

PROJET TUTEORE (M2 SSD)

Saoudi Khaoula

2025-01-13

Analyse descriptives - UT4M Cardio

L'objectif de ce script est de proposer des visuels permettant de se familiariser avec des données UT4M sur le comportement cardiaques des ultra-trailers. Nous allons tout d'abord analyser, course par course, l'évolution temporelle des paramètres physiologiques pour identifier des variations significatives. Ensuite, nous comparerons les données entre différentes courses afin de mettre en évidence les différences liées aux distances parcourues.

Les bibliothèques nécessaires

Import et traitement des données

Importation des données

Dans cette première étape, nous importons les données de la feuille 'Tableau_gl_new' du fichier Excel contenant les informations complètes de l'étude UT4M.

```
## # A tibble: 6 x 268
##   CODE_SUJET COURSE  AGE POIDS_INC TAILLE EXP_TRAIL TRAINING CL_SCRA TOT_SCRA
##   <chr>         <dbl> <dbl>      <dbl>  <dbl>      <dbl>    <dbl>    <dbl>    <dbl>
## 1 AMJE41         100   26        72    176         4      3.5     461     474
## 2 BELA77         100   51        64    178         4      3.5     248     474
## 3 BLLI42         100   43        65    175         4      2.5     261     474
## 4 BOAN44         100   26        51    165         4      3      410     474
## 5 CAJE22         100   36        78    172         4      4      112     474
## 6 GODA23         100   30        75    181         4      3      154     474
## # i 259 more variables: CL_SEH <dbl>, TOT_SEH <dbl>, CL_SEF <dbl>,
## #   TOT_SEF <dbl>, CL_V1H <dbl>, TOT_V1H <dbl>, CL_V1F <dbl>, TOT_V1F <dbl>,
## #   CL_V2H <dbl>, TOT_V2H <dbl>, CL_V2F <dbl>, TOT_V2F <dbl>, DDC <dbl>,
## #   HDA <dbl>, PAS_PRE <chr>, PAD_PRE <chr>, PAM_PRE <chr>, SC <chr>,
## #   FC_PRE <chr>, S_PRE <chr>, DTDVG_PRE <chr>, DTDVGI_PRE <chr>, PP_PRE <chr>,
## #   calcul...33 <chr>, DTS_PRE <chr>, FR_PRE <chr>, FE_PRE <chr>,
## #   'H/R_PRE' <chr>, MVG_PRE <chr>, MVGI_PRE <chr>, E_PRE <chr>, ...
```

Renommer les variables “VOL_ODI_PRE” “VOL_OGI_PRE”

Sélection des colonnes d'intérêt

Dans cette étape, nous allons sélectionner les variables d'intérêt dans notre jeu de données. Il s'agit des mesures physiologiques clés associées à différents moments.

```
## # A tibble: 6 x 42
##   COURSE CODE_SUJET FC_PRE VOLOGI_PRE VOLODI_PRE VTDVGI_PRE VTSVGI_PRE FE2D_PRE
##   <dbl> <chr>      <chr> <chr>      <chr>      <chr>      <chr>      <chr>
## 1    100 AMJE41    58    27.2727272~ 24.598930~ 72.192513~ 31.550802~ 58
## 2    100 BELA77    60    24.5810055~ 32.960893~ 61.452513~ 22.905027~ 63
## 3    100 BLLI42    55    33.3333333~ 23.333333~ 50.555555~ 24.444444~ 52
## 4    100 BOAN44    54    28.1045751~ 20.915032~ 52.941176~ 24.836601~ 53
## 5    100 CAJE22    50    54.2105263~ 39.473684~ 64.736842~ 30.526315~ 53
## 6    100 GODA23    51    36.9230769~ 39.487179~ 68.717948~ 31.282051~ 54
## # i 34 more variables: STDi_PRE <chr>, STSi_PRE <chr>, FC_POST <chr>,
## #   VOLOGI_POST <chr>, VOLODI_POST <chr>, VTDVGI_POST <chr>, VTSVGI_POST <chr>,
## #   FE2D_POST <chr>, STDi_POST <chr>, STSi_POST <chr>, FC_D_2 <chr>,
## #   VOLOGI_D_2 <chr>, VOLODI_D_2 <chr>, VTDVGI_D_2 <chr>, VTSVGI_D_2 <chr>,
## #   FE2D_D_2 <chr>, STDi_D_2 <chr>, STSi_D_2 <chr>, FC_D_5 <chr>,
## #   VOLOGI_D_5 <chr>, VOLODI_D_5 <chr>, VTDVGI_D_5 <chr>, VTSVGI_D_5 <chr>,
## #   FE2D_D_5 <chr>, STDi_D_5 <chr>, STSi_D_5 <chr>, FC_D_10 <chr>, ...
```

Le nombre d'individus par course

```
## # A tibble: 3 x 2
##   COURSE Nombre_Individus
##   <dbl>      <int>
## 1     40          18
## 2    100          12
## 3    160          29
```

Remplacement des valeurs temporelles et des -1 par NA

Dans cette étape, nous remplaçons les absent et case ou il ya aucune echo par des valeurs manquantes (NA).

Renommer CODE_SUJET en Individu

Transformation des données en format long

Ici, nous transformons les données de format large à format long tout en conservant certaines colonnes, comme 'CODE_SUJET' et 'COURSE'.

```
## # A tibble: 6 x 5
##   COURSE CODE_SUJET Variable Temps Valeur
##   <dbl> <chr>      <chr>      <chr> <chr>
## 1    100 Individu 1 FC          -2    58
## 2    100 Individu 1 VOLOGI     -2    27.27272727272727
## 3    100 Individu 1 VOLODI     -2    24.598930481283421
```

```
## 4    100 Individu 1 VTDVGI    -2    72.192513368983953
## 5    100 Individu 1 VTSVGI    -2    31.55080213903743
## 6    100 Individu 1 FE2D      -2     58
```

Résumés statistiques

Réalisons maintenant les résumés statistiques sur nos 8 variables d'intérêt.

```
## # A tibble: 24 x 11
##   COURSE Variable Count Missing   Min   Q1 Median   Mean   Q3   Max
##   <dbl> <chr>      <int>   <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    40 FC          90     11  40    53    61   63.6  71.5  104
## 2    40 FE2D        90     11  40    52    55   55.2  59.5   72
## 3    40 STDi        90     11  7.51 10.8   12.7  12.3  13.7  16.9
## 4    40 STSi        90     11  3.46  5.88   6.94  6.78  7.76  10.3
## 5    40 VOLODI       90     11 15.1  25.6   30.3  32.0  37.5  54.5
## 6    40 VOLOGI       90     11 25.9  33.4   37.4  37.3  41.4  55.0
## 7    40 VTDVGI       90     11 42.9  52.2   57.6  58.8  64.4  79.2
## 8    40 VTSVGI       90     11 15.0  22.8   26.3  26.6  30.3  40.1
## 9   100 FC          60      7  42    53    56    60   66   89
## 10  100 FE2D        60      8  45    51    54   55.8  58.2   75
## # i 14 more rows
## # i 1 more variable: Missing_Percentage <dbl>
```

Les valeurs manquantes

Dans cette section, nous allons compter les valeurs manquantes en prenant en compte les courses, les visites et les individus.

```
## [1] "Pourcentage de valeurs manquantes par variable :"
```

```
## # A tibble: 8 x 4
##   Variable total_values total_missing percentage_missing
##   <chr>      <int>      <int>      <dbl>
## 1 FC          295         67         22.7
## 2 FE2D        295         68         23.1
## 3 STDi        295         67         22.7
## 4 STSi        295         68         23.1
## 5 VOLODI       295         67         22.7
## 6 VOLOGI       295         67         22.7
## 7 VTDVGI       295         67         22.7
## 8 VTSVGI       295         67         22.7
```

Le nombre de données manquantes par visite et par variable cardiaque :

```
## # A tibble: 5 x 9
##   Temps    FC FE2D STDi STSi VOLODI VOLOGI VTDVGI VTSVGI
##   <chr> <int> <int> <int> <int> <int> <int> <int> <int>
## 1 -2      2    2    2    2    2    2    2    2
## 2 1      15   16   15   15   15   15   15   15
## 3 11     17   17   17   18   17   17   17   17
## 4 3      18   18   18   18   18   18   18   18
## 5 6      15   15   15   15   15   15   15   15
```

Le nombre de données manquantes par course et par variable cardiaque :

```
## # A tibble: 8 x 4
##   Variable '40' '100' '160'
##   <chr>     <int> <int> <int>
## 1 FC        11     7    49
## 2 FE2D       11     8    49
## 3 STDi       11     7    49
## 4 STSi       11     8    49
## 5 VOLODI     11     7    49
## 6 VOLOGI     11     7    49
## 7 VTDVGI     11     7    49
## 8 VTSVGI     11     7    49
```

Visualisation

Représentations graphiques

Nous allons créer des spaghetti plots pour visualiser l'évolution des variables cardiaques (comme la fréquence cardiaque et la pression artérielle) en fonction du temps, pour différentes courses (100 km, 4x40 km, 40 km, 160 km). Chaque sujet sera représenté par une courbe individuelle avec des points aux moments clés (PRE, POST, D2, D5, D10), et nous ajouterons une courbe moyenne en noir pour illustrer la tendance générale pour tous les sujets. Nous commencerons par filtrer les données selon la variable et la course, puis nous générerons les graphiques avec des courbes colorées pour chaque sujet et des points pour indiquer les valeurs spécifiques à chaque instant temporel.

Représentations visuelles des variables d'intérêts Voici les spaghetti plot pour toutes les variables d'intérêt pour la course 100km

Analyse statistique :

Test Mann-Whitney entre 40 et 100

```
## [1] "Résultats du test de Mann-Whitney (POST entre 40 km et 100 km) avec corrections BH et BY :"
```

```
## # A tibble: 8 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>     <dbl>     <dbl>     <dbl>
## 1 FC        0.0154     0.123     0.335
## 2 FE2D       0.958     0.958     1
## 3 STDi       0.902     0.958     1
## 4 STSi       0.786     0.958     1
## 5 VOLODI     0.416     0.831     1
## 6 VOLOGI     0.312     0.831     1
## 7 VTDVGI     0.0575     0.230     0.625
## 8 VTSVGI     0.711     0.958     1
```

Test Mann-Whitney entre 40 et 160

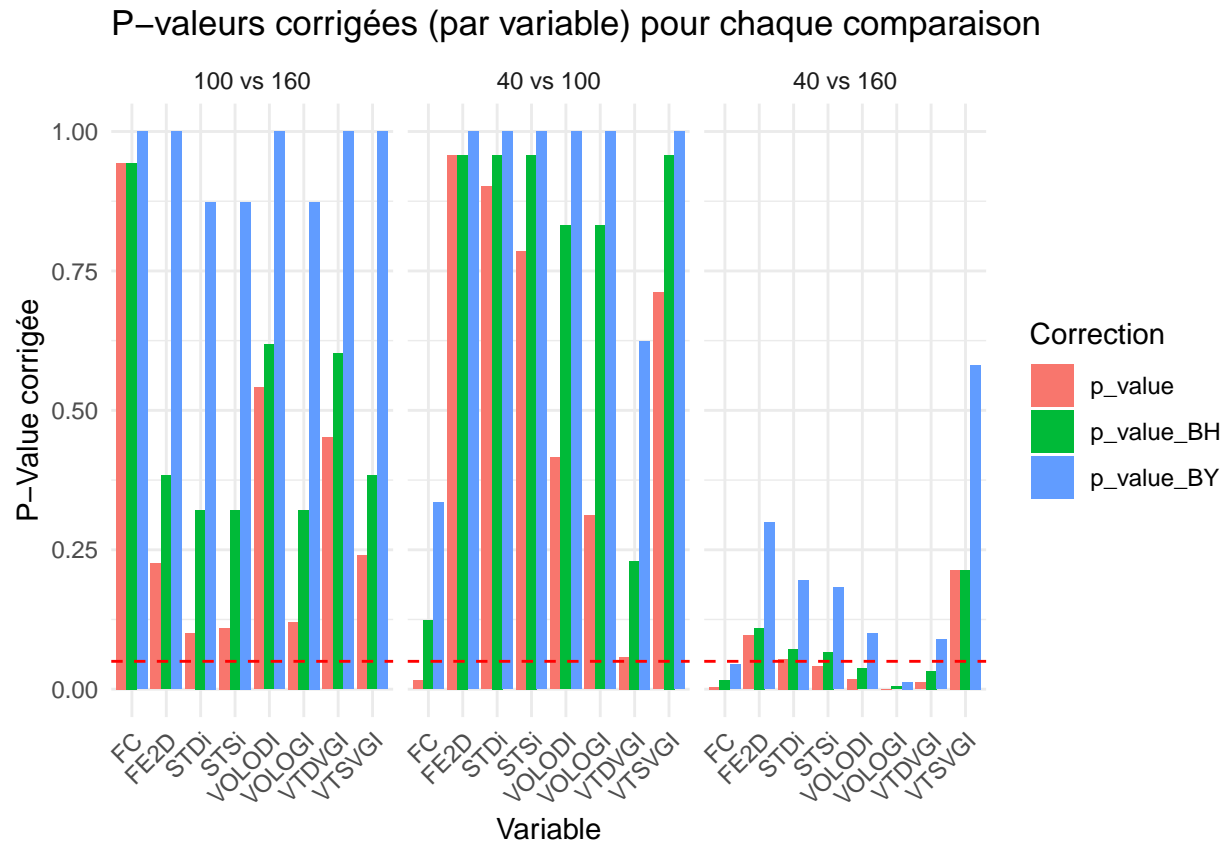
```
## [1] "Résultats du test de Mann-Whitney (POST entre 40 km et 160 km) avec corrections BH et BY :"
```

```
## # A tibble: 8 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>      <dbl>      <dbl>      <dbl>
## 1 FC        0.00409      0.0164      0.0445
## 2 FE2D      0.0961      0.110      0.298
## 3 STDi      0.0540      0.0719      0.196
## 4 STSi      0.0418      0.0669      0.182
## 5 VOLODI    0.0183      0.0366      0.0994
## 6 VOLOGI    0.000582     0.00465     0.0126
## 7 VTDVGI    0.0123      0.0328      0.0891
## 8 VTSVGI    0.214       0.214      0.582
```

Test Mann-Whitney entre 100 et 160

```
## [1] "Résultats du test de Mann-Whitney (POST entre 100 km et 160 km) avec corrections BH et BY :"
```

```
## # A tibble: 8 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>      <dbl>      <dbl>      <dbl>
## 1 FC        0.944      0.944      1
## 2 FE2D      0.225      0.383      1
## 3 STDi      0.0997     0.322     0.874
## 4 STSi      0.110      0.322     0.874
## 5 VOLODI    0.541      0.618      1
## 6 VOLOGI    0.121      0.322     0.874
## 7 VTDVGI    0.452      0.602      1
## 8 VTSVGI    0.240      0.383      1
```



Test de Wilcoxon pour les Comparaisons Intra-Course (PRE vs POST)

```
## [1] "Résultats du test de Wilcoxon (PRE vs POST) pour 40 km :"
```

```
## # A tibble: 8 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>      <dbl>      <dbl>      <dbl>
## 1 FC        0.000476    0.00381    0.0104
## 2 FE2D      0.864      0.864      1
## 3 STDi      0.776      0.864      1
## 4 STSi      0.0382      0.153     0.415
## 5 VOLODI    0.755      0.864      1
## 6 VOLOGI    0.453      0.864      1
## 7 VTDVGI    0.132      0.353     0.959
## 8 VTSVGI    0.755      0.864      1
```

```
## [1] "Résultats du test de Wilcoxon (PRE vs POST) pour 100 km :"
```

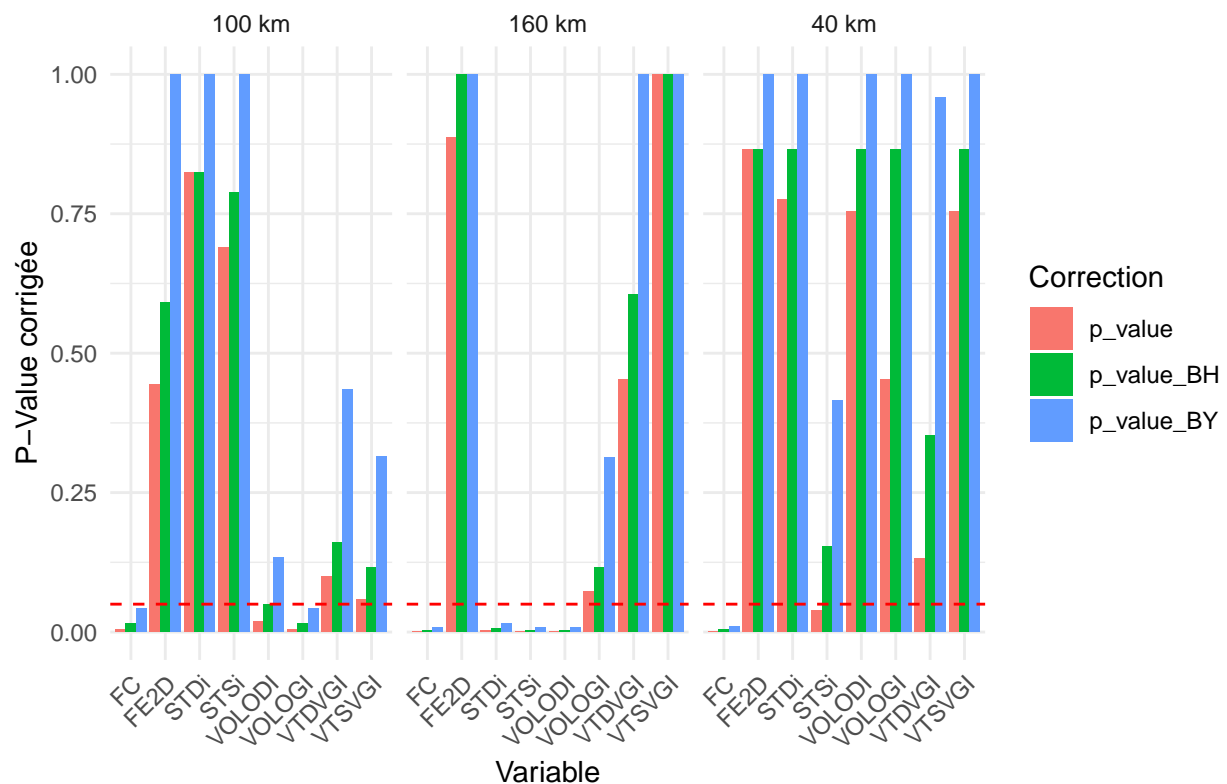
```
## # A tibble: 3 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>      <dbl>      <dbl>      <dbl>
## 1 FC        0.00384    0.0154    0.0419
## 2 FE2D      0.444      0.592      1
## 3 STDi      0.824      0.824      1
```

```
## 4 STSi      0.689      0.788      1
## 5 VOLODI    0.0185     0.0492     0.134
## 6 VOLOGI    0.00386    0.0154     0.0419
## 7 VTDVGI    0.100      0.160      0.435
## 8 VTSVGI    0.0580     0.116      0.315
```

```
## [1] "Résultats du test de Wilcoxon (PRE vs POST) pour 160 km :"
```

```
## # A tibble: 8 x 4
##   Variable p_value p_value_BH p_value_BY
##   <chr>      <dbl>      <dbl>      <dbl>
## 1 FC        0.000415    0.00291    0.00790
## 2 FE2D      0.887        1          1
## 3 STDi      0.00286    0.00572    0.0156
## 4 STSi      0.000921    0.00291    0.00790
## 5 VOLODI    0.00109    0.00291    0.00790
## 6 VOLOGI    0.0720     0.115      0.313
## 7 VTDVGI    0.453      0.605      1
## 8 VTSVGI    1          1          1
```

P-valeurs corrigées (PRE vs POST) par course et variable



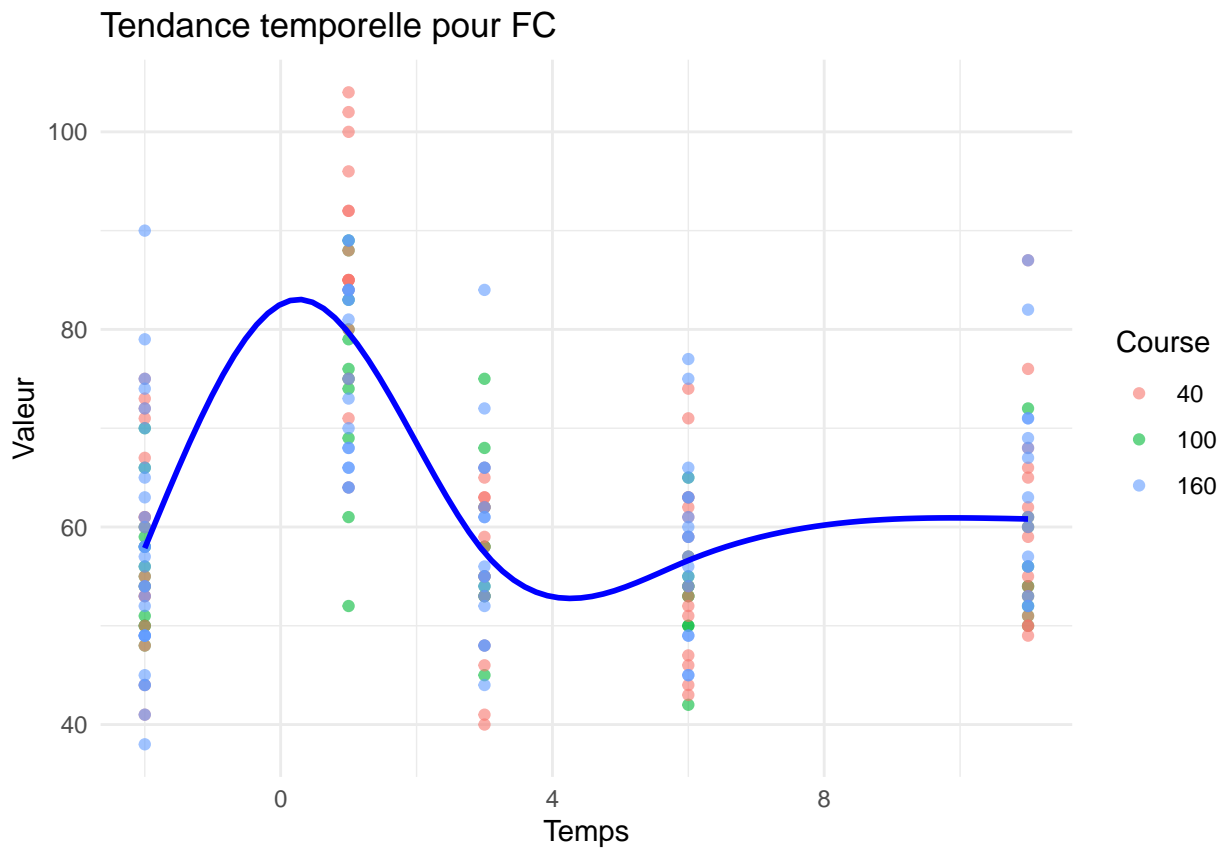
Modèle mixte avec splines

On va ajuster un modèle mixte pour analyser l'évolution temporelle des paramètres physiologiques des ultra-traileurs. Ce modèle inclut :

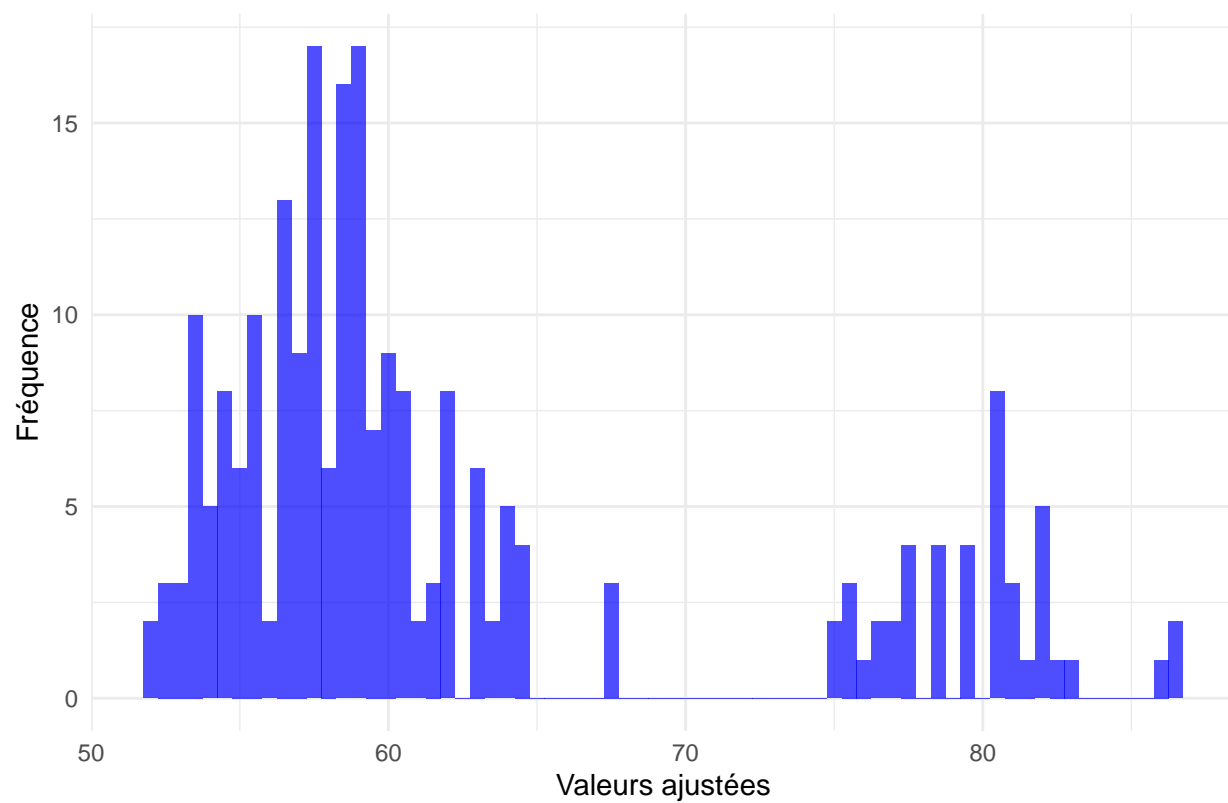
- **Des splines naturelles** pour capturer les variations non linéaires dans le temps.
- **Un effet fixe** pour comparer les différentes courses.
- **Un effet aléatoire** pour tenir compte des spécificités individuelles de chaque participant.

L'objectif est de mieux comprendre comment ces paramètres évoluent en fonction du temps et de la distance parcourue, tout en intégrant la variabilité intra-individuelle.

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Valeur ~ ns(Temps, df = df_spline) + COURSE + (1 | CODE_SUJET)
## Data: data_filtered
##
## REML criterion at convergence: 1672.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.5229 -0.6528 -0.1098  0.5905  3.5627
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## CODE_SUJET (Intercept) 15.01     3.875
## Residual              86.90     9.322
## Number of obs: 228, groups: CODE_SUJET, 29
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      5.783e+01  2.136e+00  27.076
## ns(Temps, df = df_spline)1 -1.280e+01  3.123e+00  -4.099
## ns(Temps, df = df_spline)2 -9.641e+00  2.749e+00  -3.506
## ns(Temps, df = df_spline)3  3.717e+01  3.995e+00   9.302
## ns(Temps, df = df_spline)4 -1.807e+01  2.147e+00  -8.415
## COURSE              -9.307e-04  1.312e-02  -0.071
##
## Correlation of Fixed Effects:
##              (Intr) n(T,d=d_)1 n(T,d=d_)2 n(T,d=d_)3 n(T,d=d_)4
## ns(T,d=d_)1 -0.264
## ns(T,d=d_)2 -0.193 -0.262
## ns(T,d=d_)3 -0.419 -0.289  0.444
## ns(T,d=d_)4 -0.161  0.531 -0.018 -0.218
## COURSE      -0.736  0.063  0.035  0.091  0.039
##
##              Estimate Std. Error t value
## (Intercept)      5.782733e+01  2.13575455  27.07583343
## ns(Temps, df = df_spline)1 -1.279997e+01  3.12287733  -4.09877532
## ns(Temps, df = df_spline)2 -9.640636e+00  2.74938062  -3.50647574
## ns(Temps, df = df_spline)3  3.716545e+01  3.99540404   9.30205018
## ns(Temps, df = df_spline)4 -1.806963e+01  2.14731599  -8.41498360
## COURSE              -9.307093e-04  0.01311704  -0.07095421
```

Histogramme des valeurs ajustées



Graphique des résidus

