MATH2349 Semester 2, 2018

Code

Assignment 3

Phalgun Haribabu Chintal, s3702107 and Syed Junaid Ahmed, s3731300

Required packages

All the packages required to satsify tasks are installed.

Hide

```
library(dplyr)
library(readr)
library(Hmisc)
library(outliers)
library(tidyr)
library(knitr)
library(magrittr)
library(forecast)
```

Executive Summary

The data preprocessing plays an essential role because the data is made ready before the start of the analysis. With a specific end goal to discover from the knowledge gained in this course, the datasets are gathered through www.kaggle.com, which has a csv extension, contains data regarding the powerlifting. Firstly, two datasets are imported into rstudio through base r function. Secondly, these datasets are merged from inner_join by 'MeetID'. Furthermore, types of variables, attributes, dimensions, and the required type conversion are processed. The dataset has been reshaped because it violates the tidy format. Moreover, the new column MHR is mutated that holds for Maximal Heart Rate. The missing values and inconsistencies of the merged dataset are checked if any they are replaced by mean and mode. Possible outliers are inspected and handled them by the capping method. At last, BodyweightKg variable is transformed to normal distribution from left skewed.

Data

The datasets are two comma-separated values file namely, meets.csv and openpowerlifting.csv. These data are from www.kaggle.com. This dataset is a depiction about an association called OpenPowerlifting which keep tracks of all information of meets and contender results. Contenders achieve to lift the maximum weight for their position in three different weightlifting classifications.

1. Meets: meets.csv is a file of information about all the competitors incorporated into the OpenPowerlifting database.

- MeetID: Identification Number
- MeetPath: represents the direction
- Federation: shows the group
- Date: represents the date
- Meet Country: shows the country name
- MeetState: displays the name of the state
- MeetTown: represents town name
- MeetName: shows the name of the meet that are held

2. openpowerlifting: openpowerlifting.csv is a file of information about all the competitors who attended those meets and the details and lifts that they posted at them.

- MeetID: Identification Number
- Name: Name of the competitors
- Sex: gender of the competitors
- Equipment: shows the equipments
- Age: determines the age of the competitors
- Division: shows which category that competitors belong to
- BodyweightKg: It represents competitors weight in kg
- WeightclassKg: determines the weight category that competitors can take part
- Squat4Kg: it is the first lift performed at every single powerlifting meet
- BestSquat4Kg: the time performed in the squat by competitors
- Bench4Kg: the competitiors lay down on the bench and lifts the bar
- BestBenchKg: the time performed in the bench by competitors
- Deadlift4Kg: the competitors lifts the bar off the ground to the level of the hips, then lowered to the ground
- BestDeadliftKg: the time performed in the deadlift by competitors
- TotalKg: shows the total kg lifted by the competitors
- Place: shows the result where the competitors stand after their lift
- Wilks: it is the formula used to measure the strength of the powerlifter against other powerlifters

The datasets have been obtained from the following source:

https://www.kaggle.com/open-powerlifting/powerlifting-database (https://www.kaggle.com/open-powerlifting/powerlifting-database)

The datasets, meets.csv, and openpowerlifting.csv are merged through inner_join by common attribute (MeetID) and named the new dataset as merge.

Hide

```
meets <- read.csv("meets.csv")
openpowerlifting <- read.csv("openpowerlifting.csv")
merge <- inner_join(meets, openpowerlifting)
```

```
Joining, by = "MeetID"
```

Hide

```
head(merge)
```

	MeetID <int>	MeetPath <fctr>	Federation <fctr>	Date <fctr>	MeetCountry <fctr>	MeetState <fctr>	MeetTown <fctr>
1	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte
2	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte
3	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte
4	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte
5	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte
6	0	365strong/1601	365Strong	2016-10-29	USA	NC	Charlotte

6 rows | 1-8 of 24 columns

Understand

Hide

```
merge$Sex <- as.character(merge$Sex)
sapply(merge, typeof)
```

```
       MeetID       MeetPath     Federation           Date    MeetCountry      MeetState
   MeetTown
     "integer"      "integer"      "integer"      "integer"      "integer"      "integer"
  "integer"
      MeetName           Name            Sex      Equipment            Age       Division   B
odyweightKg
     "integer"      "integer"    "character"      "integer"       "double"      "integer"
   "double"
 WeightClassKg       Squat4Kg    BestSquatKg       Bench4Kg    BestBenchKg    Deadlift4Kg Bes
tDeadliftKg
     "integer"       "double"       "double"       "double"       "double"       "double"
   "double"
       TotalKg          Place          Wilks
      "double"      "integer"      "double"
```

Hide

```
str(merge)
```

```
'data.frame':   386414 obs. of  24 variables:
 $ MeetID        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ MeetPath      : Factor w/ 8482 levels "365strong/1601",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Federation    : Factor w/ 60 levels "365Strong","AAPF",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ Date          : Factor w/ 2652 levels "1974-03-02","1974-03-30",..: 2421 2421 2421 2421 24
21 2421 2421 2421 2421 2421 ...
 $ MeetCountry   : Factor w/ 45 levels "Argentina","Australia",..: 44 44 44 44 44 44 44 44 44
44 ...
 $ MeetState     : Factor w/ 81 levels "","AB","ACT",..: 39 39 39 39 39 39 39 39 39 39 ...
 $ MeetTown      : Factor w/ 1540 levels "","Ã\230. Ã□rdal",..: 249 249 249 249 249 249 249
249 249 249 ...
 $ MeetName      : Factor w/ 5217 levels "015 Pennsylvania State Bench Press and Deadlif
t",..: 719 719 719 719 719 719 719 719 719 719 ...
 $ Name          : Factor w/ 136687 levels "A'daireon Madlock",..: 9239 35550 35550 35550 371
27 29916 91588 91588 106278 106278 ...
 $ Sex           : chr  "F" "F" "F" "F" ...
 $ Equipment     : Factor w/ 5 levels "Multi-ply","Raw",..: 5 3 3 2 2 5 2 2 5 2 ...
 $ Age           : num  47 42 42 42 18 28 60 60 52 52 ...
 $ Division      : Factor w/ 4247 levels "","-100kg","11-12R",..: 3176 3175 3288 3288 4000 32
88 3179 3288 67 3812 ...
 $ BodyweightKg  : num  59.6 58.5 58.5 58.5 63.7 ...
 $ WeightClassKg : Factor w/ 52 levels "","100","100+",..: 31 31 31 31 35 35 35 35 35 35 ...
 $ Squat4Kg      : num  NA NA NA NA NA ...
 $ BestSquatKg   : num  47.6 142.9 142.9 NA NA ...
 $ Bench4Kg      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ BestBenchKg   : num  20.4 95.2 95.2 95.2 31.8 ...
 $ Deadlift4Kg   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ BestDeadliftKg: num  70.3 163.3 163.3 NA 90.7 ...
 $ TotalKg       : num  138.3 401.4 401.4 95.2 122.5 ...
 $ Place         : Factor w/ 82 levels "","1","10","11",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ Wilks         : num  155 456 456 108 130 ...
```

Hide

```
attributes(merge)
```

```
$`row.names`
   [1]    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
18   19   20
  [21]   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37
38   39   40
  [41]   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56   57
58   59   60
  [61]   61   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76   77
78   79   80
  [81]   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95   96   97
98   99  100
 [101]  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117
118  119  120
 [121]  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137
138  139  140
 [141]  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157
158  159  160
 [161]  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177
178  179  180
 [181]  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197
198  199  200
 [201]  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217
218  219  220
 [221]  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237
238  239  240
 [241]  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257
258  259  260
 [261]  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277
278  279  280
 [281]  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297
298  299  300
 [301]  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317
318  319  320
 [321]  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337
338  339  340
 [341]  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357
358  359  360
 [361]  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377
378  379  380
 [381]  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397
398  399  400
 [401]  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417
418  419  420
 [421]  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437
438  439  440
 [441]  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457
458  459  460
 [461]  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477
478  479  480
 [481]  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497
498  499  500
 [501]  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517
518  519  520
 [521]  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537
538  539  540
 [541]  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557
558  559  560
```

```
 [561]  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577
578  579  580
 [581]  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597
598  599  600
 [601]  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617
618  619  620
 [621]  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637
638  639  640
 [641]  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657
658  659  660
 [661]  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677
678  679  680
 [681]  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697
698  699  700
 [701]  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717
718  719  720
 [721]  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737
738  739  740
 [741]  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  756  757
758  759  760
 [761]  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777
778  779  780
 [781]  781  782  783  784  785  786  787  788  789  790  791  792  793  794  795  796  797
798  799  800
 [801]  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817
818  819  820
 [821]  821  822  823  824  825  826  827  828  829  830  831  832  833  834  835  836  837
838  839  840
 [841]  841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857
858  859  860
 [861]  861  862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877
878  879  880
 [881]  881  882  883  884  885  886  887  888  889  890  891  892  893  894  895  896  897
898  899  900
 [901]  901  902  903  904  905  906  907  908  909  910  911  912  913  914  915  916  917
918  919  920
 [921]  921  922  923  924  925  926  927  928  929  930  931  932  933  934  935  936  937
938  939  940
 [941]  941  942  943  944  945  946  947  948  949  950  951  952  953  954  955  956  957
958  959  960
 [961]  961  962  963  964  965  966  967  968  969  970  971  972  973  974  975  976  977
978  979  980
 [981]  981  982  983  984  985  986  987  988  989  990  991  992  993  994  995  996  997
998  999 1000
 [ reached getOption("max.print") -- omitted 385414 entries ]

$names
 [1] "MeetID"         "MeetPath"       "Federation"     "Date"           "MeetCountry"    "Me
etState"
 [7] "MeetTown"       "MeetName"       "Name"           "Sex"            "Equipment"      "Ag
e"
[13] "Division"       "BodyweightKg"   "WeightClassKg"  "Squat4Kg"       "BestSquatKg"    "Be
nch4Kg"
[19] "BestBenchKg"    "Deadlift4Kg"    "BestDeadliftKg" "TotalKg"        "Place"          "Wi
lks"

$class
[1] "data.frame"
```

Hide

```
dim(merge)
```

```
[1] 386414     24
```

Hide

```
merge$Equipment <- factor(merge$Equipment, levels = c("Straps", "Single-ply", "Multi-ply", "Raw", "Wraps"),
                          labels = c("Straps", "Single-ply", "Multi-ply", "Raw", "Wraps"), ordered = TRUE)
levels(merge$Equipment)
```

```
[1] "Straps"     "Single-ply" "Multi-ply"  "Raw"        "Wraps"
```

- For data type conversion, an as.character function is used to convert from factor to character. So, the sex variable is converted into character.
- When typeof is used in the merge, it returns the type of all variables.
- str() is used to display the structure of merge dataset.
- attributes() is used to display the attributes of merge dataset.
- dim() is used to obtain the lengths of a merge. So, it retrieves the dimension as 386414 and 24.
- Equipment variable is factored, levels and it's labels are ordered according to its dimensions.

Tidy & Manipulate Data I

This dataset is in an untidy format as it contains two values in its own cell. So, separate() is used to overcome this problem inorder to look tidy.

Hide

```
merge <- merge %>% separate(MeetPath, into = c("Path", "Number"), sep = "/")
head(merge)
```

	Mee… <int>	Path <chr>	Nu… <chr>	Federation <fctr>	Date <fctr>	MeetCountry <fctr>	MeetState <fctr>	MeetTown <fctr>
1	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
2	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
3	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
4	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
5	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
6	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte

6 rows | 1-9 of 25 columns

- MeetPath variable is now separated into two variables, Path and Number.

- This dataset now satisfies the tidy data principle as it contains the following information:
    1. Each variable has its own column.
    2. Each observation has its own row.
    3. Each value has its own cell.

Tidy & Manipulate Data II

Hide

```
merge <- mutate(merge, MHR = 220 - Age)
head(merge)
```

	Mee… <int>	Path <chr>	Nu… <chr>	Federation <fctr>	Date <fctr>	MeetCountry <fctr>	MeetState <fctr>	MeetTown <fctr>
1	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
2	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
3	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
4	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
5	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte
6	0	365strong	1601	365Strong	2016-10-29	USA	NC	Charlotte

6 rows | 1-9 of 26 columns

- The new variable MHR is created from the existing variable through mutate().
- MHR stands for Maximal Heart Rate, that shows the upper limit of what the cardiovascular system can handle during physcial activity when subtracted 220 with age.

Scan I

Hide

```
colSums(is.na(merge))
```

```
       MeetID          Path        Number    Federation          Date   MeetCountry
  MeetState
            0             0             0             0             0             0
          0
     MeetTown      MeetName          Name           Sex     Equipment           Age
   Division
            0             0             0             0             0        239267
          0
  BodyweightKg  WeightClassKg      Squat4Kg   BestSquatKg      Bench4Kg   BestBenchKg
Deadlift4Kg
          2402             0        385171         88343        384452         30050
     383614
BestDeadliftKg       TotalKg         Place         Wilks           MHR
         68567         23177             0         24220        239267
```

Hide

```
sum(is.nan(merge$MeetID))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Path))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Number))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Federation))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Date))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$MeetCountry))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$MeetState))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$MeetTown))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$MeetName))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Name))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Sex))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Equipment))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Age))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Division))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$BodyweightKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$WeightClassKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Squat4Kg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$BestSquatKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Bench4Kg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$BestBenchKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Deadlift4Kg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$BestDeadliftKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$TotalKg))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Place))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$Wilks))
```

```
[1] 0
```

Hide

```
sum(is.nan(merge$MHR))
```

```
[1] 0
```

Hide

```
merge$Number <- impute(merge$Number, fun = mean)
merge$MeetState <- impute(merge$MeetState, fun = mode)
merge$MeetTown <- impute(merge$MeetTown, fun = mode)
merge$Age <- impute(merge$Age, fun = mean)
merge$BodyweightKg <- impute(merge$BodyweightKg, fun = mean)
merge$Division <- impute(merge$Division, fun = mode)
merge$WeightClassKg <- impute(merge$WeightClassKg, fun = mode)
merge$Squat4Kg <- impute(merge$Squat4Kg, fun = mean)
merge$BestSquatKg <- impute(merge$BestSquatKg, fun = mean)
merge$Bench4Kg <- impute(merge$Bench4Kg, fun = mean)
merge$BestBenchKg <- impute(merge$BestBenchKg, fun = mean)
merge$Deadlift4Kg <- impute(merge$Deadlift4Kg, fun = mean)
merge$BestDeadliftKg <- impute(merge$BestDeadliftKg, fun = mean)
merge$TotalKg <- impute(merge$TotalKg, fun = mean)
merge$Place <- impute(merge$Place, fun = mode)
merge$Wilks <- impute(merge$Wilks, fun = mean)
merge$MHR <- impute(merge$MHR, fun = mean)
sum(is.na(merge))
```

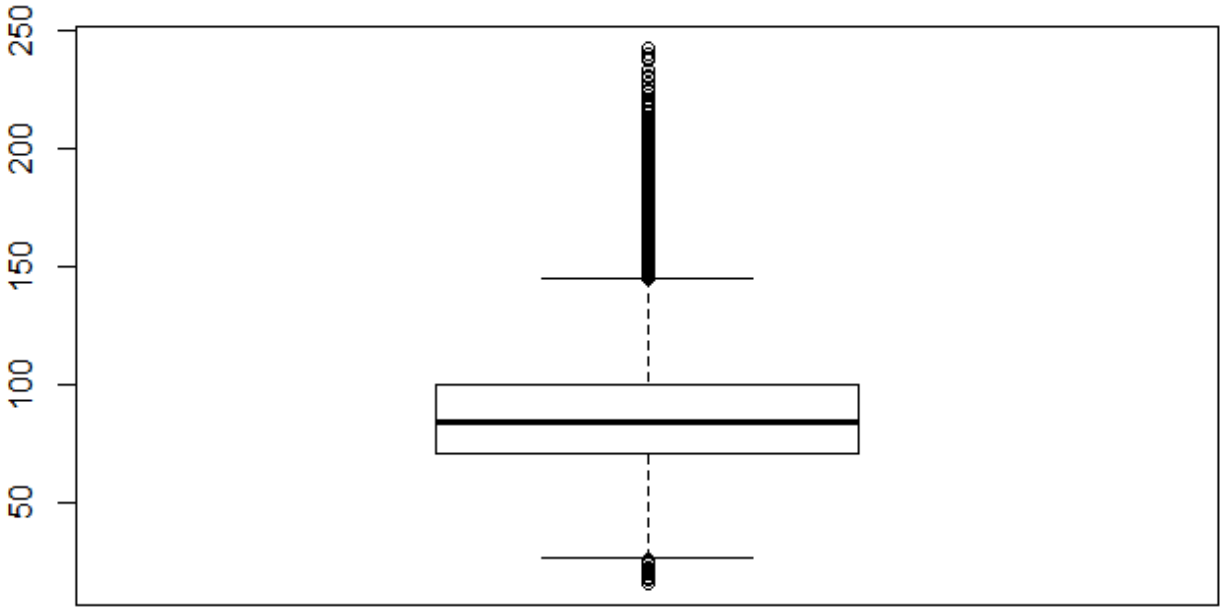
```
[1] 0
```

- colSums is used to identify the total number of NA in each column. When executed 23927 missing values are found in Age, 2402 in BodyweightKg, 385171 in Squat4Kg, 88343 in BestSquatKg, 384452 in Bench4Kg, 30050 in BestBenchKg, 383614 in Deadlift4Kg, 68567 in BestDeadliftKg, 23177 in TotalKg, 24220 in Wilks, 239267 in MHR.
- is.nan() is used to check for the NaN (Not a Number). The output shows zero meaning there are no errors in the merge.
- Imputation method is used for dealing the missing values. The numeric variables are replaced by mean and categorical/factor are replaced by mode.
- After imputing the missing values are zero.

Scan II

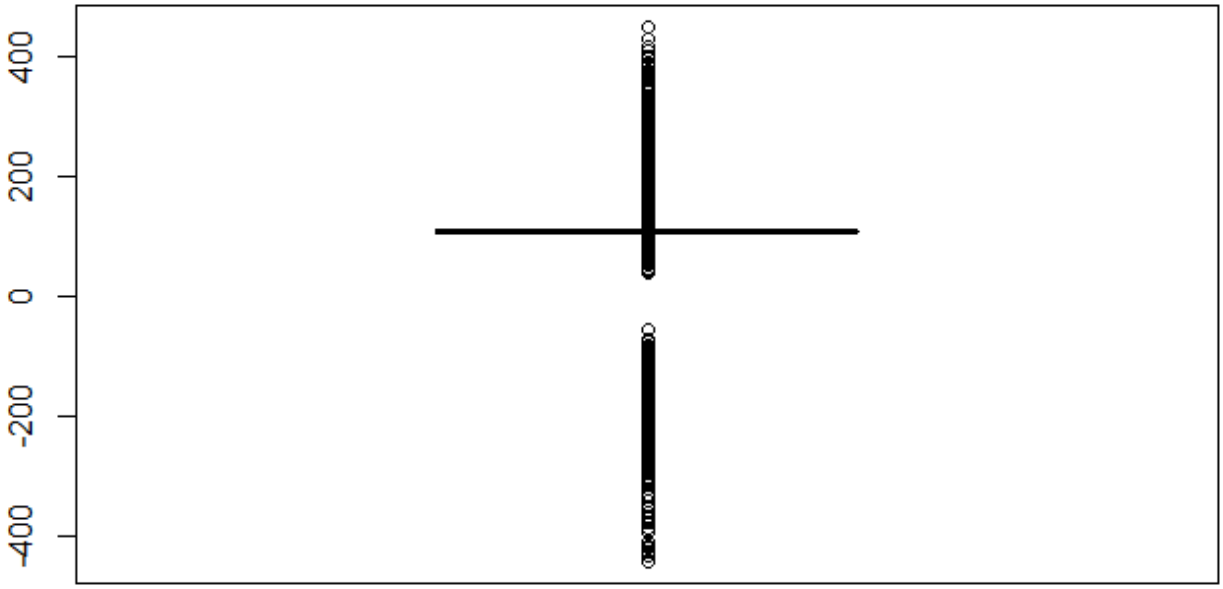
Hide

```
merge <- merge %>%  select(-c(Number, MHR, Age))
merge$BodyweightKg <- as.numeric(merge$BodyweightKg)
boxplot(merge$BodyweightKg)
```



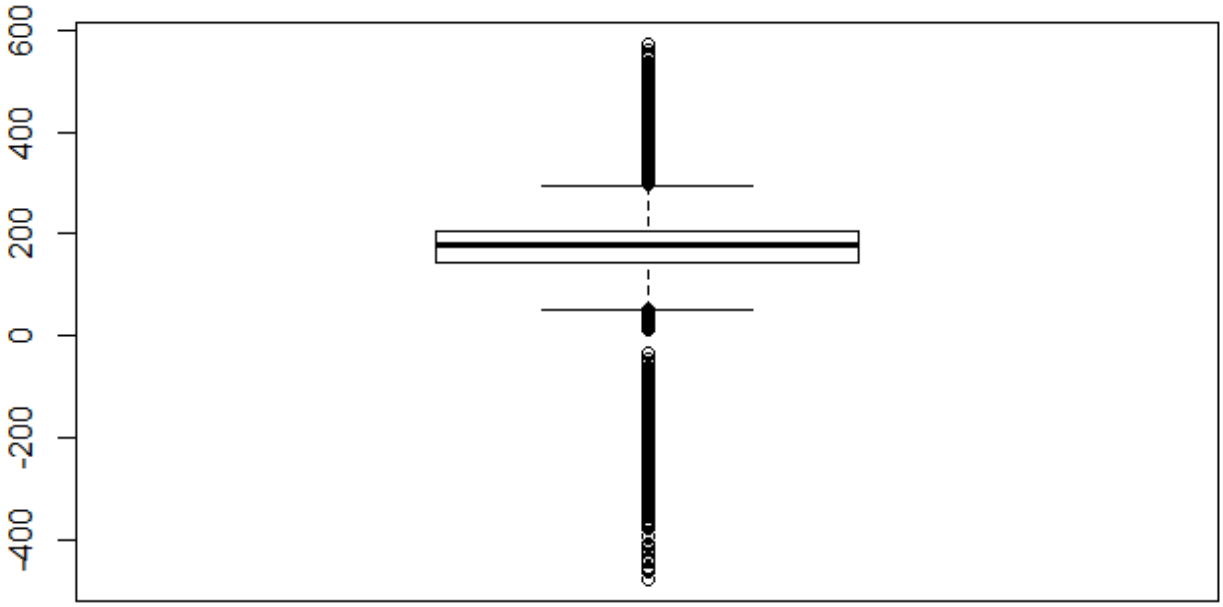
Hide

```
merge$Squat4Kg <- as.numeric(merge$Squat4Kg)
boxplot(merge$Squat4Kg)
```



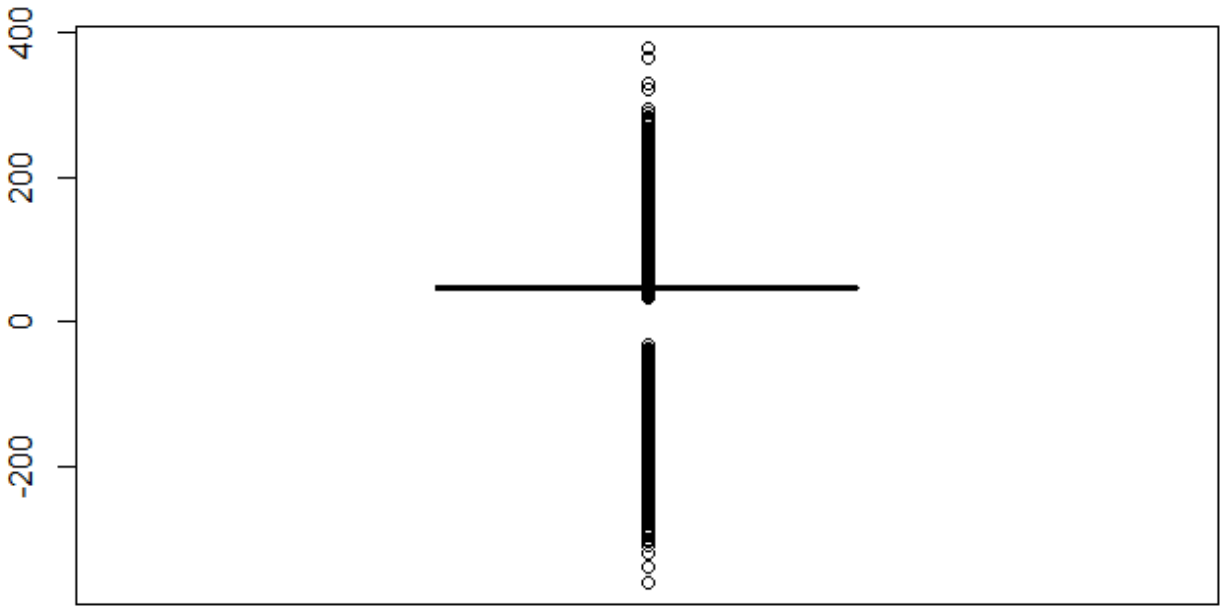
Hide

```
merge$BestSquatKg <- as.numeric(merge$BestSquatKg)
boxplot(merge$BestSquatKg)
```



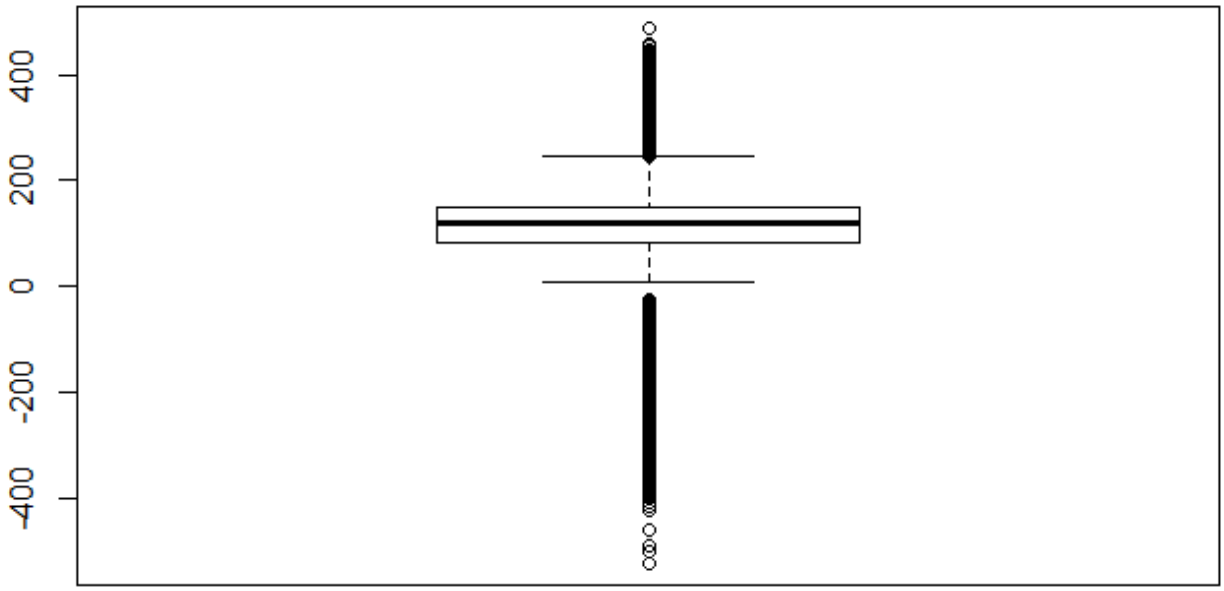
Hide

```
merge$Bench4Kg <- as.numeric(merge$Bench4Kg)
boxplot(merge$Bench4Kg)
```



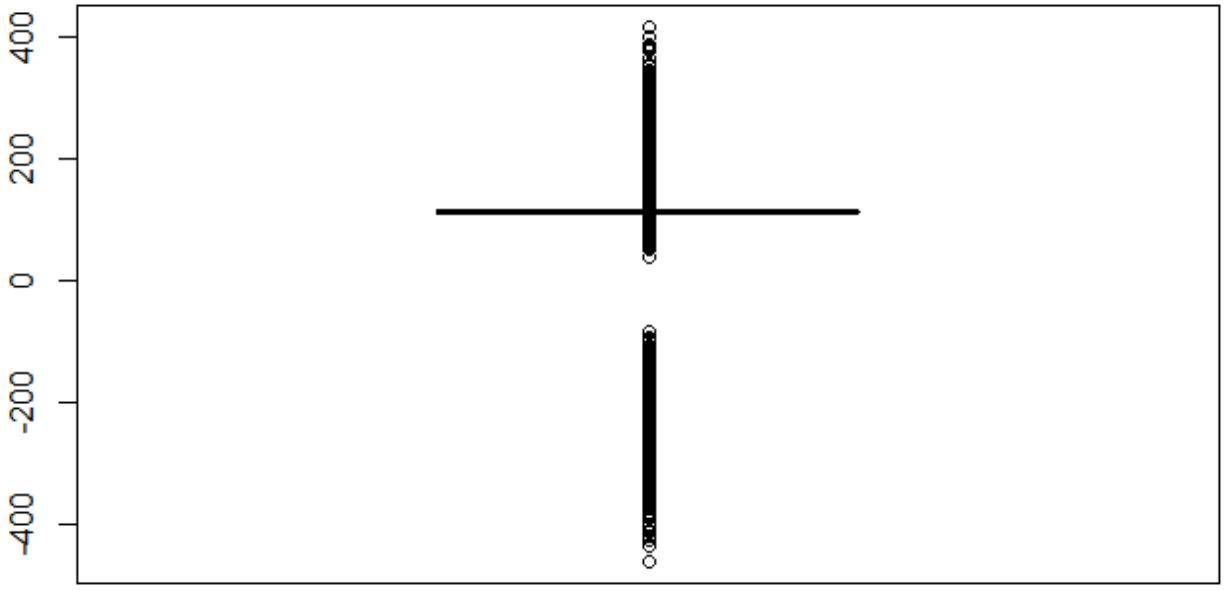
Hide

```
merge$BestBenchKg <- as.numeric(merge$BestBenchKg)
boxplot(merge$BestBenchKg)
```



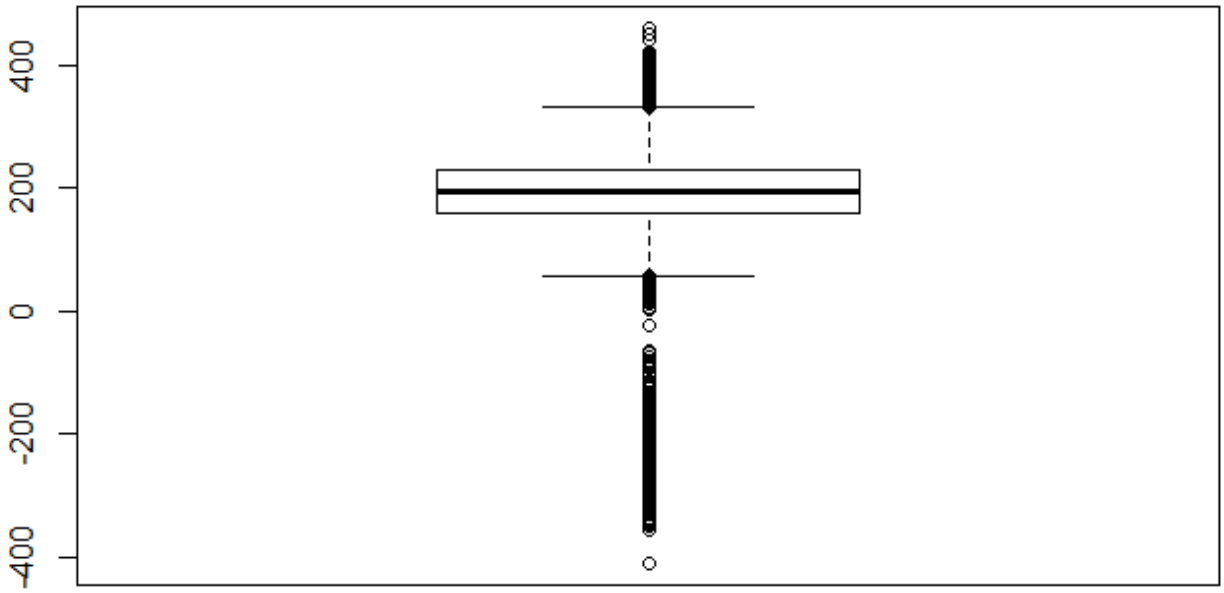
Hide

```
merge$Deadlift4Kg <- as.numeric(merge$Deadlift4Kg)
boxplot(merge$Deadlift4Kg)
```



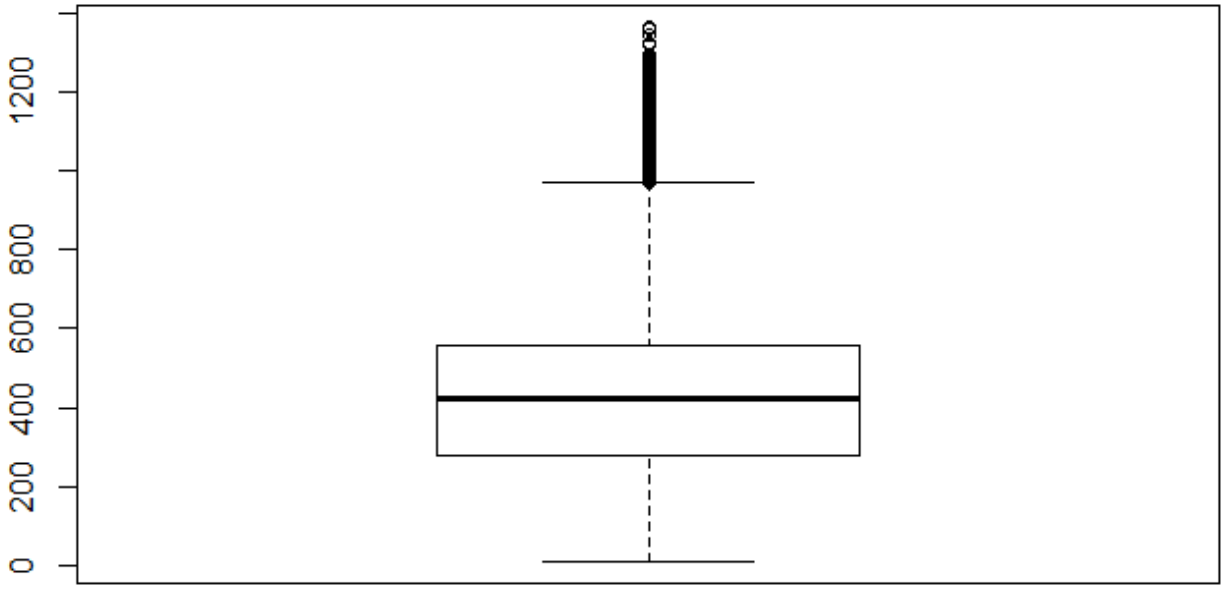
Hide

```
merge$BestDeadliftKg <- as.numeric(merge$BestDeadliftKg)
boxplot(merge$BestDeadliftKg)
```



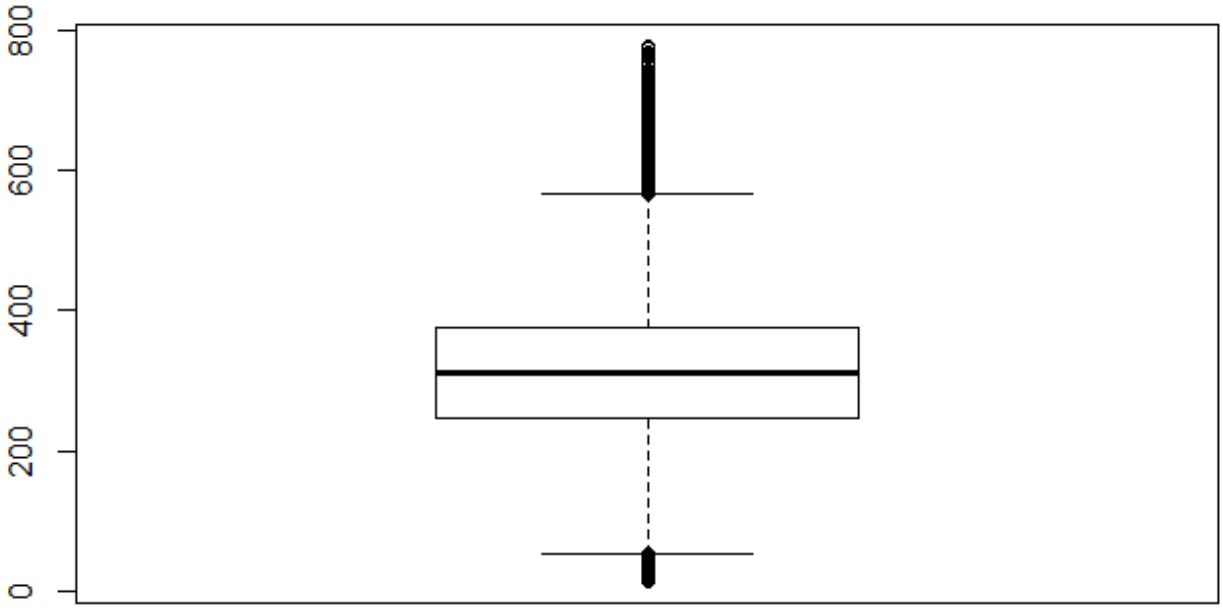
Hide

```
merge$TotalKg <- as.numeric(merge$TotalKg)
boxplot(merge$TotalKg)
```



Hide

```
merge$Wilks <- as.numeric(merge$Wilks)
boxplot(merge$Wilks)
```



Hide

```
cap <- function(x){
  quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
  x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
  x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
  x
}
MeetID_capped <- merge$MeetID %>% cap()
BodyweightKg_capped <- merge$BodyweightKg %>% cap()
Squat4Kg_capped <- merge$Squat4Kg%>% cap()
BestSquatKg_capped <- merge$BestSquatKg%>% cap()
Bench4Kg_capped <- merge$Bench4Kg%>% cap()
BestBenchKg_capped <- merge$BestBenchKg %>% cap()
BestDeadliftKg_capped <- merge$BestDeadliftKg %>% cap()
TotalKg_capped <- merge$TotalKg %>% cap()
Wilks_capped <- merge$Wilks %>% cap()
merge_sub <- merge %>% dplyr:: select(MeetID, BodyweightKg, Squat4Kg, BestSquatKg, Bench4Kg,
 BestBenchKg, Deadlift4Kg, BestDeadliftKg, TotalKg, Wilks)
summary(merge_sub)
```

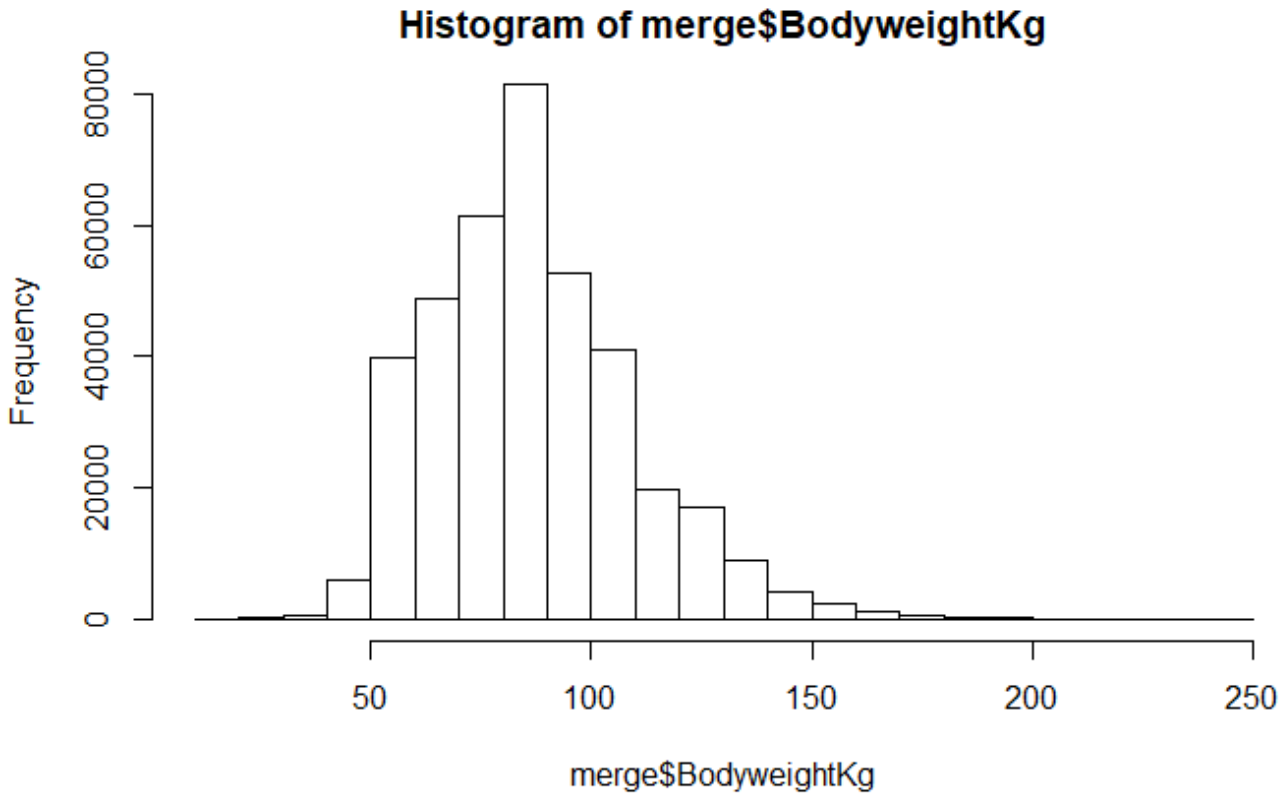
```
     MeetID      BodyweightKg       Squat4Kg       BestSquatKg        Bench4Kg        BestBen
chKg
 Min.   :   0   Min.   : 15.88   Min.   :-440.5   Min.   :-477.5   Min.   :-360.00   Min.   :
-522.5
 1st Qu.:2979   1st Qu.: 70.40   1st Qu.: 107.0   1st Qu.: 142.5   1st Qu.:  45.72   1st Qu.:
  82.5
 Median :5960   Median : 83.60   Median : 107.0   Median : 176.6   Median :  45.72   Median :
118.3
 Mean   :5143   Mean   : 86.93   Mean   : 107.0   Mean   : 176.6   Mean   :  45.72   Mean   :
118.3
 3rd Qu.:7175   3rd Qu.:100.00   3rd Qu.: 107.0   3rd Qu.: 204.1   3rd Qu.:  45.72   3rd Qu.:
147.5
 Max.   :8481   Max.   :242.40   Max.   : 450.0   Max.   : 573.8   Max.   : 378.75   Max.   :
488.5
  Deadlift4Kg     BestDeadliftKg      TotalKg           Wilks
 Min.   :-461.0   Min.   :-410.0   Min.   :  11.0   Min.   : 13.73
 1st Qu.: 113.6   1st Qu.: 158.8   1st Qu.: 280.0   1st Qu.:246.10
 Median : 113.6   Median : 195.0   Median : 424.0   Median :311.48
 Mean   : 113.6   Mean   : 195.0   Mean   : 424.0   Mean   :301.08
 3rd Qu.: 113.6   3rd Qu.: 227.5   3rd Qu.: 555.6   3rd Qu.:374.86
 Max.   : 418.0   Max.   : 460.4   Max.   :1365.3   Max.   :779.38
```

- Three variables (Age, Number, MHR) are filtered out in the dataset. This is because when all the numeric variables are executed for outliers, page numbers were exceeded as it does not meet the assignment principles.
- All numeric variables in the dataset are scanned for outliers. MeetID, BodyweightKg, Squat4Kg, BestSquatKg, Bench4Kg, BestBenchKg, Deadlift4Kg, BestDeadlift4Kg, TotalKg, Wilks has many outliers.
- Capping (a.k.a Winsorising) method is used for dealing the outliers. This means replacing the outliers with the nearest neighbors that are not outliers. 5% percentile of values replaces observations that lie outside the lower limit. 95% percentile of values replaces observations that lie above the upper limit.
- summary() is used to display the descriptive statistics

Transform

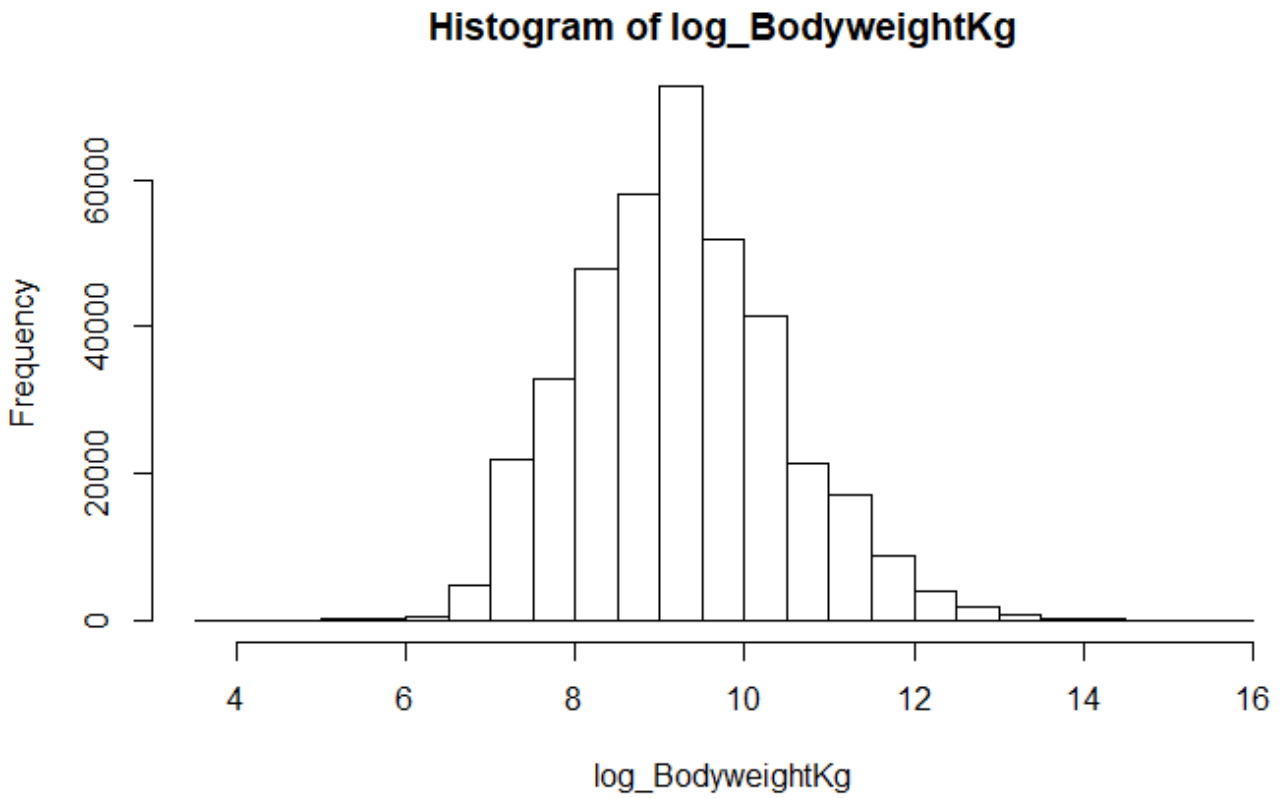
Hide

```
hist(merge$BodyweightKg)
```



Hide

```
log_BodyweightKg <- sqrt(merge$BodyweightKg)
hist(log_BodyweightKg)
```



Hide

```
summary(log_BodyweightKg)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.985   8.390   9.143   9.244  10.000  15.569
```

- Tranformation of BodyweightKg variable is choosen.
- The histogram shows left-skewed.
- Mathematical operations are performed to decrease the skewness and convert into the normal distribution. After applying the square root, it turned out to be symmetric one.