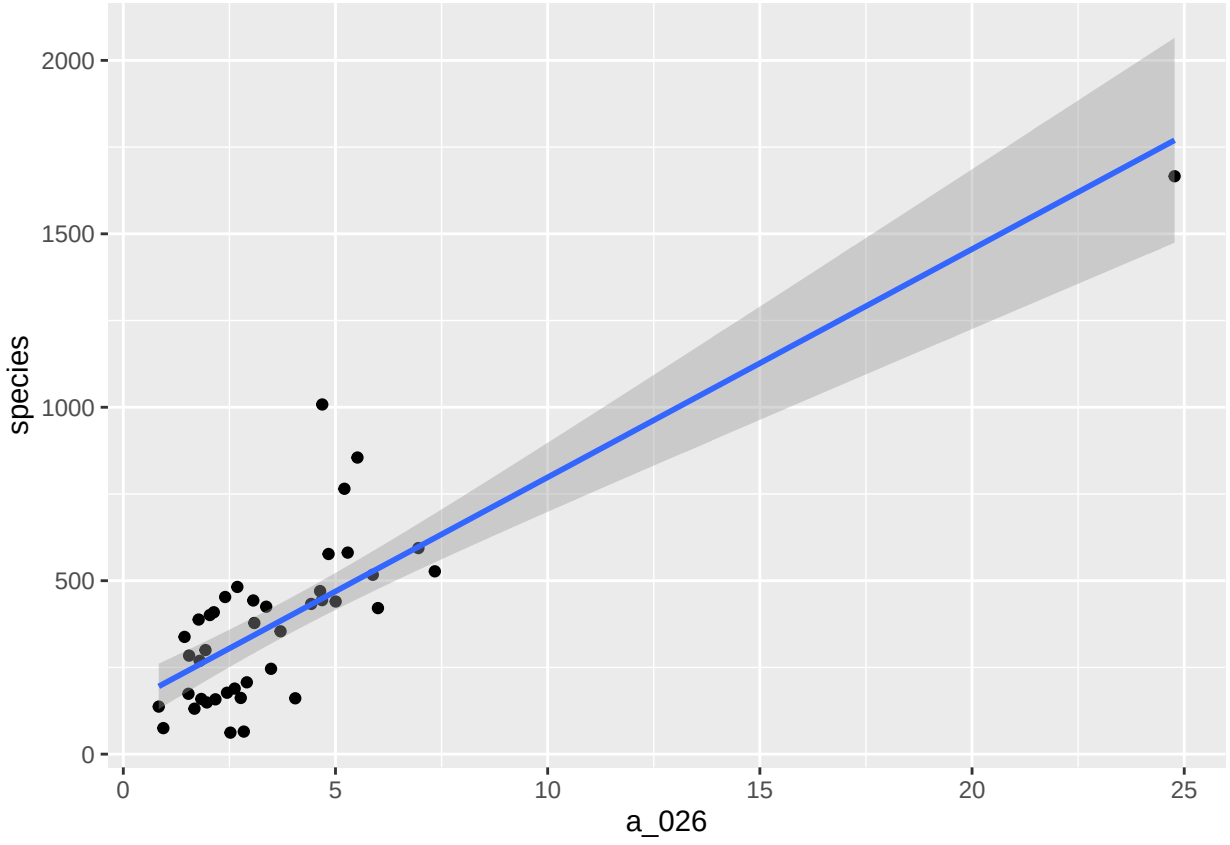
La variable `a_026` est ajoutée au tableau de données :

```
britain_species |>
  mutate(a_026 = area^0.26) ->
  britain_species_a026
```

Aperçu de la relation :

```
britain_species_a026 |>
  ggplot(aes(x = a_026, y = species)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Le modèle est le suivant :

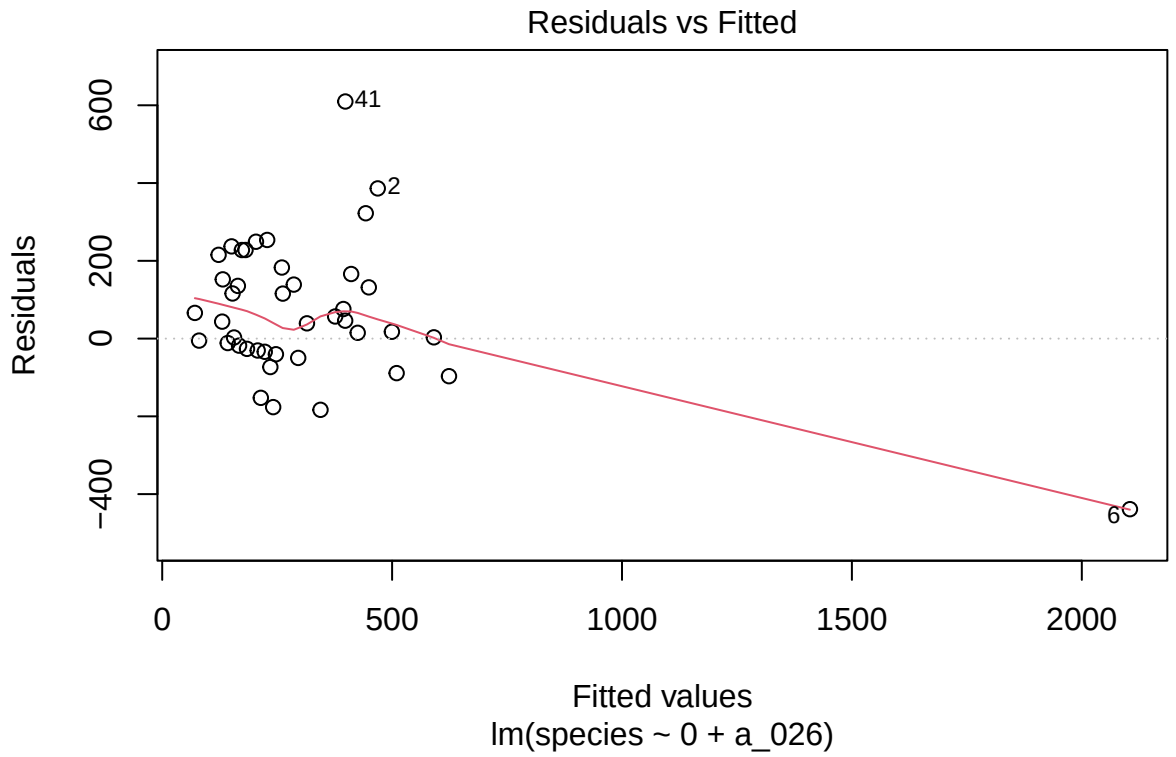
```
britain_species_lm <- lm(species ~ 0 + a_026, data = britain_species_a026)
summary(britain_species_lm)
```

```
##
## Call:
## lm(formula = species ~ 0 + a_026, data = britain_species_a026)
```

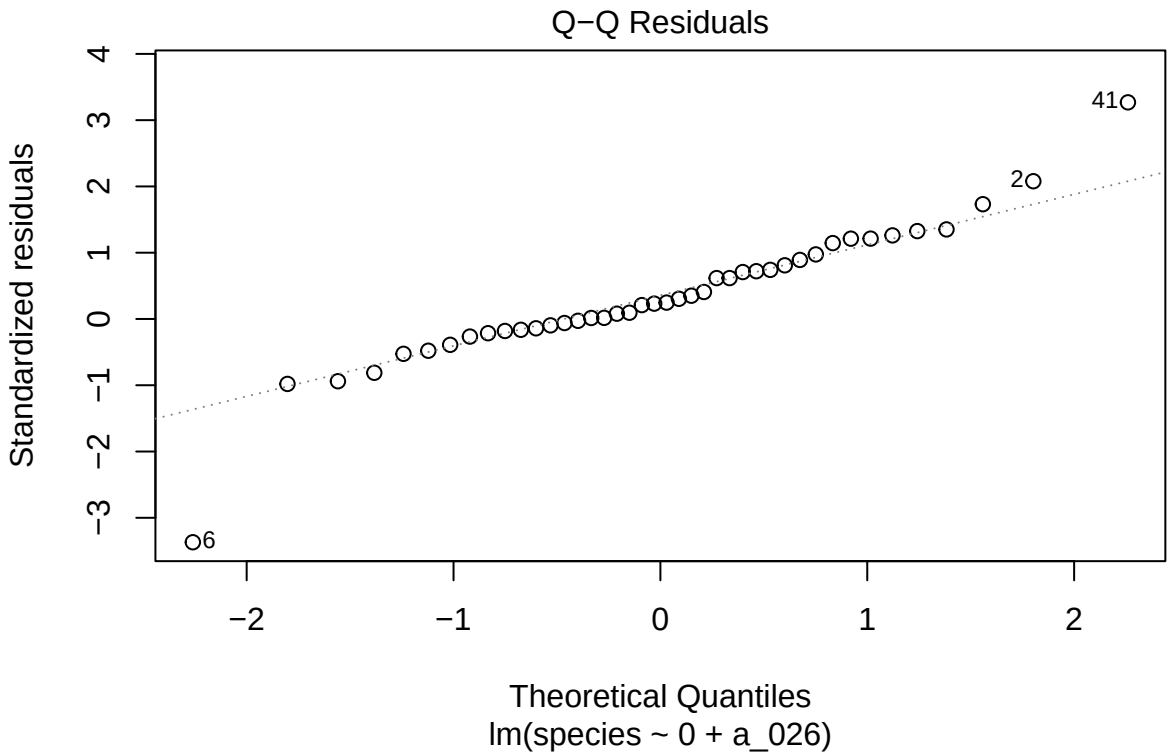
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -438.73  -29.59   44.83  162.62  609.73
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## a_026   84.961      5.489   15.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 188.3 on 41 degrees of freedom
## Multiple R-squared:  0.8539, Adjusted R-squared:  0.8503
## F-statistic: 239.6 on 1 and 41 DF,  p-value: < 2.2e-16
```

Vérification des hypothèses : Le point 6, la Grande Bretagne entière, pose problème parce que son erreur est grande.

```
plot(britain_species_lm, which = 1)
```




```
plot(britain_species_lm, which = 2)
```



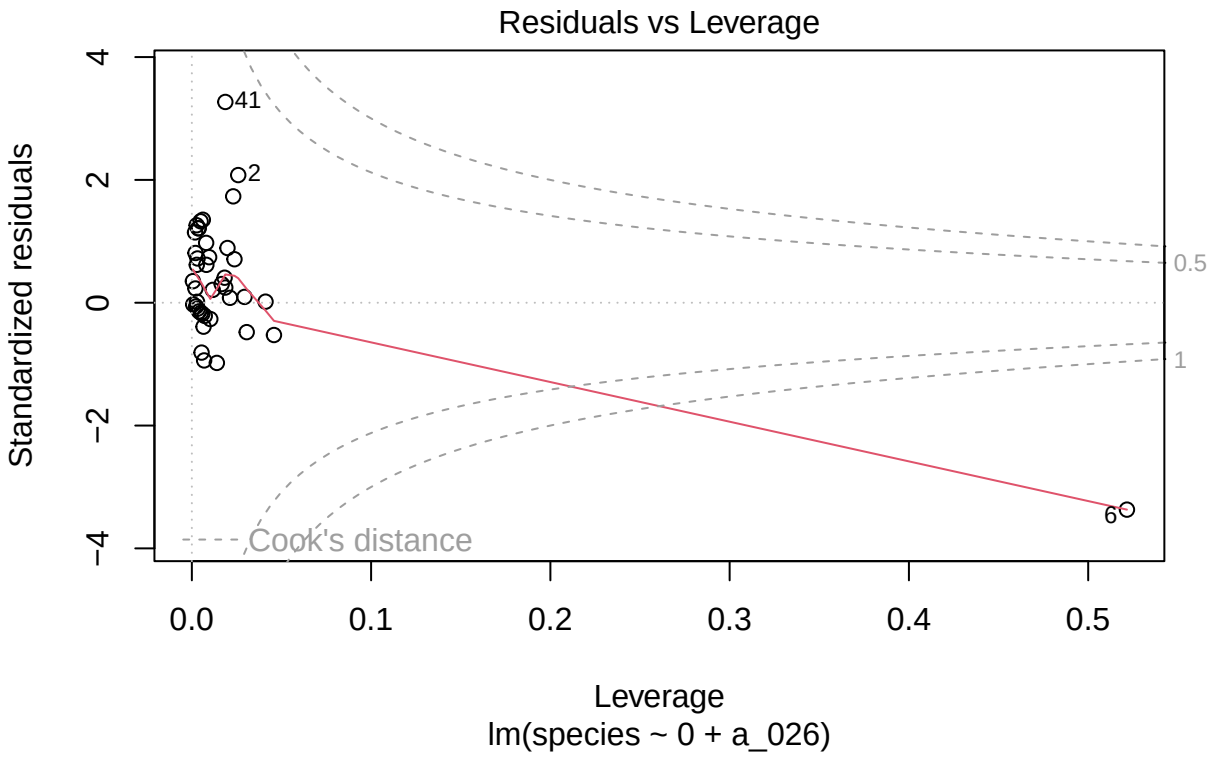
Les résidus sont approximativement normaux mais le point 6 à nouveau, comme le point 41, pose problème. Le test de Shapiro ne rejette pas la normalité.

```
shapiro.test(residuals(britain_species_lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(britain_species_lm)
## W = 0.96199, p-value = 0.174
```

L'effet de levier du point 6 est disproportionné : comme il se trouve loin du reste du nuage de points, il tire la régression à lui seul.

```
plot(britain_species_lm, which = 5)
```



En conclusion, le modèle respecte plus ou moins les hypothèses. Le seul paramètre est la pente de la droite, égale à 85 environ (valeur non interprétable facilement parce que la surface est à la puissance 0,26). La Grande-Bretagne entière a un effet de levier très grand, qui met en doute la robustesse du résultat.

Modèle logarithmique

Transformez le modèle original (où z n'est pas fixé) en en prenant le logarithme et recommencez. Quel est l'intérêt de cette transformation ?

La transformation logarithmique

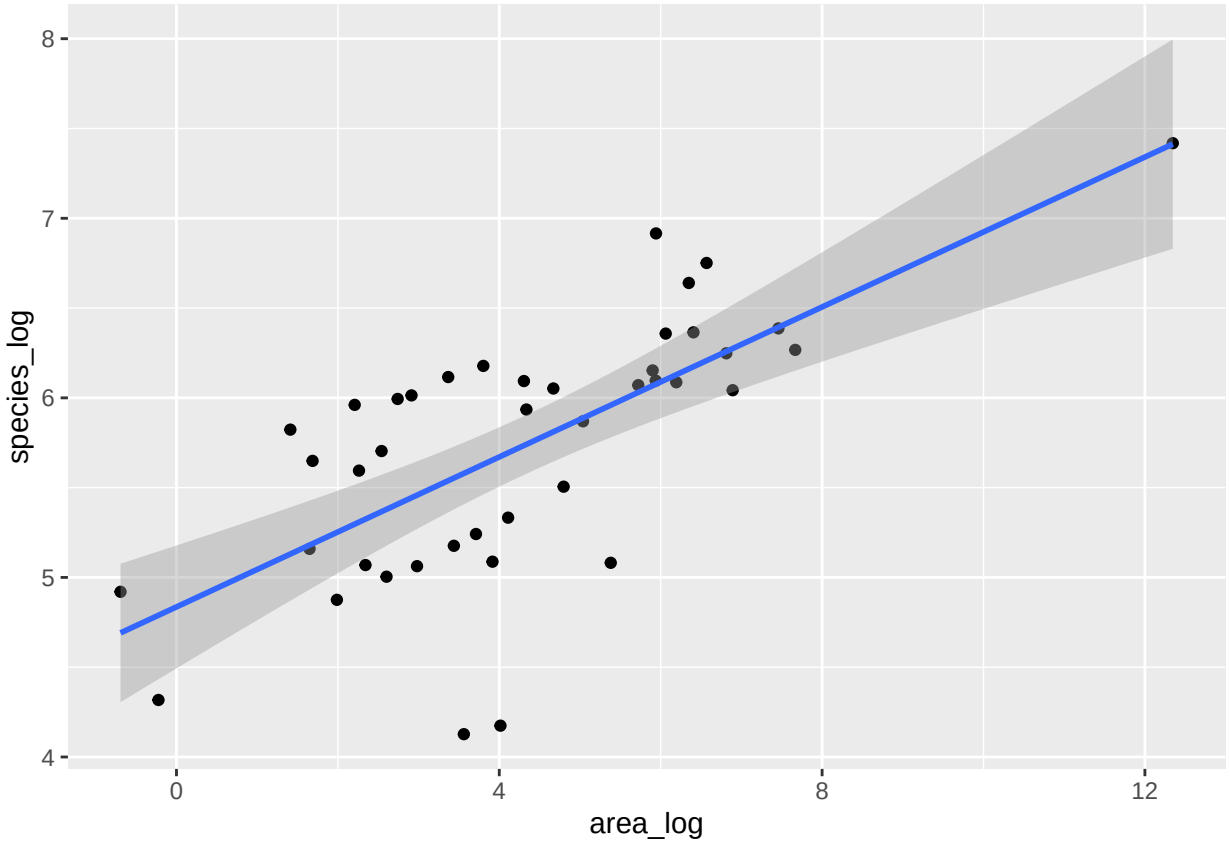
$$\ln(S) = \beta_0 + \beta_1 \ln(A)$$

permet d'estimer z au lieu de fixer sa valeur.

```
britain_species |>
  mutate(species_log = log(species), area_log = log(area)) ->
  britain_species_log

britain_species_log |>
  ggplot(aes(x = area_log, y = species_log)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



D'autre part, elle étale les petites valeurs et rapproche la Grande Bretagne du reste du nuage de points, ce qui limite son effet de levier.

Effectuez les vérifications des hypothèses, faites des figures, discutez le résultat.

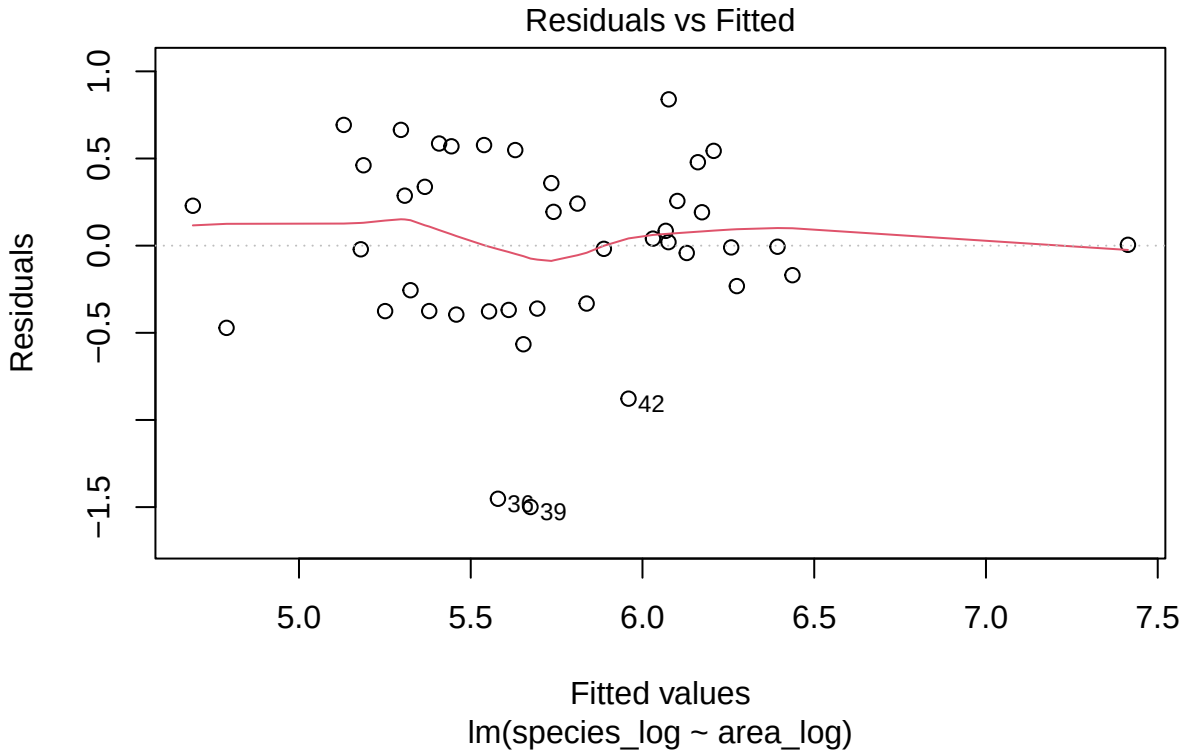
Le modèle est le suivant :

```
britain_species_loglm <- lm(species_log ~ area_log, data = britain_species_log)
summary(britain_species_loglm)
```

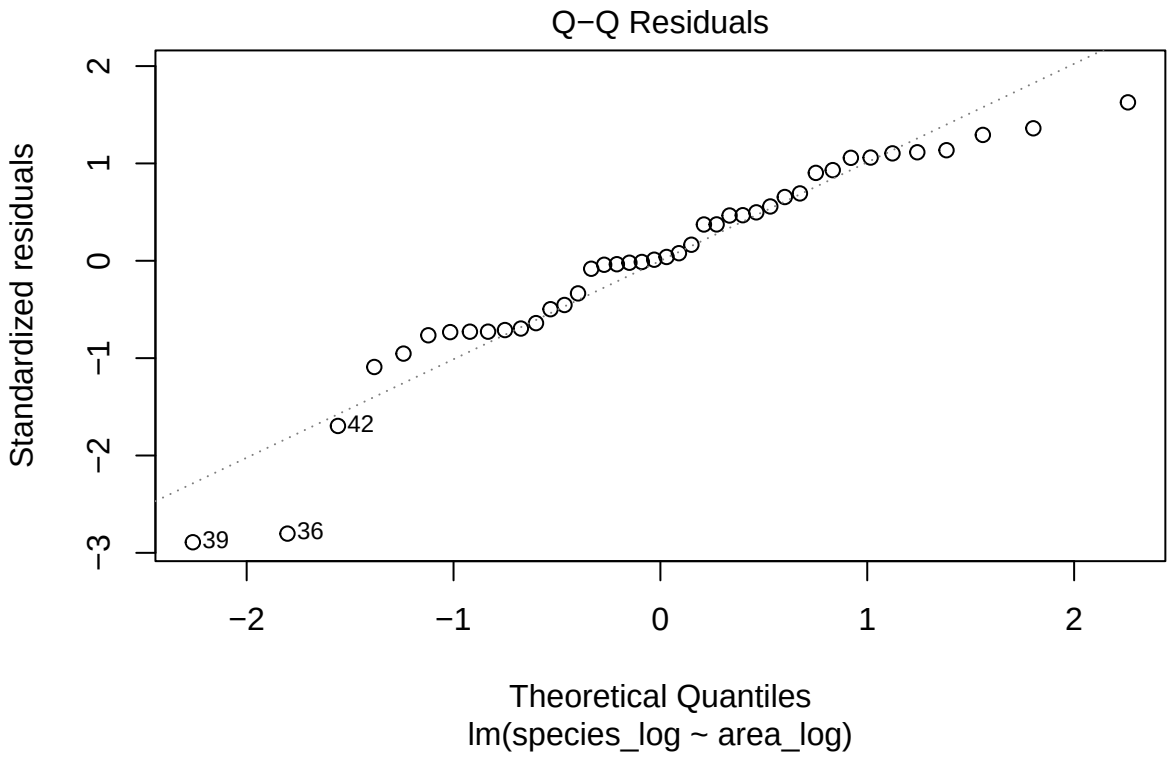
```
##
## Call:
## lm(formula = species_log ~ area_log, data = britain_species_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49954 -0.35374  0.01252  0.35354  0.83936
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.83570    0.16930  28.563  < 2e-16 ***
## area_log     0.20880    0.03447   6.057 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5251 on 40 degrees of freedom
## Multiple R-squared:  0.4784, Adjusted R-squared:  0.4653
## F-statistic: 36.69 on 1 and 40 DF,  p-value: 3.932e-07
```

Vérification des hypothèses : Les résidus sont maintenant homogènes.

```
plot(britain_species_loglm, which = 1)
```




```
plot(britain_species_loglm, which = 2)
```



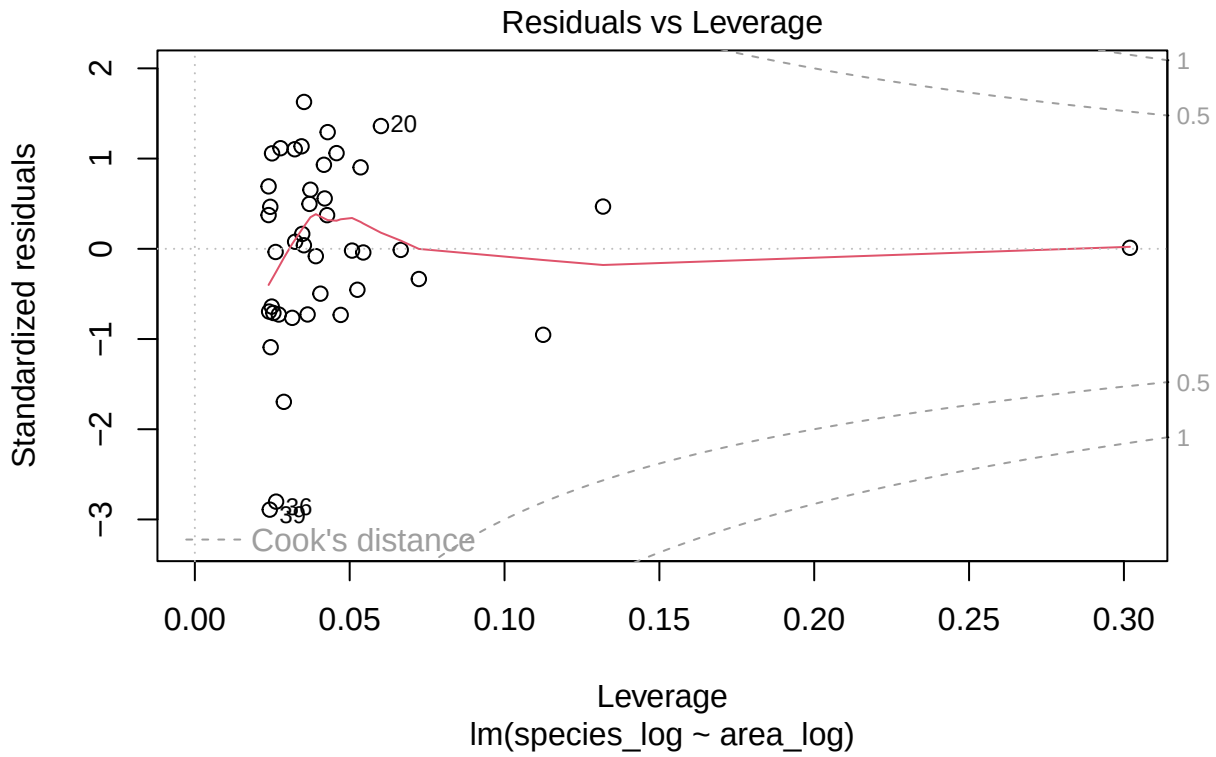
Le test de Shapiro rejette maintenant l'hypothèse de normalité. L'estimation de z sera correcte mais pas son intervalle de confiance. L'alternative est d'effectuer une régression sur les rangs mais la valeur de z ne sera pas utilisable.

```
shapiro.test(residuals(britain_species_loglm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(britain_species_loglm)
## W = 0.9298, p-value = 0.01275
```

L'effet de levier du point 6 reste grand mais sa distance de Cook est petite : il influe beaucoup sur la régression mais ne perturbe pas son résultat : son résidu est presque nul.

```
plot(britain_species_loglm, which = 5)
```



La valeur de z finalement retenue est 0,20. Quand la surface d'une île est multipliée par 10, le nombre d'espèces est multiplié par $2^{0.2}$, environ 1,6.

Covariables

La théorie prévoit que la biodiversité diminue avec la latitude, l'altitude et la distance au continent (qui est ici la Grande-Bretagne).

Ajoutez ces variables (non transformées) au modèle logarithmique, et sélectionnez le meilleur modèle selon le critère AIC.

Modèle complet :

```
library("MASS")
stepAIC(lm(species_log ~ area_log + dist_britain + latitude + elevation, data = britain_species_log))
```

```
## Start:  AIC=-74.04
## species_log ~ area_log + dist_britain + latitude + elevation
##
##                Df Sum of Sq    RSS     AIC
## - elevation     1    0.0329 5.7118 -75.796
## - dist_britain  1    0.0525 5.7314 -75.652
## <none>                      5.6789 -74.038
## - latitude      1    3.1658 8.8447 -57.430
## - area_log      1    3.5558 9.2347 -55.617
##
## Step:  AIC=-75.8
## species_log ~ area_log + dist_britain + latitude
```

```
##
##                Df Sum of Sq     RSS     AIC
## - dist_britain  1    0.0412  5.7530 -77.494
## <none>                       5.7118 -75.796
## - latitude      1    3.1681  8.8799 -59.263
## - area_log      1    8.4208 14.1326 -39.746
##
## Step:  AIC=-77.49
## species_log ~ area_log + latitude
##
##            Df Sum of Sq    RSS     AIC
## <none>                   5.753 -77.494
## - latitude  1    5.2751 11.028 -52.163
## - area_log  1    8.3854 14.138 -41.729
```

```
##
## Call:
## lm(formula = species_log ~ area_log + latitude, data = britain_species_log)
##
## Coefficients:
## (Intercept)     area_log     latitude
##     13.6108       0.1914      -0.1519
```

Le modèle retenu par la méthode stepwise contient le logarithme de la surface et la latitude. z est estimé à 0,19 et le nombre d'espèces diminue avec la latitude.

Discutez.

L'AIC du modèle augmente énormément quand on retire la covariable latitude. On peut afficher le détail du modèle avec latitude pour le comparer au précédent.

```
summary(lm(species_log ~ area_log + latitude, data = britain_species_log))
```

```
##
## Call:
## lm(formula = species_log ~ area_log + latitude, data = britain_species_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.19844 -0.12273  0.05292  0.24854  0.50252
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.61084    1.47264   9.243 2.27e-11 ***
## area_log     0.19138    0.02538   7.540 3.94e-09 ***
## latitude    -0.15188    0.02540  -5.980 5.52e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3841 on 39 degrees of freedom
## Multiple R-squared:  0.7279, Adjusted R-squared:  0.7139
## F-statistic: 52.16 on 2 and 39 DF,  p-value: 9.495e-12
```

La variance expliquée augmente beaucoup en ajoutant la latitude au modèle original : 72% au lieu de 48% La relation d'Arrhenius est définie pour des écosystèmes similaires, dont seule la taille varie. En Grande-Bretagne, la latitude est un déterminant important de la biodiversité : le climat varie énormément du nord au sud du pays. Après avoir contrôlé pour cette covariable essentielle,

la relation d'Arrhenius est validée par les données, avec une puissance z égale à 0,19, inférieure à la valeur attendue. Une raison est peut-être que les surfaces disponibles pour les végétaux (non anthropisées) ne sont qu'une partie de la surface des îles : A est mal mesuré, et la perte de surface augmente avec la taille des îles (les petites îles sont inhabitées alors que la Grande-Bretagne entière est largement occupée par l'agriculture et l'urbanisation). L'étape suivante serait donc de préciser les mesures de surface.

Estimation de la richesse spécifique

L'objectif de cet exercice est d'écrire une fonction qui accepte comme argument unique un vecteur d'abondances (nombres d'individus de plusieurs espèces) et renvoie le nombre d'espèces estimé.

L'estimateur du jackknife (Burnham et Overton 1978, 1979) estime le nombre d'espèces totales d'une communauté comme le nombre d'espèces observées auquel s'ajoute le nombre d'espèces observées une seule fois (dit autrement, on manque autant d'espèces qu'on en a vues une seule fois). Par exemple, si l'inventaire est le suivant

```
abondances <- c(5, 3, 1, 10, 1, 0)
```

alors le nombre d'espèces estimé est 7 (pensez à ne pas compter les espèces dont l'effectif est zéro).

Fonction

Ecrivez une fonction que vous appelerez `jackknife()` qui renverra l'estimation décrite ci-dessus.

La fonction s'écrit :

```
jackknife <- function(abondances) {
  especes_observees <- sum(abondances > 0)
  especes_non_observees <- sum(abondances == 1)
  return(especes_observees + especes_non_observees)
}
```

Testez-la avec le vecteur `abondances`.

Test de la fonction :

```
jackknife(abondances)
```

```
## [1] 7
```

Données de Paracou

L'inventaire de la parcelle 6 de Paracou est dans le fichier `Paracou6.csv`. C'est un fichier français.

Lecture (et affichage) des données :

```
(paracou6 <- read_csv2("data/Paracou6.csv"))
```

```
## i Using "',' as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
## Rows: 3541 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr (3): Family, Genus, Species
## dbl (5): SubPlot, idTree, Xfield, Yfield, CircCorr
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
## # A tibble: 3,541 x 8
##    SubPlot idTree Xfield Yfield Family           Genus          Species CircCorr
##      <dbl>  <dbl>  <dbl>  <dbl> <chr>            <chr>          <chr>      <dbl>
##  1       1 100655    7.5   180. Peraceae         Pogonophora    schomb~     44
##  2       1 100657    8     184. Peraceae         Pogonophora    schomb~     43.5
##  3       1 100658    5.5   182. Chrysobalanaceae Licania        membra~     53.5
##  4       1 100659    1.5   186  Euphorbiaceae    Sandwithia     guyane~     38.5
##  5       1 100660    0.5   190  Fabaceae         Eperua         falcata     77
##  6       1 100661    9     196. Celastraceae     Maytenus       oblong~     49.5
##  7       1 100662    9.5   196. Sapindaceae      Indet.Sapinda~ Indet.      57.5
##  8       1 100664    8     205  Myristicaceae    Iryanthera     sagoti~     50
##  9       1 100665    2.5   206  Fabaceae         Eperua         falcata    167
## 10       1 100666    2     204  Clusiaceae       Moronobea      coccin~     62.5
## # i 3,531 more rows
```

Comptez le nombre d'individus par espèce et faites-en un vecteur que vous passerez à votre fonction jackknife pour estimer la richesse spécifique de la communauté dont la parcelle 6 est un échantillon. Vous aurez besoin de la fonction `pull()` pour transformer une colonne d'un tibble en vecteur.

Les espèces sont définies par le binome genre-espèce, à utiliser pour regrouper les arbres :

```
paracou6 %>%
  # Regroupement des arbres par espèce
  group_by(Genus, Species) %>%
  # Comptage
  summarise(abondances = n()) -> abondances_paracou6
```

```
## `summarise()` has regrouped the output.
## i Summaries were computed grouped by Genus and Species.
## i Output is grouped by Genus.
## i Use `summarise(.groups = "drop_last")` to silence this message.
## i Use `summarise(.by = c(Genus, Species))` for per-operation grouping
##   (`?dplyr::dplyr_by`) instead.
```

```
# Affichage
abondances_paracou6
```

```
## # A tibble: 335 x 3
## # Groups:   Genus [169]
##    Genus      Species              abondances
##    <chr>      <chr>                     <int>
##  1 Abarema    jupunba var. jupunba         10
##  2 Abarema    mataybifolia                  4
##  3 Albizia    pedicellaris                  3
##  4 Amaioua    guianensis                    2
##  5 Amanoa     congesta                      1
##  6 Amanoa     guianensis                    2
##  7 Ambelania  acida                         4
##  8 Amphirrhox longifolia                    3
##  9 Anacardium spruceanum                    3
## 10 Anaxagorea dolichocarpa                  2
## # i 325 more rows
```

335 espèces sont inventoriées dans le jeu de données. Certaines sont mal déterminées ("Indet." dans le nom du genre ou de l'espèce) mais nous n'avons pas d'information pour les traiter différemment des autres.

La fonction jackknife est utlisée pour ajouter le nombre d'espèces vues une seule fois :

```
abondances_paracou6 %>%
  # Extraction du vecteur des abondances
  pull(abondances) %>%
  # Estimation de la richesse
  jackknife()
```

```
## [1] 433
```

Le nombre d'espèces estimé est 433.

Climat des villes

Le fichier `cities_climate.csv` contient des données climatiques tirées de WorldClim pour 49 grandes villes.

Ce sont :

- `t_mean`: Température moyenne annuelle.
- `t_diu`: Variation journalière de température, différence moyenne entre la température maximale et minimale dans un même mois.
- `t_sd`: Variation saisonnière de température, écart-type de la température moyenne entre les mois.
- `t_max`: Température maximale du mois le plus chaud.
- `t_min`: Température minimale du mois le plus froid.
- `p_ann`: Précipitation annuelle.
- `p_max`: Précipitation du mois le plus humide.
- `p_min`: Précipitation du mois le plus sec.
- `p_cv`: Coefficient de variation (ratio écart-type / moyenne) de la précipitation entre les mois.

Toutes les températures sont en °C et toutes les variables de précipitation (sauf le coefficient de variation) sont en mm.

La question à traiter est celle des composantes les plus importantes du climat. L'ACP centrée et réduite est la bonne méthode ici.

- Lisez le fichier de données (attention : il est américain) et inspectez-le.

  Lecture des données

```
cities_climate <- read_csv("data/cities_climate.csv")
```

```
## Rows: 49 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): city
## dbl (11): long, lat, t_mean, t_diu, t_sd, t_max, t_min, p_ann, p_max, p_min,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

- la fonction `prcomp()` de *stats* a besoin d'un tableau avec des noms de ligne. Vous devez donc préparer les données :
  - mettez de côté dans un vecteur les noms des villes,
  - éliminez du dataframe les colonnes qui ne sont pas des variables climatiques (les 4 premières),
  - nommez les lignes du dataframe : `rownames(nom_du_tableau) <- vecteur_des_noms_de_villes`.

```
# Nom des villes
villes <- cities_climate$city
# Réduction du tableau
```

```r
cities_climate <- cities_climate[, -(1:4)]
# Noms
rownames(cities_climate) <- villes
```

```
## Warning: Setting row names on a tibble is deprecated.
```

Faites l'ACP :

```r
cities_climate_pca <- prcomp(cities_climate, scale. = TRUE)
```

- Affichez les valeurs propres, justifiez la sélection des deux premiers axes seulement,

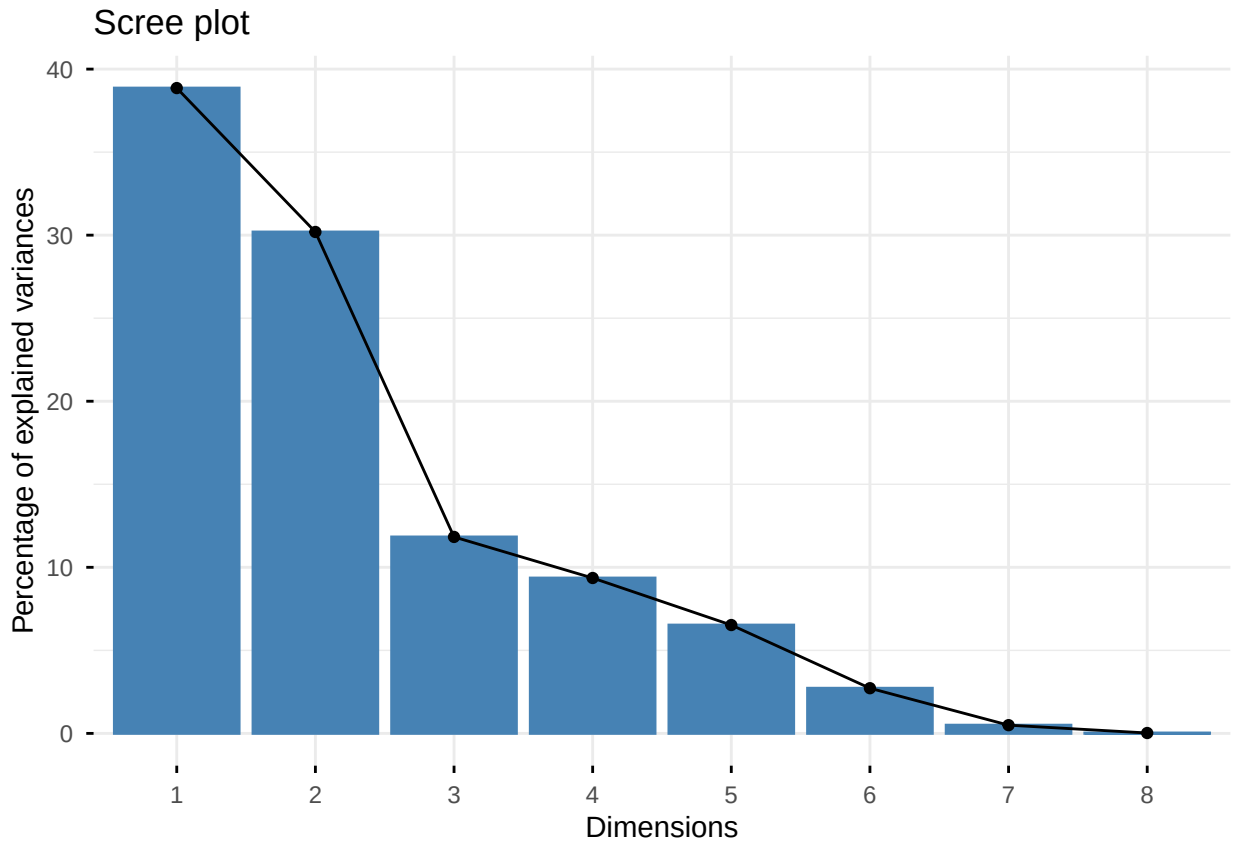
  Valeurs propres :

```r
library("factoextra")
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
fviz_eig(cities_climate_pca)
```

```
## Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
## Ignoring empty aesthetic: `width`.
```

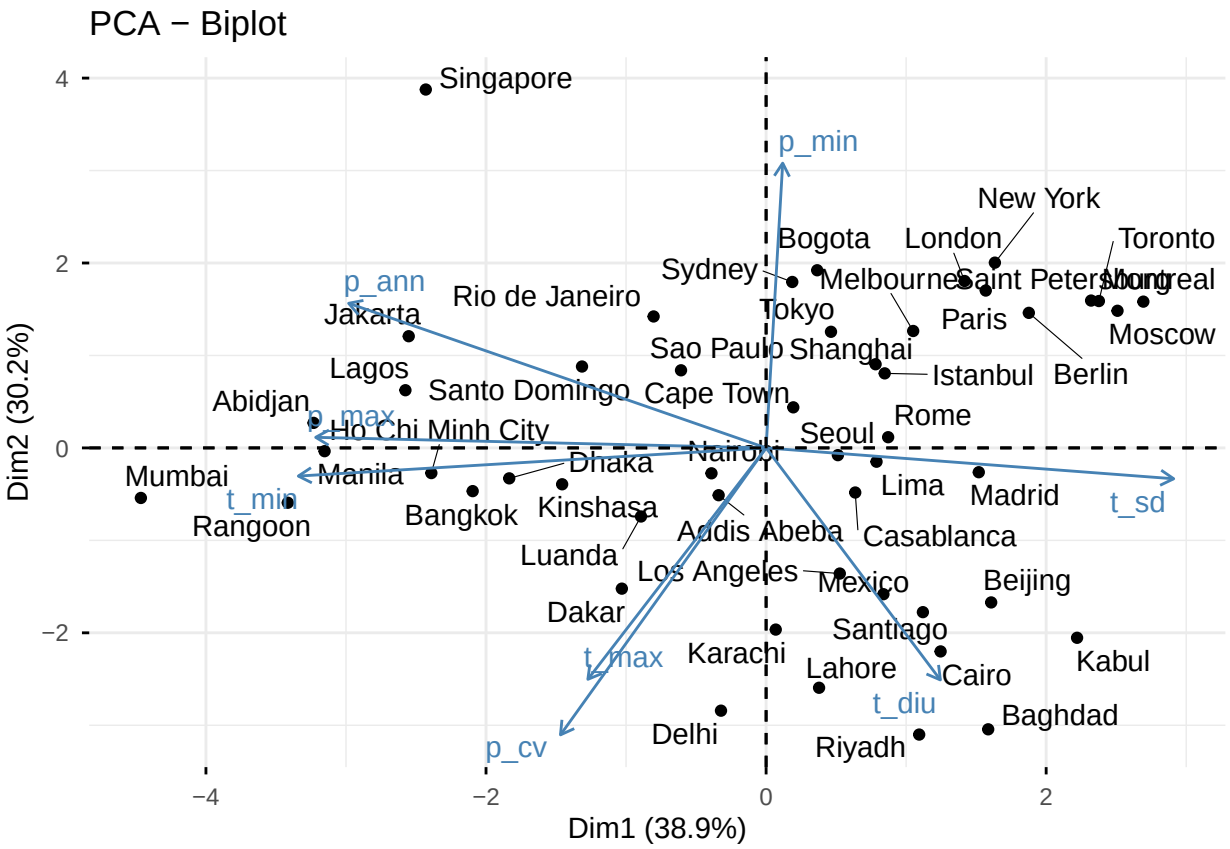


Les deux premières composantes principales résument la majorité de l'information (près de 70%). La troisième valeur propre est proche de la valeur moyenne en absence de structuration des données (1/8) : nous retiendrons seulement les deux premiers axes.

- Faites un biplot et interprétez : quels sont les gradients importants et les villes intéressantes ?

  Biplot :

```
fviz_pca_biplot(cities_climate_pca, repel = TRUE)
```



Le premier axe est généré par les variables `t_sd` (la continentalité), et `t_min` (température la plus froide) et `p_max` (précipitations maximales.). De la gauche vers la droite, le gradient va donc de tropical à continental. Le deuxième axe représente les variables “inverses”, `t_max` et `p_min`, la variabilité des précipitations et dans une moindre mesure la température diurne. Les villes tropicales sont par exemple Mumbai et Abidjan. Les villes les plus continentales sont par exemple Moscou et Kaboul.

Du bas vers le haut, le gradient va d’un climat chaud (température moyenne et maximale élevées), avec des précipitations inégales, à un climat froid avec sans mois très sec. La tendance est moins nette sur l’axe 2 que sur l’axe 1. La ville de Singapour se distingue par des précipitations abondantes toute l’année (`p_ann`), avec un minimum élevé.