MATH2349 Semester 1, 2018

Code

Assignment 3

Vamika Pardeshi-s3701024, Ritwick Dev-s3702041

Required packages

Hide

```
library(readr)     #Useful for importing data
library(readxl)    #Useful for importing excel sheets
library(foreign)   #Useful for importing SPSS, SAS, STATA etc. data files
library(gdata)     #useful for providing various tools for data manipulation
library(rvest)     #Useful for scraping HTML data
library(tidyr)     #Useful for tidying data
library(dplyr)     #Useful for data manipulation
library(deductive) #Useful for deductive data correction and Imputation
library(validate)  #Useful for data validation
library(Hmisc)     #Useful in recoding missing values
library(stringr)   #Useful for sampling character vectors for string manipulations
library(lubridate) #Useful for working with dates and times
library(outliers)  #Useful in removing outliers
library(MVN)       #Useful for multivariate normality tests
library(infotheo)  #Useful for implementing measures of information theory based on several e
ntropy estimators
library(MASS)      #Useful to support Venables and Ripley
library(caret)     #Useful for model training process for complex regression & classification
 problems
library(mlr)       #Useful for providing unified interface for machine learning tasks in R
library(ggplot2)   #Useful for creating graphics based on 'The Grammar of graphics'
library(knitr)     #Useful for creating nice tables
library(raster)    #Useful in creating functions
library(mosaic)    #Useful for Descriptive Statistics
```

Executive Summary

The two datasets ‘races’ and ‘runs’ have been imported to R and then merged to form `Racing_data` . Furthermore, inspected the variable and the data structure of this combined dataset.Factorized the horse country variable and changed its labels. Since, the dataset was already following the tidy data principles, there was no need to tidy steps performed. Four new columns have been mutated, i.e,

1.Ratio of win to place-to check the probability of the horse in a particular race. Higher the Ratio of win to place, higher will be the chances of the horse to win the race. Although other factors also play a crucial role in deriving a probability of the winning horse like the speed of the horse, track condition, jockey weight,trainer of the horse and handicapped weights on the horse etc.

2.Horse weight-to get the measured weight of the horse subtracted the actual weight (jockey weight+ handicapped weight on the horse) from the declared weight (horse weight+ jockey weight+ handicapped weight on the horse).

3.Ratio of horse weight to actual weight-to calculate the proportion of the horse weight to the actual weight in a particular race, this affects the speed of the horse.

4.Average horse speed-to get the average speed of the horse in a race with respect to the length of the race track.

After this, scanned for the NA values and inconsistencies which were removed from the data. Although, many numeric columns are present in the dataset, but performing outlier treatment for all those columns is not relevant, as few of these columns are ID's of a race, horse, jockey and trainer etc. Detected and removed the outlier, firstly plotted a boxplot for the length behind column to determine the length by which the horse is lagging behind from the winning horse. Next, in order to compare a quantitative variable with a qualitative variable, bivariate boxplot was plotted for horse weight and going (track condition). In addition to that, a scatter plot was plotted for horse weight and actual weight variables. Moreover, a multivariate outlier treatment is performed on a particular class of the race with horse weight and actual weight variables. Lastly, tranformed the data by using Log and Square root transformation for reducing the skewness, on the variables ratio of horse to actual weight and declared weight, respectively.

Data

- The datasets contain data of thoroughbred horse racing in Hong Kong. Horse racing being a massive business in Hong Kong, resulting in betting pools bigger than all racetracks in US combined.
- There are two datasets i.e `races` and `runs`, presented in CSV format. * `races.csv` represents data on condition of each race that includes distance, track condition, distance and dividends paid. Whereas, `runs.csv` represents data of each horse running in each of the races mentioned in `races.csv`.
- In `runs.csv`, each line describes the characteristics of one horse run, in one of the races given in races.csv, and it contains the following variables-
  1. race_id- unique identifier for the race
  2. horse_no- the number assigned to this horse, in the race
  3. horse_id- unique identifier for this horse
  4. result- finishing position of this horse in the race
  5. won- whether horse won (1) or otherwise (0)
  6. lenghts_behind- finishing position, as the number of horse lengths behind the winner
  7. horse_age- current age of this horse at the time of the race
  8. horse_country- country of origin of the horse
  9. horse_type- sex of the horse, e.g. `Gelding`, `Mare`, `Horse`, `Rig`, `Colt`, `Filly`
  10.horse_rating- rating number assigned by HKJC to this horse at the time of the race
  11.declared_weight- declared weight of the horse and jockey, in lbs
  12.actual_weight- actual weight carried by the horse, in lbs
  13.draw- post position number of the horse in this race
  14.finish_time- finishing time of the horse in this race (in sec)
  15.win_odds- win odds for this horse at start of race
  16.place_odds- place odds for this horse at start of race (finishing in 1st, 2nd or 3rd position)
  17.trainer_id- unique identifier of the horse's trainer at the time of the race
  18.jockey_id- unique identifier of the jockey riding the horse in this race
- In `races.csv`, the condition of an individual race is described in each line, and it contains the following variables-
  1. race_id- unique identifier for the race
  2. date- date of the race, in YYYY-MM-DD format.
  3. venue- a 2-character string, representing which of the 2 race courses this race took place at: ST = Shatin,HV = Happy Valley
  4. race_no- race number of the race in the day's meeting
  5. config- race track configuration, mostly related to the position of the inside rail
  6. surface- a number representing the type of race track surface: 1 = dirt, 0 = turf
  7. distance- distance of the race, in metres
  8. going- track condition
  9. horse_ratings- range of horse ratings that may participate in this race
  10.prize- the winning prize, in HK Dollars
  11.race_class- a number representing the class of the race

12.place_combination1- placing horse no. 1st
13.place_combination2- placing horse no. 2nd
14.place_combination3- placing horse no. 3rd
15.place_dividend1- placing dividend paid (for place_combination1)
16.place_dividend2- placing dividend paid (for place_combination2)
17.place_dividend3- placing dividend paid (for place_combination3)
18.win_combination1- winning horse number

Source: `https://www.kaggle.com/gdaley/hkracing`
`https://ev.turnitin.com/app/carta/en_us/?student_user=1&lang=en_us&o=955457124&u=1072354159`

Read/Import Data

- Data has been imported to R, by using `read.csv()` function, from the package `readr` ,using the argument `stringsAsFactors = FALSE` as by default `read.csv` converts strings to factors.
- Imported datasets are saved as `Race` and `Run` .
- `Race` and `Run` are then merged using the key variable `race_id` .
- Using the generic function `left_join` , `Race` is added to the `Run` dataframe and the combined dataframe is renamed as `Racing_data` .
- Validated the first few rows of the dataframes using the generic function `head()` .

Hide

```
getwd()
```

```
[1] "C:/Users/Ritwick Dev/Documents/Data Preprocessing"
```

Hide

```
setwd("C:\\Users\\Ritwick Dev\\Documents\\Data Preprocessing")
Race <- read.csv("races.csv", stringsAsFactors = FALSE)
Run <- read.csv("runs.csv", stringsAsFactors = FALSE)
head(Race)
```

	race_id <int>	date <chr>	ve… <chr>	race_… <int>	config <chr>	surface <int>	distance <int>	going <chr>	horse_ratings <chr>
1	0	2/06/1997	ST	1	A	0	1400	GOOD TO FIRM	40-15
2	1	2/06/1997	ST	2	A	0	1200	GOOD TO FIRM	40-15
3	2	2/06/1997	ST	3	A	0	1400	GOOD TO FIRM	60-40
4	3	2/06/1997	ST	4	A	0	1200	GOOD TO FIRM	120-95
5	4	2/06/1997	ST	5	A	0	1600	GOOD TO FIRM	60-40
6	5	2/06/1997	ST	6	A	0	1200	GOOD TO FIRM	60-40

6 rows | 1-10 of 18 columns

Hide

```
head(Run)
```

	race_id <int>	horse_... <int>	horse_id <int>	result <int>	... <int>	lengths_behind <dbl>	horse_age <int>	horse_country <chr>	horse_t <chr>
1	0	1	3917	10	0	8.00	3	AUS	Gelding
2	0	2	2157	8	0	5.75	3	NZ	Gelding
3	0	3	858	7	0	4.75	3	NZ	Gelding
4	0	4	1853	9	0	6.25	3	SAF	Gelding
5	0	5	2796	6	0	3.75	3	GB	Gelding
6	0	6	3296	3	0	1.25	3	NZ	Gelding

6 rows | 1-10 of 18 columns

Hide

```
Racing_data <- Run %>% left_join(Race , by = "race_id")
head(Racing_data)
```

	race_id <int>	horse_... <int>	horse_id <int>	result <int>	... <int>	lengths_behind <dbl>	horse_age <int>	horse_country <chr>	horse_t <chr>
1	0	1	3917	10	0	8.00	3	AUS	Gelding
2	0	2	2157	8	0	5.75	3	NZ	Gelding
3	0	3	858	7	0	4.75	3	NZ	Gelding
4	0	4	1853	9	0	6.25	3	SAF	Gelding
5	0	5	2796	6	0	3.75	3	GB	Gelding
6	0	6	3296	3	0	1.25	3	NZ	Gelding

6 rows | 1-10 of 35 columns

Understand

- The base R function `class()` returned the `$names` , `$row_names` and `$class` of `Racing_data` .
- The base R function `length()` is used to check the number of columns in `Racing_data` dataframe.
- Using the base R function `dim()` returned the dimensions of the dataframe `Racing_data` .
- Generic function `str()` is used to see the detailed structure of the dataframe `Racing_data` .
- Using the base R function `factor()` changed the labels for the column `horse_country` .
- Moreover, ordered the levels using the argument `ordered=TRUE` for `Racing_data$horse_country` .
- `levels()` is used to check levels for the column `horse_country` .
- `dmy()` function is from the package `lubridate` is used to change the datatype from character to date.

Hide

```
class(Racing_data)
```

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

Hide

```
length(Racing_data)
```

```
[1] 35
```

Hide

```
dim(Racing_data)
```

```
[1] 79447    35
```

Hide

```
str(Racing_data)
```

```
'data.frame':   79447 obs. of  35 variables:
 $ race_id           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ horse_no          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ horse_id          : int  3917 2157 858 1853 2796 3296 911 2170 1730 2998 ...
 $ result            : int  10 8 7 9 6 3 12 1 13 14 ...
 $ won               : int  0 0 0 0 0 0 0 1 0 0 ...
 $ lengths_behind    : num  8 5.75 4.75 6.25 3.75 1.25 9.5 0 9.75 999 ...
 $ horse_age         : int  3 3 3 3 3 3 3 3 3 3 ...
 $ horse_country     : chr  "AUS" "NZ" "NZ" "SAF" ...
 $ horse_type        : chr  "Gelding" "Gelding" "Gelding" "Gelding" ...
 $ horse_rating      : int  60 60 60 60 60 60 60 60 60 60 ...
 $ declared_weight   : num  1020 980 1082 1118 972 ...
 $ actual_weight     : int  133 133 132 127 131 127 123 128 123 125 ...
 $ draw              : int  7 12 8 13 14 5 11 2 6 9 ...
 $ finish_time       : num  83.9 83.6 83.4 83.6 83.2 ...
 $ win_odds          : num  9.7 16 3.5 39 50 7 99 12 38 39 ...
 $ place_odds        : num  3.7 4.9 1.5 11 14 1.8 28 3.6 13 12 ...
 $ trainer_id        : int  118 164 137 80 9 54 55 47 75 109 ...
 $ jockey_id         : int  2 57 18 59 154 34 149 183 131 145 ...
 $ date              : chr  "2/06/1997" "2/06/1997" "2/06/1997" "2/06/1997" ...
 $ venue             : chr  "ST" "ST" "ST" "ST" ...
 $ race_no           : int  1 1 1 1 1 1 1 1 1 1 ...
 $ config            : chr  "A" "A" "A" "A" ...
 $ surface           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ distance          : int  1400 1400 1400 1400 1400 1400 1400 1400 1400 1400 ...
 $ going             : chr  "GOOD TO FIRM" "GOOD TO FIRM" "GOOD TO FIRM" "GOOD TO FIRM" ...
 $ horse_ratings     : chr  "40-15" "40-15" "40-15" "40-15" ...
 $ prize             : int  485000 485000 485000 485000 485000 485000 485000 485000 485000 48
5000 ...
 $ race_class        : int  5 5 5 5 5 5 5 5 5 5 ...
 $ place_combination1: int  8 8 8 8 8 8 8 8 8 8 ...
 $ place_combination2: int  11 11 11 11 11 11 11 11 11 11 ...
 $ place_combination3: int  6 6 6 6 6 6 6 6 6 6 ...
 $ place_dividend1   : num  36 36 36 36 36 36 36 36 36 36 ...
 $ place_dividend2   : num  25 25 25 25 25 25 25 25 25 25 ...
 $ place_dividend3   : num  18 18 18 18 18 18 18 18 18 18 ...
 $ win_combination1  : int  8 8 8 8 8 8 8 8 8 8 ...
```

Hide

```
attributes(Racing_data)
```

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

$row.names
   [1]    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17
18   19   20   21   22   23   24   25
  [26]   26   27   28   29   30   31   32   33   34   35   36   37   38   39   40   41   42
43   44   45   46   47   48   49   50
  [51]   51   52   53   54   55   56   57   58   59   60   61   62   63   64   65   66   67
68   69   70   71   72   73   74   75
  [76]   76   77   78   79   80   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  122  123  124  125
 [126]  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142
143  144  145  146  147  148  149  150
 [151]  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167
168  169  170  171  172  173  174  175
 [176]  176  177  178  179  180  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  222  223  224  225
 [226]  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242
243  244  245  246  247  248  249  250
 [251]  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267
268  269  270  271  272  273  274  275
 [276]  276  277  278  279  280  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  322  323  324  325
 [326]  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342
343  344  345  346  347  348  349  350
 [351]  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367
368  369  370  371  372  373  374  375
 [376]  376  377  378  379  380  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  422  423  424  425
 [426]  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442
443  444  445  446  447  448  449  450
 [451]  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467
468  469  470  471  472  473  474  475
 [476]  476  477  478  479  480  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  522  523  524  525
 [526]  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542
543  544  545  546  547  548  549  550
 [551]  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567
568  569  570  571  572  573  574  575
 [576]  576  577  578  579  580  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  622  623  624  625
 [626]  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642
643  644  645  646  647  648  649  650
 [651]  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667
```

```
668  669  670  671  672  673  674  675
 [676]  676  677  678  679  680  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  722  723  724  725
 [726]  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742
743  744  745  746  747  748  749  750
 [751]  751  752  753  754  755  756  757  758  759  760  761  762  763  764  765  766  767
768  769  770  771  772  773  774  775
 [776]  776  777  778  779  780  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  822  823  824  825
 [826]  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840  841  842
843  844  845  846  847  848  849  850
 [851]  851  852  853  854  855  856  857  858  859  860  861  862  863  864  865  866  867
868  869  870  871  872  873  874  875
 [876]  876  877  878  879  880  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  922  923  924  925
 [926]  926  927  928  929  930  931  932  933  934  935  936  937  938  939  940  941  942
943  944  945  946  947  948  949  950
 [951]  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966  967
968  969  970  971  972  973  974  975
 [976]  976  977  978  979  980  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 78447 entries ]

$names
 [1] "race_id"            "horse_no"           "horse_id"           "result"             "wo
n"                "lengths_behind"
 [7] "horse_age"          "horse_country"      "horse_type"         "horse_rating"       "dec
lared_weight"    "actual_weight"
[13] "draw"               "finish_time"        "win_odds"           "place_odds"         "tra
iner_id"         "jockey_id"
[19] "date"               "venue"              "race_no"            "config"             "sur
face"            "distance"
[25] "going"              "horse_ratings"      "prize"              "race_class"         "pla
ce_combination1" "place_combination2"
[31] "place_combination3" "place_dividend1"    "place_dividend2"    "place_dividend3"    "win
_combination1"
```

Hide

```
Racing_data$horse_country <- factor(Racing_data$horse_country,levels = c("AUS","NZ","SAF","G
B","USA","IRE","FR","CAN"),labels = c("AUSTRALIA","NEWZEALAND","SOUTHAFRICA","GREATBRITAIN",
"UNITEDSTATESAMERICA","IRELAND","FRANCE","CANADA"), ordered = TRUE)
levels(Racing_data$horse_country)
```

```
[1] "AUSTRALIA"           "NEWZEALAND"          "SOUTHAFRICA"         "GREATBRITAIN"
"UNITEDSTATESAMERICA"
[6] "IRELAND"             "FRANCE"              "CANADA"
```

Hide

```
Racing_data$date <- dmy(Racing_data$date)
head(Racing_data)
```

	race_id <int>	horse_... <int>	horse_id <int>	result <int>	... <int>	lengths_behind <dbl>	horse_age <int>	horse_country <ord>	horse_t <chr>
1	0	1	3917	10	0	8.00	3	AUSTRALIA	Gelding
2	0	2	2157	8	0	5.75	3	NEWZEALAND	Gelding
3	0	3	858	7	0	4.75	3	NEWZEALAND	Gelding
4	0	4	1853	9	0	6.25	3	SOUTHAFRICA	Gelding
5	0	5	2796	6	0	3.75	3	GREATBRITAIN	Gelding
6	0	6	3296	3	0	1.25	3	NEWZEALAND	Gelding

6 rows | 1-10 of 35 columns

Tidy & Manipulate Data I

- Tidying the data is not required as the dataframe is already conforming the tidy data principles that is-

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

- Mutated the column `Ratio_of_win_to_place` and `ratio_of_horse_to_actual` by dividing the column `win_odds` by `place_odds` and `horse_weight` by `actual_weight` , respectively.
- Calculated `Average_horse_speed` by didviding `distance` by `finish_time`
- Created `horse_weight` by substracting `actual_weight` from `declared_weight` ..

Hide

```
Racing_Data_mutate <- Racing_data %>% mutate(Ratio_of_win_to_place = win_odds / place_odds ,
                                             horse_weight = declared_weight - actual_weight ,
                                             Average_horse_speed = distance/finish_time,
                                             ratio_of_horse_to_actual = horse_weight / actual
_weight)
head(Racing_Data_mutate)
```

	race_id <int>	horse_... <int>	horse_id <int>	result <int>	... <int>	lengths_behind <dbl>	horse_age <int>	horse_country <ord>	horse_t <chr>
1	0	1	3917	10	0	8.00	3	AUSTRALIA	Gelding
2	0	2	2157	8	0	5.75	3	NEWZEALAND	Gelding
3	0	3	858	7	0	4.75	3	NEWZEALAND	Gelding
4	0	4	1853	9	0	6.25	3	SOUTHAFRICA	Gelding
5	0	5	2796	6	0	3.75	3	GREATBRITAIN	Gelding

	race_id <int>	horse_… <int>	horse_id <int>	result <int>	… <int>	lengths_behind <dbl>	horse_age <int>	horse_country <ord>	horse_t <chr>
6	0	6	3296	3	0	1.25	3	NEWZEALAND	Gelding

6 rows | 1-10 of 39 columns

Scan I

- Subsetted the dataframe `Racing_Data_mutate` using subset by column numbers on and named it `RDC` .
- Calculated the column mean of price for winning horse and named it `col_means` , got the index for each NA values and stored it in the variable `Index` and finally replaced the NA values with mean.
- Deleted the column price from the `Racing_Data_mutate` .
- Created a new variable `Racing_Data_New` by adding `RDC` to `Racing_Data_mutate` using the function `left_join` .
- Deleted the duplicate values from the dataframe `Racing_Data_New` .
- Performed these above steps as `impute()` function, from the package `hmisc` , was throwing an error.
- Imputed the variables `declared_weight` , `horse_weight` , `ratio_of_horse_to_actual` using the argument `fun=mean` and for variables `horse_country` and `horse_type` using `fun=mode` .
- Inspected the dataframe `Racing_Data_New` for NA values by using the function `is.na()` and calculated the total sum of NA values.
- Moreover created a custom function to check for any inconsistencies and special values.

Hide

```
RDC <- Racing_Data_mutate[,c(1,27)]
col_means <- colMeans(RDC[,-1], na.rm = TRUE)
```

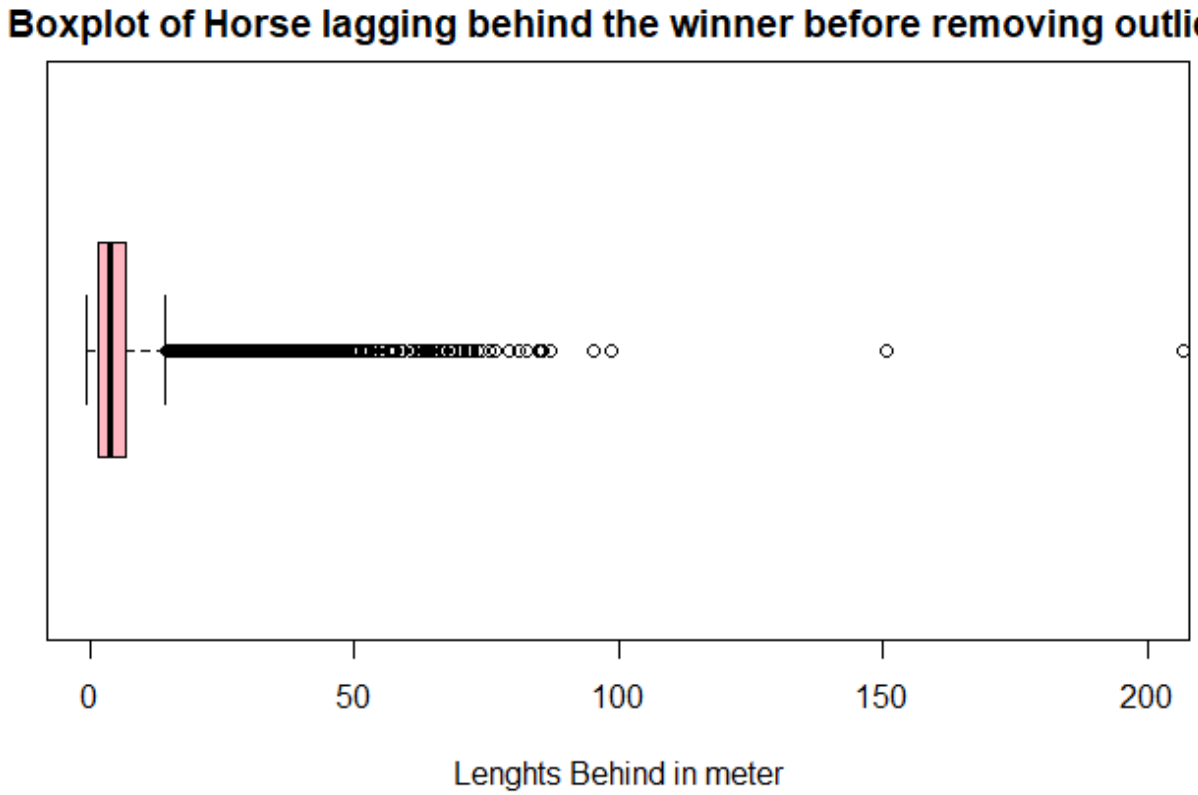
```
Error in base::colMeans(x, na.rm = na.rm, dims = dims, ...) :
  'x' must be an array of at least two dimensions
```

Scan II

- Using the `Racing_Data_New` dataframe, inspected the variable `lengths_behind` for outlier in by plotting a boxplot.
- Using the function `favstats` , calculated the descriptive statistics like mean, standard deviation and quartile values.
- Calculated the upper fence (Q3 + 1.5(IQR)) and lower fence (Q1 - 1.5(IQR)) using the Quartile values.
- Created a subset of the dataframe `Racing_Data_New` by removing the univariate outliers using the upper fence values and named it `Racing_Data_New_subset` .
- Plotted a boxplot for the variable `lengths_behind` again for dataframe `Racing_Data_New_subset` .
- A bivariate box plot is illustrated, by using a quantitative variable `horse_weight` and a qualitative variable `going` in the dataset `Racing_Data_New` .
- In order to get the scatter plot, `plot()` function is used and outliers in `horse_weight` and `actual_weight` have been detected from the dataframe `Racing_Data_New` . *To detect multivariate outliers, firstly subsetted the* `Racing_Data_New` *data, which is* `race class=13` *with the two variables* `horse_weight` *and* `actual weight` . Then, `mvn()` function is used to detect multivariate outliers with argument `multivariateOutlierMethod="quan"` and `showOutliers= TRUE` , using the chi-square distribution critical value approach and represent them on a plot.

Hide

```
boxplot(Racing_Data_New$lengths_behind , main="Boxplot of Horse lagging behind the winner bef
ore removing outlier", horizontal= TRUE , col="light pink" , xlab = "Lenghts Behind in meter"
 ,ylim=c(0,200))
```



Hide

```
favstats(~lengths_behind , data = Racing_Data_New)
```

min	Q1	median	Q3	m...	mean	sd	n	missing
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>
-0.5	1.75	4	6.75	999	6.108901	33.63621	79447	0

1 row

Hide

```
Upper_fence <- (6.75 + (3/2)*(6.75 - 1.75)) #q3 + (3/2)(q3-q1)
Upper_fence
```

```
[1] 14.25
```

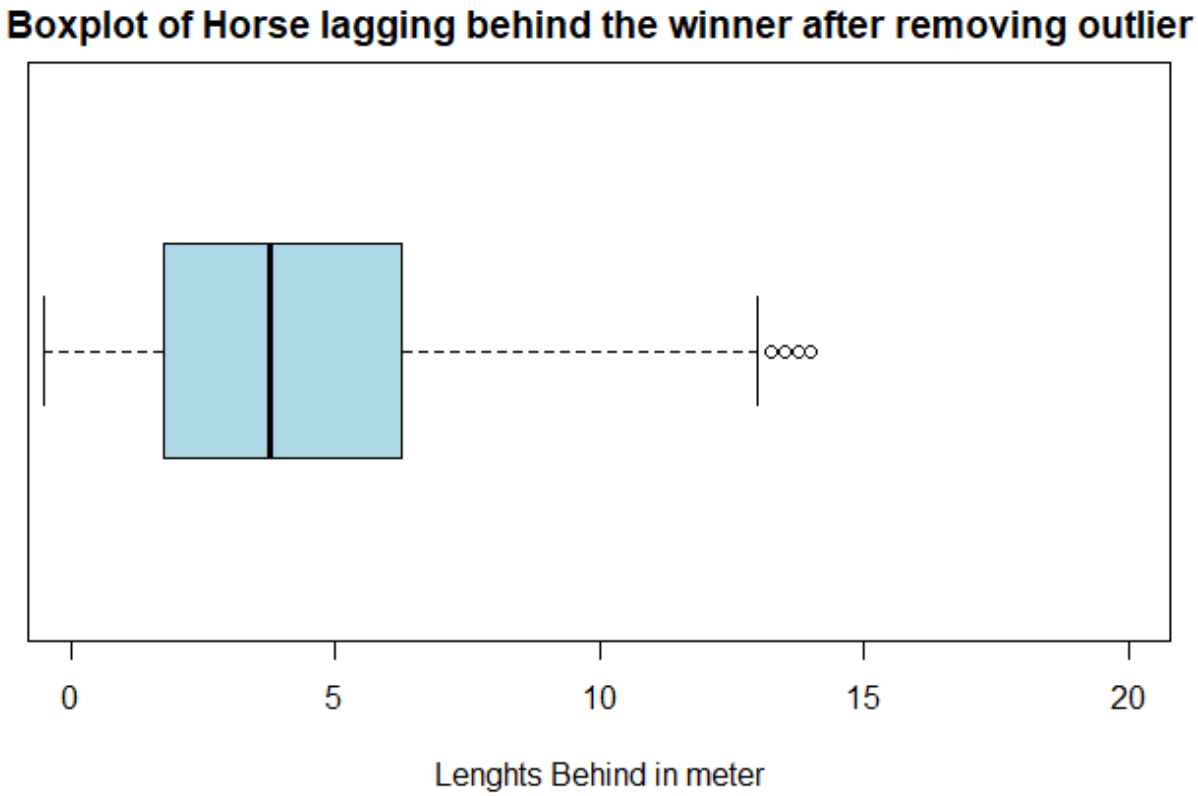
Hide

```
Lower_fence <- (1.75 - (3/2)*(6.75 - 1.75)) #q1 - (3/2)(q3-q1)
Lower_fence
```

```
[1] -5.75
```

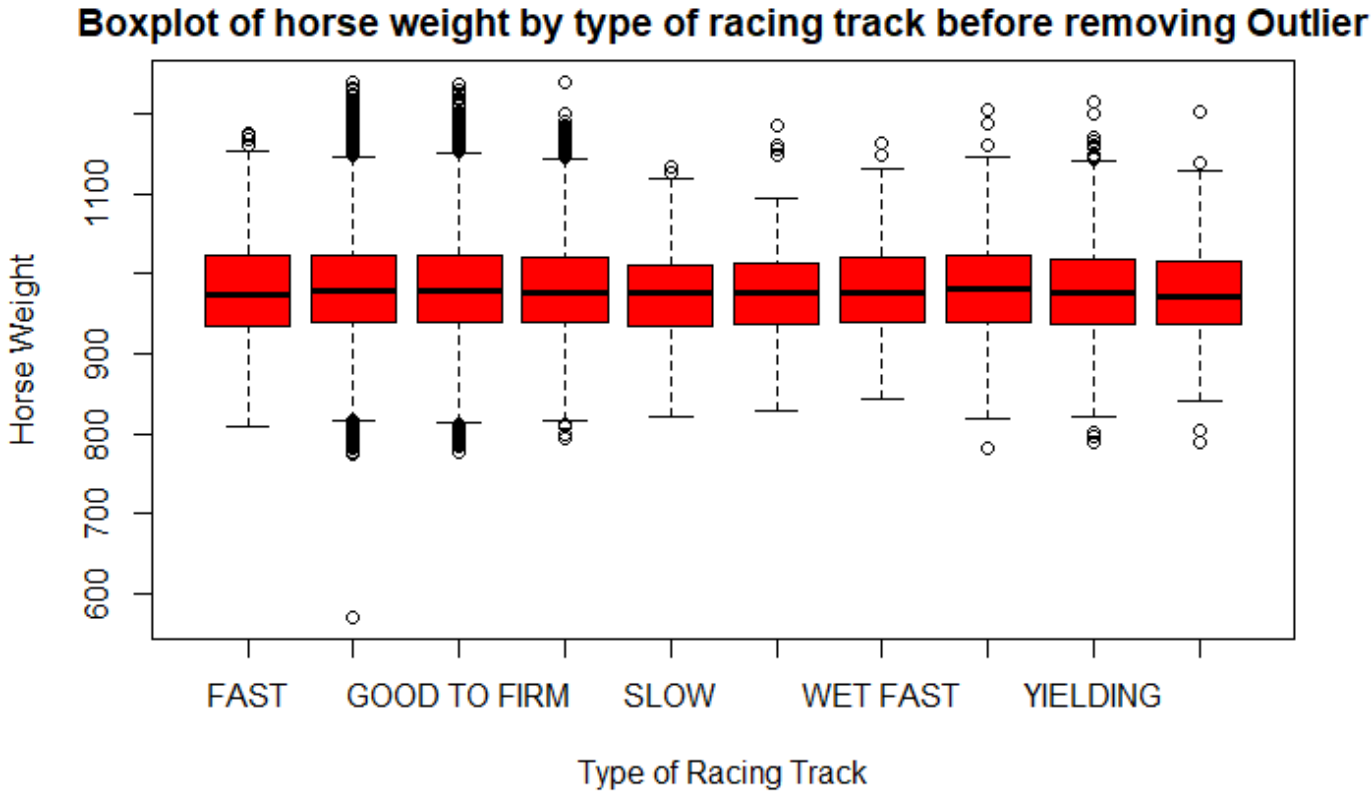
Hide

```
Racing_Data_New_subset <- subset(Racing_Data_New , (lengths_behind < Upper_fence))
boxplot(Racing_Data_New_subset$lengths_behind , main="Boxplot of Horse lagging behind the win
ner after removing outlier", horizontal= TRUE , col="light blue" , xlab = "Lenghts Behind in
 meter" ,ylim=c(0,20))
```



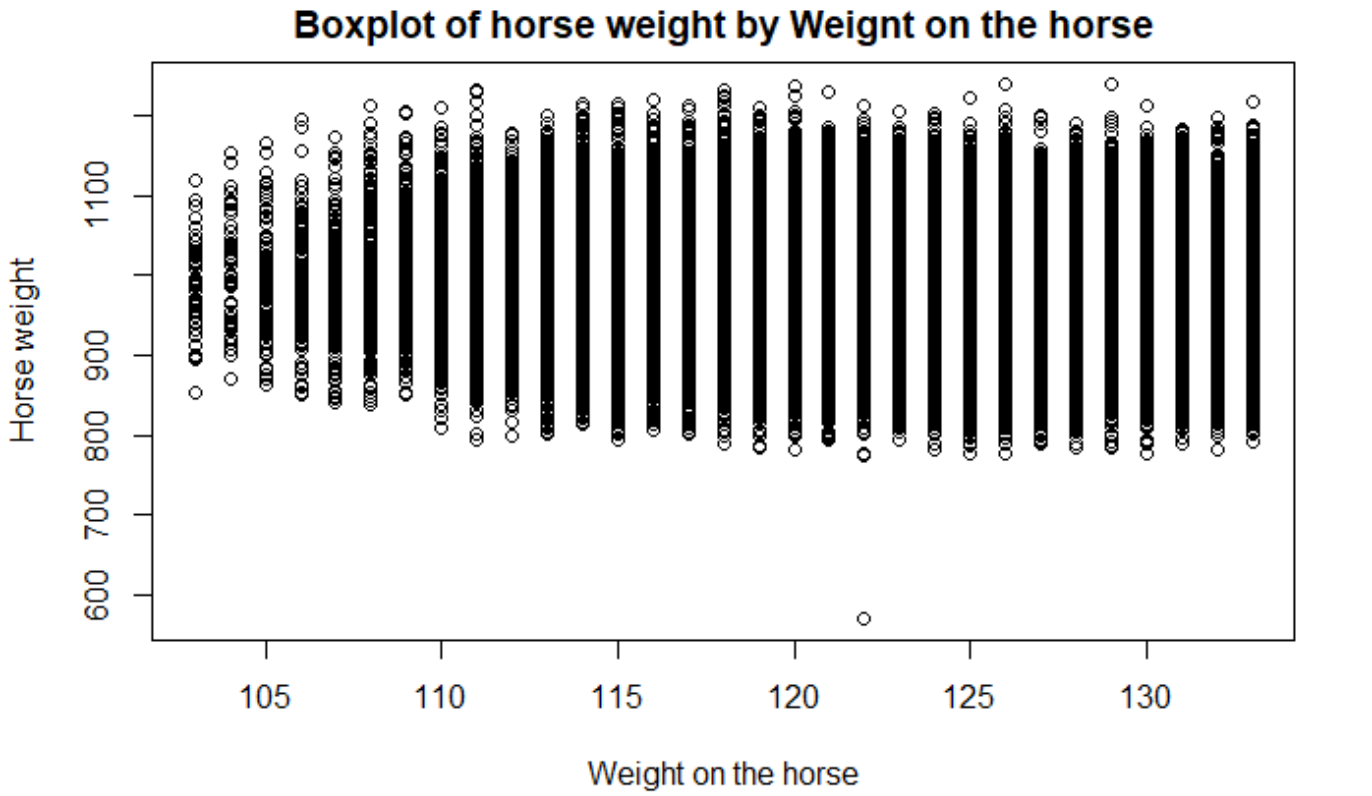
Hide

```
boxplot(Racing_Data_New$horse_weight ~ Racing_Data_New$going, main = "Boxplot of horse weight
 by type of racing track before removing Outlier", ylab = "Horse Weight", xlab = "Type of Rac
ing Track",col = "red")
```



Hide

```
Racing_Data_New %>% plot(horse_weight ~ actual_weight,data=., ylab="Horse weight", xlab="Weig
ht on the horse", main ="Boxplot of horse weight by Weignt on the horse")
```



Hide

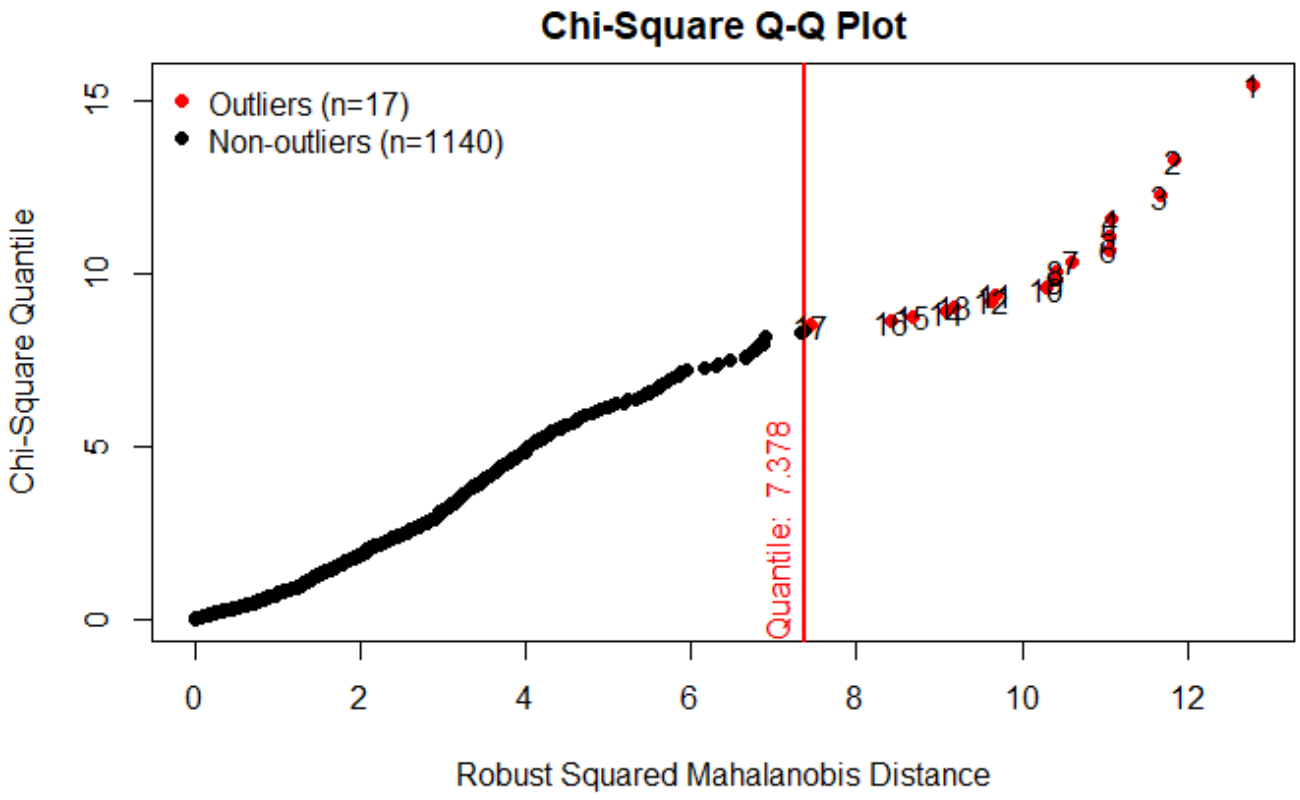
```
Racing_subset_for_mvn <- Racing_Data_New %>% filter( race_class == 13) %>%  dplyr::select(hor
se_weight, actual_weight)
head(Racing_subset_for_mvn)
```

horse_weight <dbl>	actual_weight <int>
949	133
949	127
996	127
1005	124
1138	124
957	123

6 rows

Hide

```
Mahalanobis_distance_QQ_plot <- mvn(data = Racing_subset_for_mvn, multivariateOutlierMethod =
 "quan", showOutliers = TRUE)
```



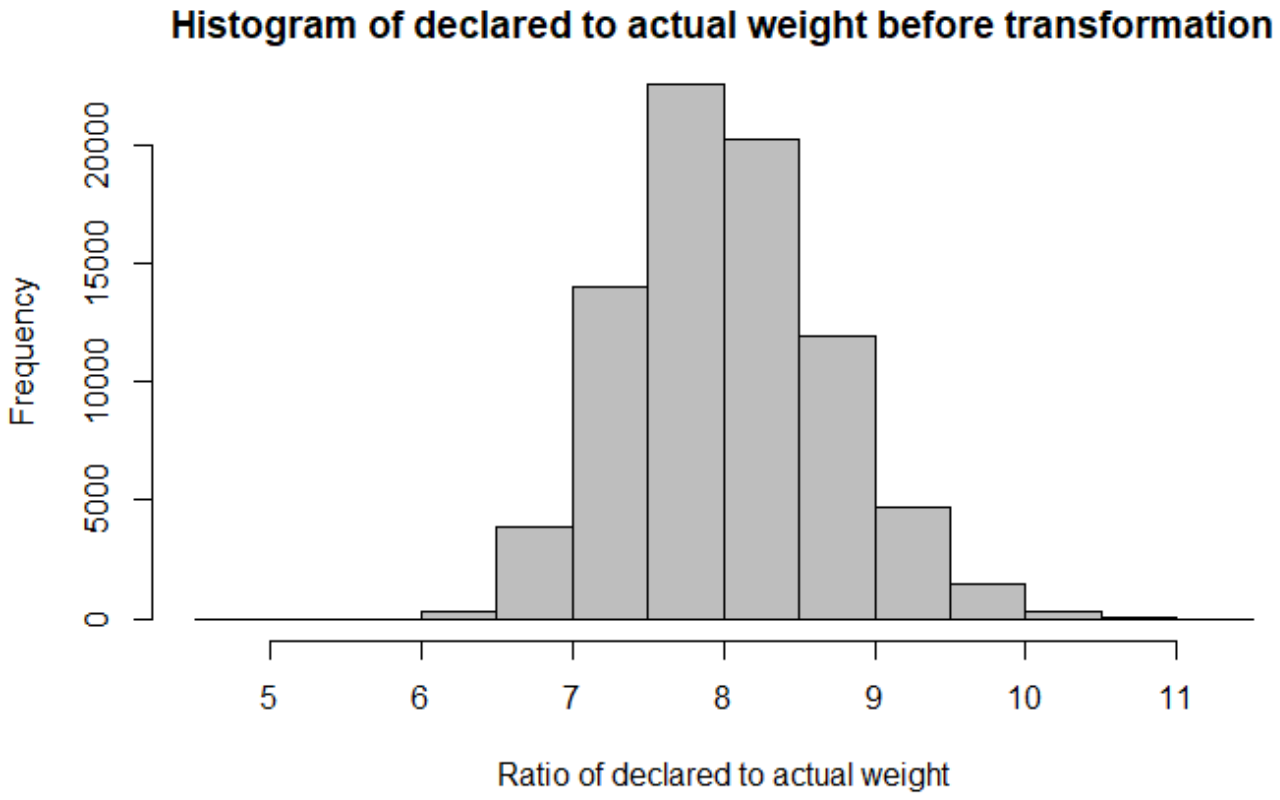
Transform

- By plotting histogram using the generic function `hist()` , it is noticed that `ratio_of_horse_to_actual` weight and `declared_weight` is slightly skewed to the right.
- Log and square root transformation is performed to reduce the skewness and named it as `Log_ratio_of_horse_to_actual` and `Square_Root` respectively.

- Plotted a histogram for `Log_ratio_of_horse_to_actual` and `Square_Root` using the function `hist()` with the argument `prob = TRUE` which plots a density function instead of frequency.
- Calculated the mean and standard deviation for Log_ratio_of_horse_to_actual and Square_Root, saved it as Mean, Sd and Mean1, S1 respectively.
- Created a sequence using the function `seq` and minimum & maximum values in the data set for `Log_ratio_of_horse_to_actual` and `Square_Root` .
- Using the function `dnorm` we calculated the density of the `Log_ratio_of_horse_to_actual` and `Square_Root` with the mean and standard deviation that we calculated for them respectively.
- Plotted a sequence of points at the specified coordinates using the generic function `points()` with a normal distribution overlay over the histogram.

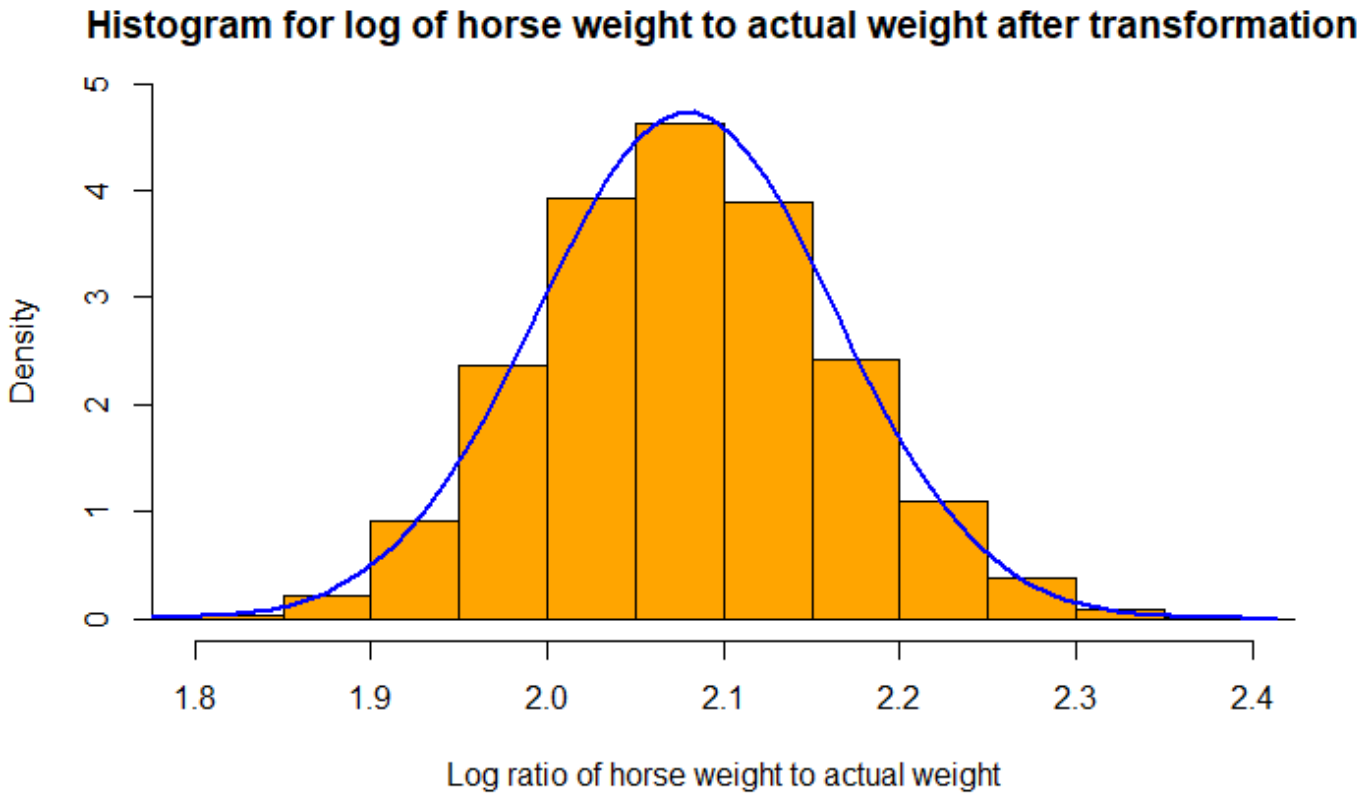
Hide

```
hist(Racing_Data_New$ratio_of_horse_to_actual,main = "Histogram of declared to actual weight
 before transformation" , xlab = "Ratio of declared to actual weight", col = "grey")
```



Hide

```
Log_ratio_of_horse_to_actual <- log(Racing_Data_New$ratio_of_horse_to_actual)
hist(Log_ratio_of_horse_to_actual ,xlim = c(1.8,2.4),ylim = c(0,5),prob = TRUE, main = "Histo
gram for log of horse weight to actual weight after transformation" , xlab = "Log ratio of ho
rse weight to actual weight", col = "orange" )
Mean <- mean(Log_ratio_of_horse_to_actual)
Sd <- sd(Log_ratio_of_horse_to_actual)
X <- seq(min(Log_ratio_of_horse_to_actual) , max(Log_ratio_of_horse_to_actual),0.01)
Y <- dnorm(X,Mean,Sd)
points(X,Y,type = "l", col = "blue",lwd = 2)
```



Hide

```
hist(Racing_Data_New$declared_weight,main = "Histogram of declared weight before transformati
on" , xlab = "Declared weight", col = "grey")
```

Hide

```
Square_Root <- sqrt(Racing_Data_New$declared_weight)
hist(Square_Root,xlim = c(30,37),ylim=c(0,0.5),prob = TRUE , main = "Histogram for Square roo
t of declared after transformation",
     xlab = "Square root of declared weight" , col = "light blue")
Mean1 <- mean(Square_Root)
Sd1 <- sd(Square_Root)
X1 <- seq(min(Square_Root) , max(Square_Root),0.01)
Y1 <- dnorm(X1,Mean1,Sd1)
points(X1,Y1,type = "l", col = "orange",lwd = 2)
```