



**AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE**

# **Pakiet R Biblioteka cycleRtools**

**Pakiet funkcji do analizy danych rowerowych**

**Wydział Geodezji Górniczej i Inżynierii Środowiska  
Katedra Ochrony Terenów Górniczych, Geoinformatyki i Geodezji Górniczej**

**Kraków, 21 grudnia 2015 roku**

**mgr inż. Dawid Mrocheń**

**Autor: Jordan Mackie**

**Licencja: MIT (X11) + pliki**

**Publikacja: 02.12.2015**

**Dodatkowe biblioteki: raster, leaflet, changepoint**

**<https://github.com/jmackie4/cycleRtools>**

## cycling\_data

```
> colnames(cycling_data)
```

```
[1] "timer.s"           "timer.min"       "timestamp"  
[4] "delta.t"          "lat"             "lng"  
[7] "distance.km"      "speed.kmh"      "elevation.m"  
[10] "delta.elev"       "VAM"            "power.W"  
[13] "power.smooth.W"  "work.J"         "Wexp.kJ"  
[16] "cadence.rpm"     "lap"            ".elevation.corrected.m"
```

```
4019    4044                67.4                2015-07-25 08:47:49  
      1                51.9669936597347  -0.617225412279367  
      34.66229           29.0844             147.2  
      -0.1999999999999932  0                  140  
      164.307052759117    857797            6.02827799994205  
      85                  "1,,              146.302014004964
```

summary(data)

```
> data(cycling_data)
> summary(cycling_data)
$date_time
  ride.date ride.start.time
"25 lip 2015"      "07:40:25"

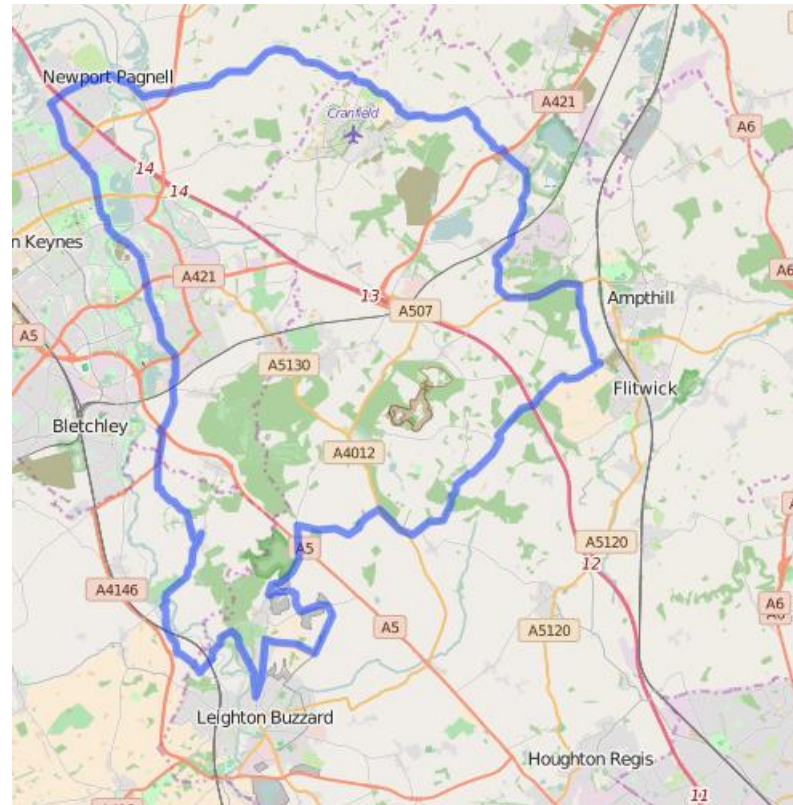
$overall
  ride.time.min elapsed.time.min distance.km work.kJ norm.work.kJ
  145.16667    147.96667      71.39062    1741.01800    2260.61503
  climbing.m   srPE.score   TRIMP.score speed.mean.kmh speed.sd.kmh
  1025.20000      NA          NA          29.06536      7.73180
  power.mean.W  power.sd.W   xPower.W   power.max.W
  199.76444    117.29323   259.54248   809.00000
```

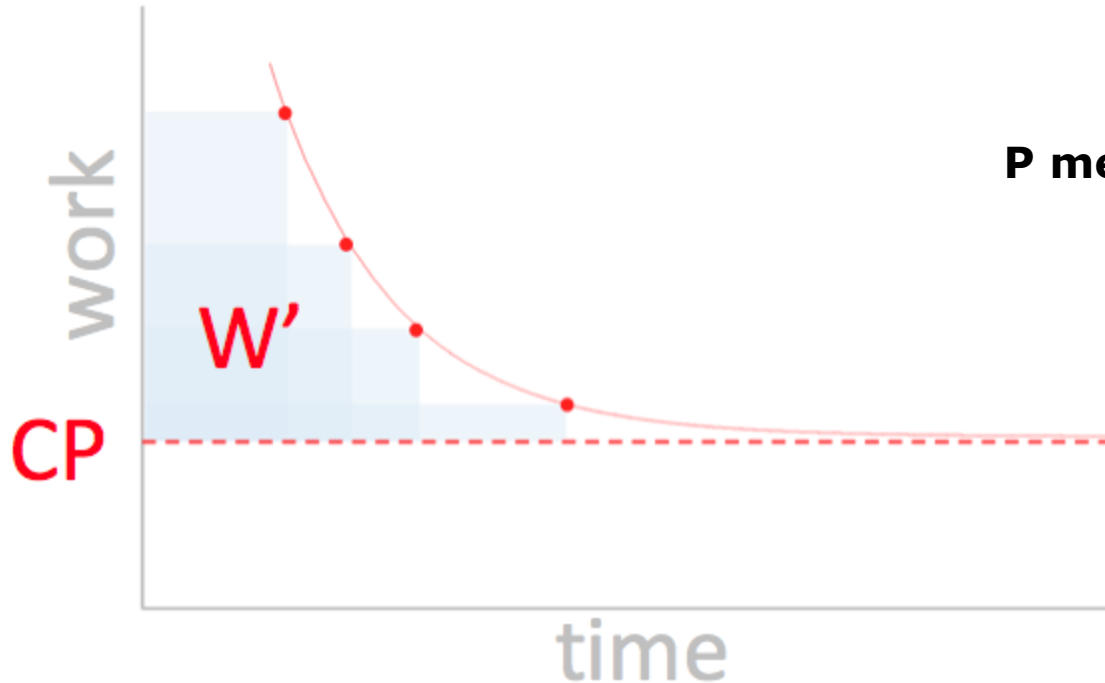
# Lokalizacja

biblioteka **leaflet**

```
leaflet(data=NULL, width = NULL, height = NULL, padding = 0)
```

```
> leaflet(cycling_data) %>% addTiles() %>% addPolylines(~lng, ~lat)  
> |
```





**P mean = 199.76 W**  
**xP = 259.54 W**  
**NP = 254.56 W**

źródło: <http://markliversedge.blogspot.com>

```
> NP(cycling_data)
Creating uniform data using timer.s , delta value = 1
Smoothing power.W using timer.s (30)...
[1] 254.5609
```

```
Wbal(data, time = "timer.s", pwr = "power.W", CP, .string = FALSE)
```

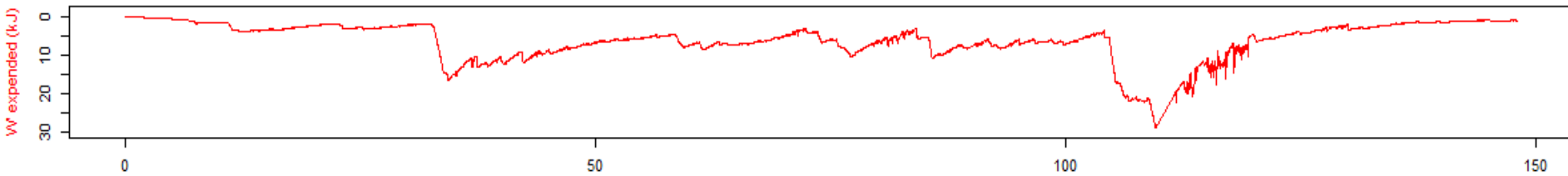
```
cycling_data$Wexp.kJ <- Wbal(data = cycling_data,  
                             time = timer.s,  
                             pwr = power.W,  
                             CP = 330)
```

```
Wbal_plots(data, x = 1, n = c(1, 2, 3), xlab = NULL, xlim = NULL,  
CP = NULL, laps = FALSE, ...)
```

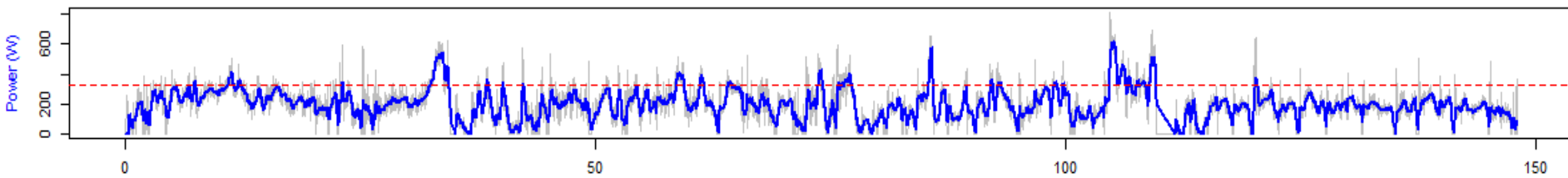
```
Wbal_plots(data = cycling_data,  
           x = 2,  
           CP = 330)
```

# Wbal

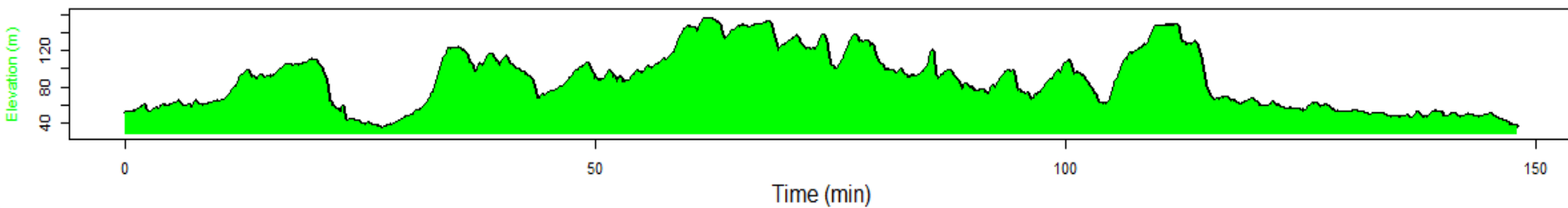
Max W' Expended (kJ): 28.58



xPower: 259.54

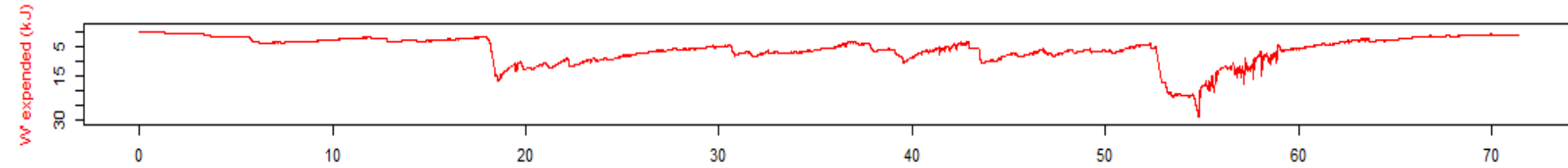


Climbing: 1025.2 m

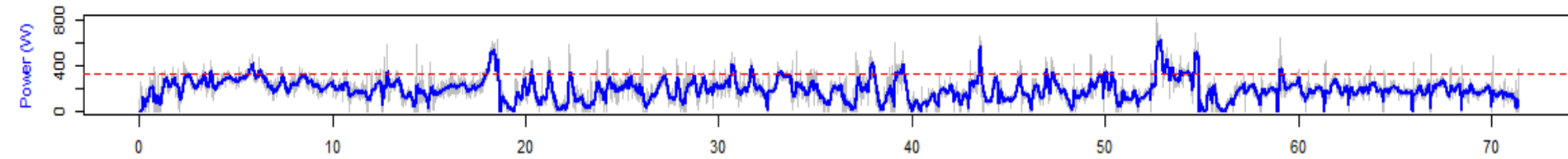




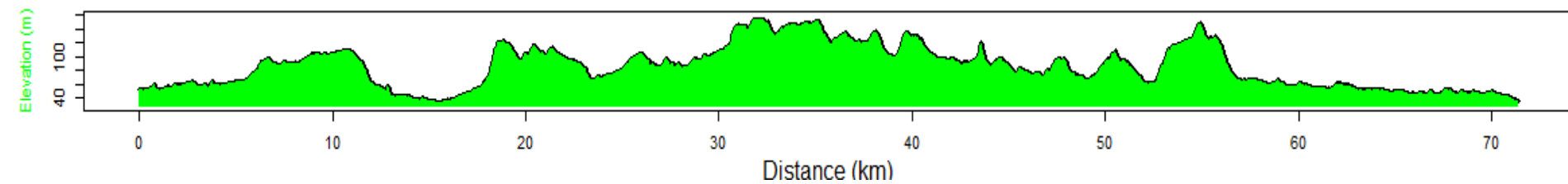
Max W Expended (kJ): 28.58



xPower: 259.54



Climbing: 1025.2 m



# Correct Elevation Data

biblioteka *raster*

```
elevation_correct(data, country)
```

```
  getData(„ISO3“)
```

```
[118,] "KEN" "Kenya"
```

```
[119,] "KIR" "Kiribati"
```

```
[120,] "XKO" "Kosovo"
```

```
[121,] "KWT" "Kuwait"
```

```
[122,] "KGZ" "Kyrgyzstan"
```

```
[123,] "LAO" "Laos"
```

```
[124,] "LVA" "Latvia"
```

```
[125,] "LBN" "Lebanon"
```

```
[126,] "LSO" "Lesotho"
```

```
[127,] "LBR" "Liberia"
```

```
[128,] "LBY" "Libya"
```

```
[129,] "LIE" "Liechtenstein"
```

```
[130,] "LTU" "Lithuania"
```

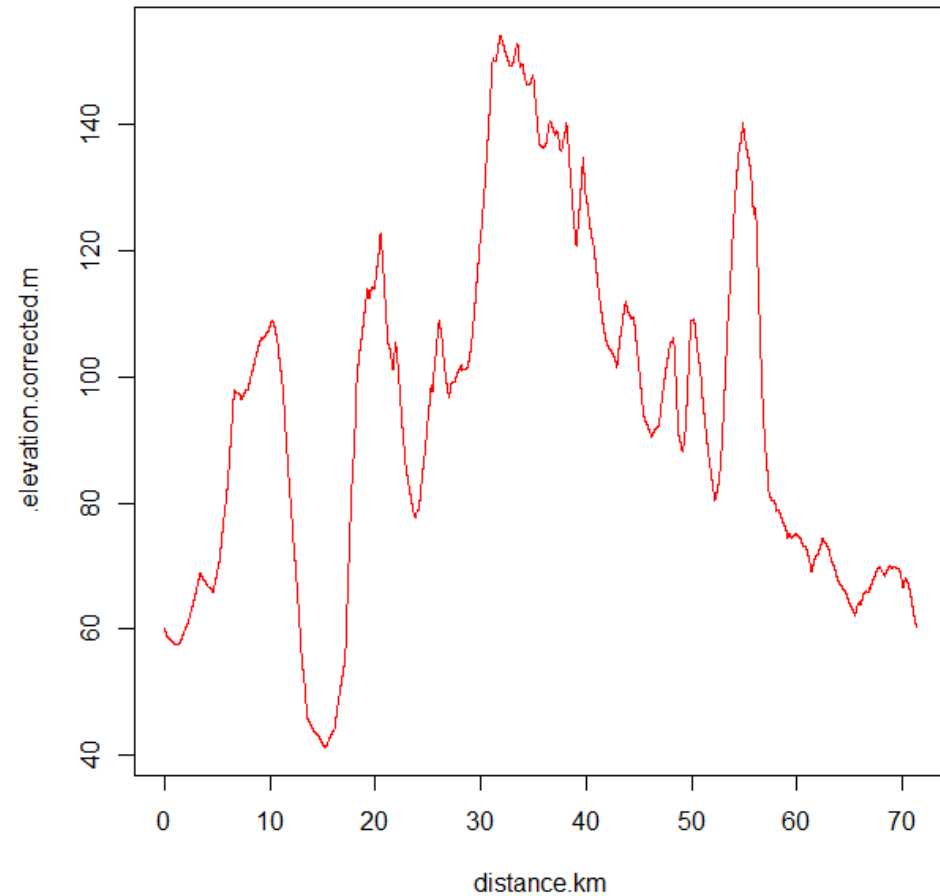
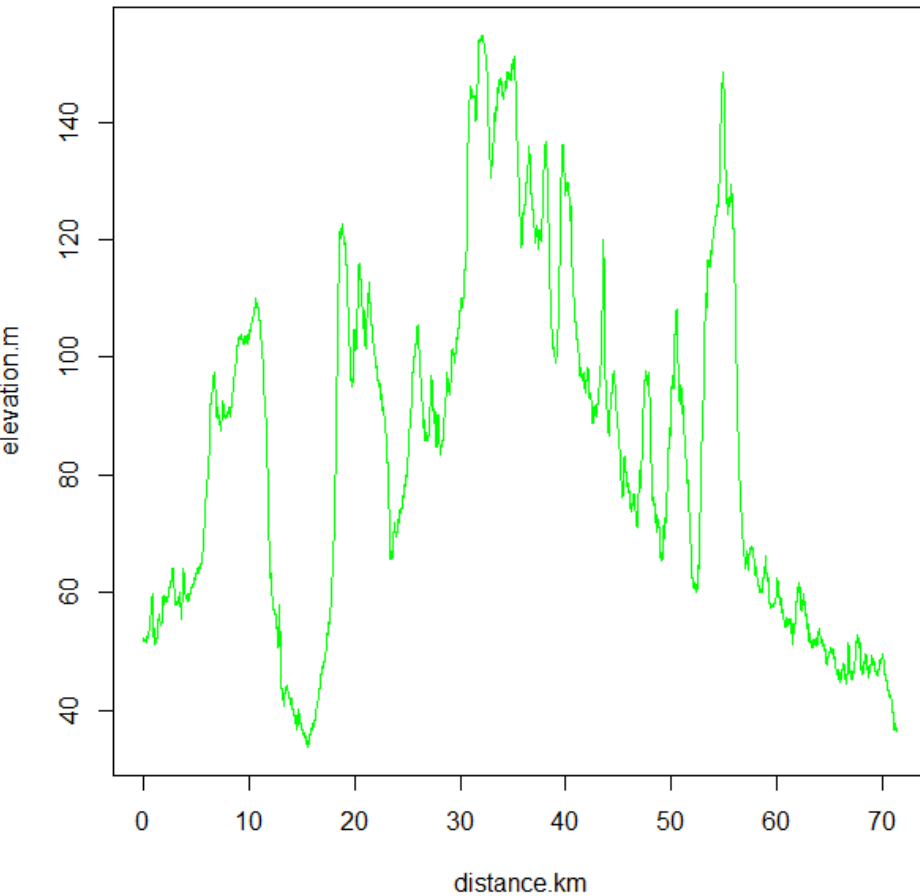
```
cycling_data$.elevation.corrected.m <-
```

```
  elevation_correct(cycling_data, country = „GBR“)
```

## Correct Elevation Data

```
par(mfrow = c(1, 2), mar = c(4.1, 4.1, 1.1, 1.1))  
with(cycling_data, plot(x = distance.km,  
                        y = elevation.m,  
                        type = "l",  
                        col = "green" ))  
with(cycling_data, plot(x = distance.km,  
                        y = .elevation.corrected.m,  
                        type = "l",  
                        col = "red" ))
```

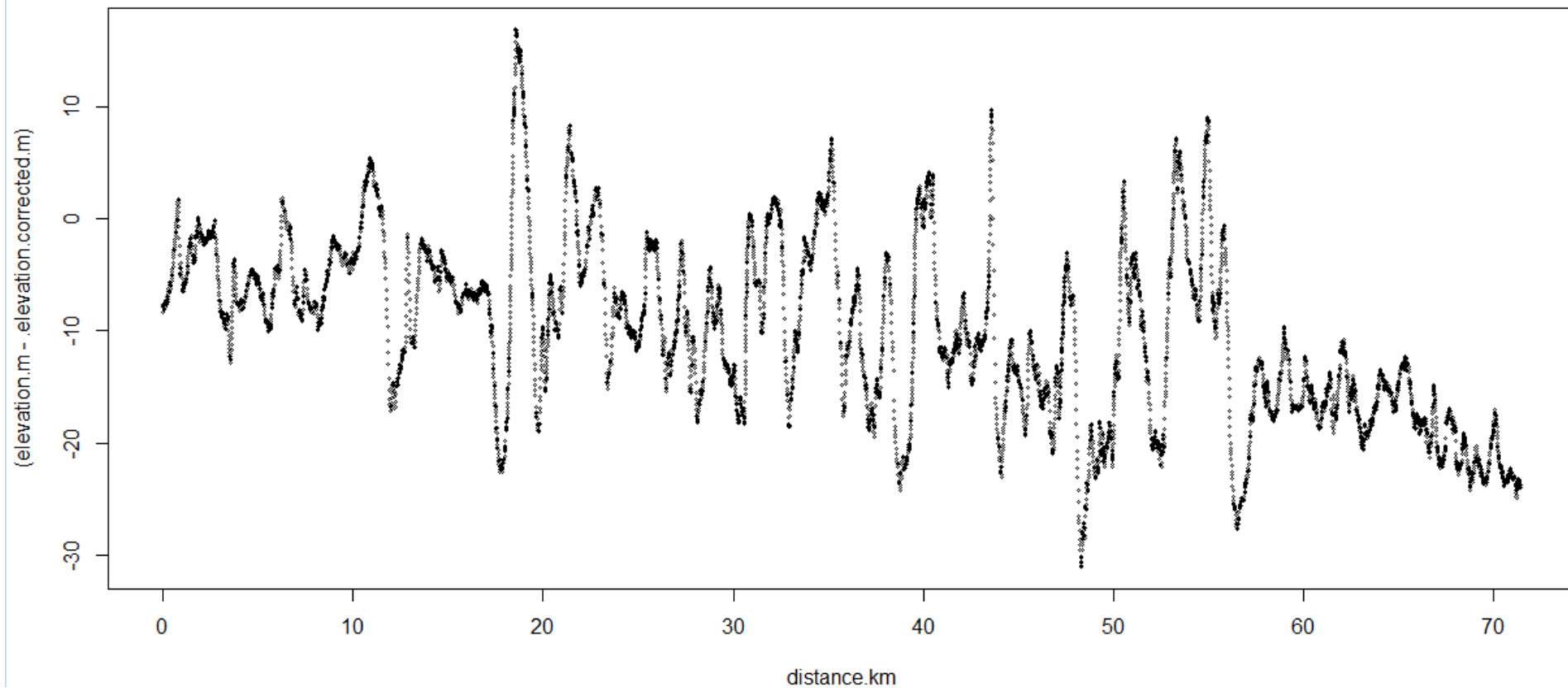
# Correct Elevation Data



## Correct Elevation Data

```
par(mar = c(4.1, 4.1, 1.1, 1.1))  
with(cycling_data, plot(x = distance.km,  
                        y = (elevation.m - .elevation.corrected.m),  
                        cex = 0.4))
```

# Correct Elevation Data



## mmv – maximal mean values

```
tsec <- c(1, 5, 20) * 60
```

```
mmv(data, column, windows, delta = NULL, verbose = NULL,  
     .uniform = FALSE, .string = FALSE)
```

```
> mmv(data      = cycling_data,  
+      windows  = tsec,  
+      column   = power.W,  
+      verbose  = FALSE)  
  
              60      300     1200  
Best mean value 522.47 390.06 245.19  
Recorded @      6275.00 6271.00 145.00  
> |
```

## zone\_time

zone\_time(data, column, zbounds, pct, .sting)

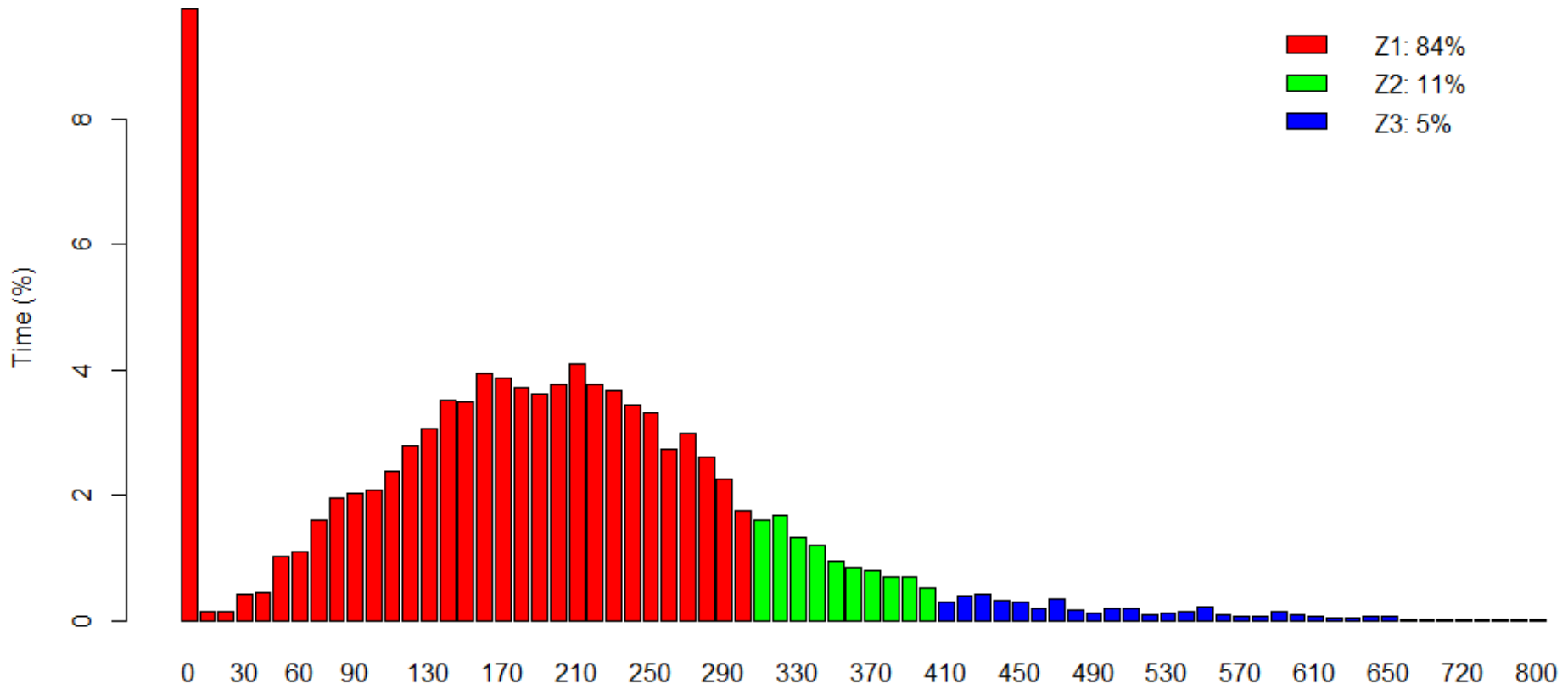
```
> zone_time(data = cycling_data,  
+ column = power.W,  
+ zbounds = c(300, 400),  
+ pct = TRUE)  
Zone 1 Zone 2 Zone 3  
      84      11      5  
> |
```

zdist\_plot(data, column, bindwith, zbounds, .string)

```
> zdist_plot(data = cycling_data,  
+ column = power.W,  
+ zbounds = c(300, 400))  
> |
```



# zone\_time



# diff\_section

podział na sekcje

```

> timer.s=c(1:60, 120:180)
> diff_section(timer.s)
 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
 [71] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [106] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

```

ilość sekcji

```

> cycling_data$stop_sections = diff_section(cycling_data$timer.s)
> unique(cycling_data$stop_sections)
 [1] 1 2 3 4 5

```

czas jazdy

```

> ride_time = (with(cycling_data, tapply(timer.s, stop_sections, max)) / 60) -
+ (with(cycling_data, tapply(timer.s, stop_sections, min)) / 60)
> ride_time
     1          2          3          4          5
34.68333 74.58333  8.10000 14.96667 12.95000

```

moc średnia na odcinkach

```

> with(cycling_data, tapply(power.W, stop_sections, mean))
     1          2          3          4          5
233.0274 199.5548 130.0452 188.3177 168.5938
> |

```

biblioteka *changepoint*

```
> colnames(interval_data)
```

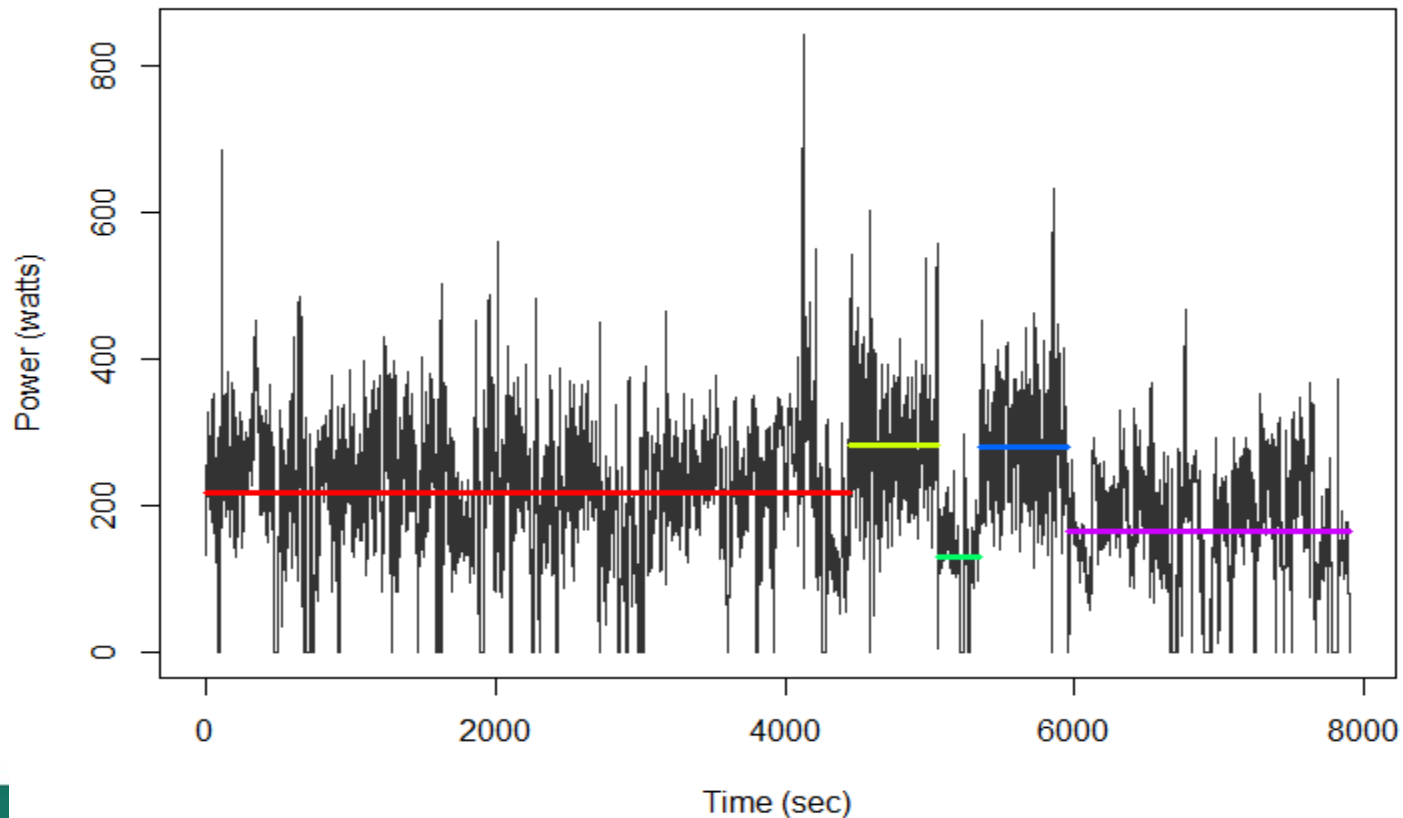
```
„timer.s”      „timer.min”      „timestamp”      „delta.t”
„lat”          „lng”            „distance.km”    „speed.kmh”
„elevation.m”  „delta.elev”    „VAM”            „power.W”
„power.smooth.W”  „work.J”      „Wexp.kJ”        „lap”
„interval”
```

```
2983      3016      50.26666666666667      2015-09-23 09:08:54      1
52.8922425303608      -0.961891785264015      26.0483      31.5936
34.8      -1      0      217
180.179870921586      217      2.54095815721202      "1,"
1
```

## interval\_detect

interval\_detect(data, sections, plot)

```
> interval_data$interval <- interval_detect(interval_data, 5, FALSE)
> tapply(interval_data$delta.t, interval_data$interval, sum) / 60
      1      2      3      4      5
74.150000 10.100000  4.900000  9.966667 32.650000
> interval_detect(interval_data, 5, TRUE)
```





Dziękuję za uwagę