FMLAssignment-1

Tarun

2024-02-05

```
library(readr)
# Reading CSV file using read_csv
chess_data <- read_csv("C:\\Users\\tarun\\OneDrive\\Desktop\\data set.csv")
```

```
## Rows: 377 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): filename, class
## dbl (6): width, height, xmin, ymin, xmax, ymax
##
## 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.
```

```
#creating quantitative variables vector
quantitative_vars <- c("width", "height", "xmin", "ymin", "xmax", "ymax")
#summary() for descriptive statistics
summary(chess_data[, quantitative_vars])
```

```
##      width          height          xmin            ymin            xmax
##  Min.   :416   Min.   :416   Min.   : 43.0   Min.   :  0.0   Min.   : 73.0
##  1st Qu.:416   1st Qu.:416   1st Qu.:101.0   1st Qu.: 52.0   1st Qu.:123.0
##  Median :416   Median :416   Median :171.5   Median :116.5   Median :195.5
##  Mean   :416   Mean   :416   Mean   :178.4   Mean   :130.2   Mean   :202.8
##  3rd Qu.:416   3rd Qu.:416   3rd Qu.:250.0   3rd Qu.:200.0   3rd Qu.:272.2
##  Max.   :416   Max.   :416   Max.   :338.0   Max.   :305.0   Max.   :369.0
##  NA's   :1     NA's   :1     NA's   :1       NA's   :1       NA's   :1
##       ymax
##  Min.   : 43.0
##  1st Qu.:101.8
##  Median :169.5
##  Mean   :184.9
##  3rd Qu.:254.2
##  Max.   :362.0
##  NA's   :1
```

```
# Print frequency tables for categorical variables
categorical_vars <- c("filename", "class")
for (var in categorical_vars) {
  cat("Frequency table for", var, ":\n")
  print(table(chess_data[[var]]))
  cat("\n")
}
```

```
## Frequency table for filename :
##
## 0b47311f426ff926578c9d738d683e76_jpg.rf.40183eae584a653181bbd795ba3c353f.jpg
##                                                                             1
## 1c0060ef868bdc326ce5e6389cb6732f_jpg.rf.f02cd668d26a53d9bf001497992b3657.jpg
##                                                                             1
## 2f6fb003bb89cd401322a535acb42f65_jpg.rf.66c0a46773a9cd583fb96c3df41a9e0c.jpg
##                                                                            31
## 410993714e325a1de3e394ffe860df3a_jpg.rf.657c49ca295ef54da23469189070a075.jpg
##                                                                            27
## 4e3117459d759798537eb52cf5bf534d_jpg.rf.ec961b62d4b0e131fae760ed1f80836b.jpg
##                                                                             1
## 5a35ba2ec3e0d0b2b12b1758a8ac29aa_jpg.rf.9dbdb057f6533c0c09c0eda0747fbc9e.jpg
##                                                                             1
## 654bb8835258b26c466b1c19893df451_jpg.rf.55fb7f23a4422a80793f01e152fabe4d.jpg
##                                                                             1
## 685b860d412b91f5d4f7f9e643b84452_jpg.rf.2d78193e4021ae5ffb49ecd1060bebd7.jpg
##                                                                             7
## 73a38a5c8f8f1b09f093f304660d5326_jpg.rf.65192fc4204952bfd1121ee212aade1e.jpg
##                                                                             1
## 749e9074a77f8d34d86e2218f26cdab4_jpg.rf.b39c00c032a7ecbb62b8792bbe05497e.jpg
##                                                                             1
## 7a34d8620235048917b28bcfd3b5572b_jpg.rf.450c577e3be66b5232c54ffc9ec9e6b7.jpg
##                                                                             1
## 8ff752f9ed443e6e49d495abfceb2032_jpg.rf.530a6c314a4848ead2b0ebc40e6ba651.jpg
##                                                                            32
## a3863d0be6002c21b20ac88817b2c56f_jpg.rf.0413d5178136ace55f588df9556c060a.jpg
##                                                                            22
## b4ff4132c8c85da97d8bf9a2a4ed3e3d_jpg.rf.ec790769b4818025b7652ca6aab9307e.jpg
##                                                                            27
## b526b661a33ff481231d1342aff2a266_jpg.rf.287d21a885ec3abeb6da818a6a9cd05b.jpg
##                                                                            21
## b9402881fa580d0eb8b9b98845417550_jpg.rf.7c401587706c0c03dab27877a8d22f55.jpg
##                                                                            27
## c4943d83c06a12ad5e0399d19514a4ca_jpg.rf.99b2d7e1faa204e71fdc71676040c4d6.jpg
##                                                                             1
## c5a012dfa72816098d23fc8baee67834_jpg.rf.6e0feae2ac0229ff5f20fc842852c81d.jpg
##                                                                             1
## cf4769d0586df6b3fb0dc618d9f8abe6_jpg.rf.81d8a4fa4e06ba4399292de7f5b5e300.jpg
##                                                                             1
## cfc306bf86176b92ffc1afbb98d7896f_jpg.rf.effd71a5dcd98ec0f24072af5f7c0a31.jpg
##                                                                             8
## d7887071e972604ddf5940d8eb2702e7_jpg.rf.5f20fe9a6c746d488d6d0478828478cb.jpg
##                                                                            15
## e4147f3d8819fc5d67a9f72596bd9e47_jpg.rf.ecc7863357d316634c6f22a2f0758303.jpg
##                                                                             1
## e4583d082076b2b549b3736ad1b193c9_jpg.rf.c64d9d89f8d479bf811e6b355b93e90e.jpg
##                                                                            14
## f1a24b6bb778ee11ba33687415aa84f2_jpg.rf.f2646d2d46b39f6510975f24d554bae1.jpg
##                                                                            14
## fdcd6ada676799da8a870f58fdf548db_jpg.rf.b0ea8552b6106bb4ab62ca8957fca40d.jpg
##                                                                            27
##                         IMG_0159_JPG.rf.f0d34122f8817d538e396b04f2b70d33.jpg
##                                                                            31
```

```
##                         IMG_0169_JPG.rf.1de291413bb78ef8ff0eaa8ffac38b06.jpg
##                                                                          29
##                         IMG_0170_JPG.rf.480e7164cb4727f6654402882f0ce942.jpg
##                                                                          32
##
## Frequency table for class :
##
## black-bishop   black-king black-knight   black-pawn  black-queen   black-rook
##           21           17           25           86           14           26
## white-bishop   white-king white-knight   white-pawn  white-queen   white-rook
##           25           15           25           88           14           20
```

```
#Transforming one variable that is squaring the width variable
head(chess_data)
```

```
## # A tibble: 6 x 8
##   filename                            width height class  xmin  ymin  xmax  ymax
##   <chr>                               <dbl>  <dbl> <chr> <dbl> <dbl> <dbl> <dbl>
## 1 5a35ba2ec3e0d0b2b12b1758a8ac29aa_j~   416    416 whit~   209   159   232   211
## 2 e4583d082076b2b549b3736ad1b193c9_j~   416    416 whit~    94   130   125   215
## 3 e4583d082076b2b549b3736ad1b193c9_j~   416    416 blac~   283   225   319   309
## 4 e4583d082076b2b549b3736ad1b193c9_j~   416    416 blac~   233   100   261   174
## 5 e4583d082076b2b549b3736ad1b193c9_j~   416    416 blac~   336   295   369   356
## 6 e4583d082076b2b549b3736ad1b193c9_j~   416    416 whit~   254   293   283   353
```

```
# Use a for loop to square the 'width' variable
for (i in 1:nrow(chess_data)) {
  chess_data$width_squared[i] <- chess_data$width[i]^2
}
```

```
## Warning: Unknown or uninitialised column: `width_squared`.
```

```
# Displaying the first few rows of the updated dataset
head(chess_data)
```

```
## # A tibble: 6 x 9
##   filename              width height class  xmin  ymin  xmax  ymax width_squared
##   <chr>                 <dbl>  <dbl> <chr> <dbl> <dbl> <dbl> <dbl>         <dbl>
## 1 5a35ba2ec3e0d0b2b12b~   416    416 whit~   209   159   232   211        173056
## 2 e4583d082076b2b549b3~   416    416 whit~    94   130   125   215        173056
## 3 e4583d082076b2b549b3~   416    416 blac~   283   225   319   309        173056
## 4 e4583d082076b2b549b3~   416    416 blac~   233   100   261   174        173056
## 5 e4583d082076b2b549b3~   416    416 blac~   336   295   369   356        173056
## 6 e4583d082076b2b549b3~   416    416 whit~   254   293   283   353        173056
```

```
#plotting Bar and scatter plot

# Display the structure of the dataset
str(chess_data)
```

```
## spc_tbl_ [377 x 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

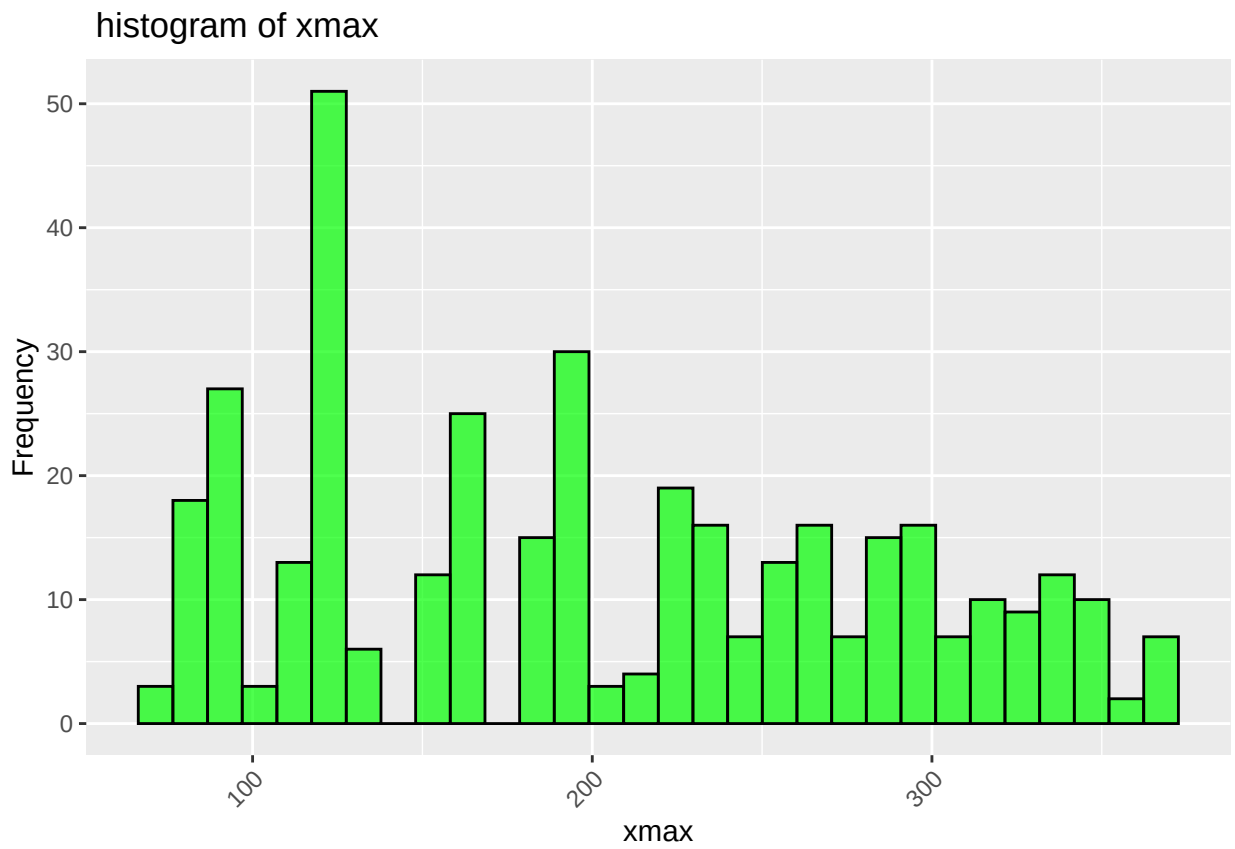
```
##  $ filename     : chr [1:377] "5a35ba2ec3e0d0b2b12b1758a8ac29aa_jpg.rf.9dbdb057f6533c0c09c0eda0747fb
##  $ width        : num [1:377] 416 416 416 416 416 416 416 416 416 416 ...
##  $ height       : num [1:377] 416 416 416 416 416 416 416 416 416 416 ...
##  $ class        : chr [1:377] "white-knight" "white-king" "black-king" "black-queen" ...
##  $ xmin         : num [1:377] 209 94 283 233 336 254 168 59 94 250 ...
##  $ ymin         : num [1:377] 159 130 225 100 295 293 153 78 253 203 ...
##  $ xmax         : num [1:377] 232 125 319 261 369 283 196 87 119 270 ...
##  $ ymax         : num [1:377] 211 215 309 174 356 353 211 133 302 250 ...
##  $ width_squared: num [1:377] 173056 173056 173056 173056 173056 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   filename = col_character(),
##   ..   width = col_double(),
##   ..   height = col_double(),
##   ..   class = col_character(),
##   ..   xmin = col_double(),
##   ..   ymin = col_double(),
##   ..   xmax = col_double(),
##   ..   ymax = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
library(ggplot2)

# Plotting a histogram for the 'width' variable
histogram_plot <- ggplot(chess_data, aes(x = xmax)) +
  geom_histogram(fill = "green", color = "black", alpha = 0.7) +
  labs(title = " histogram of xmax", x = "xmax", y = "Frequency") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
# Display the bar graph
print(histogram_plot)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

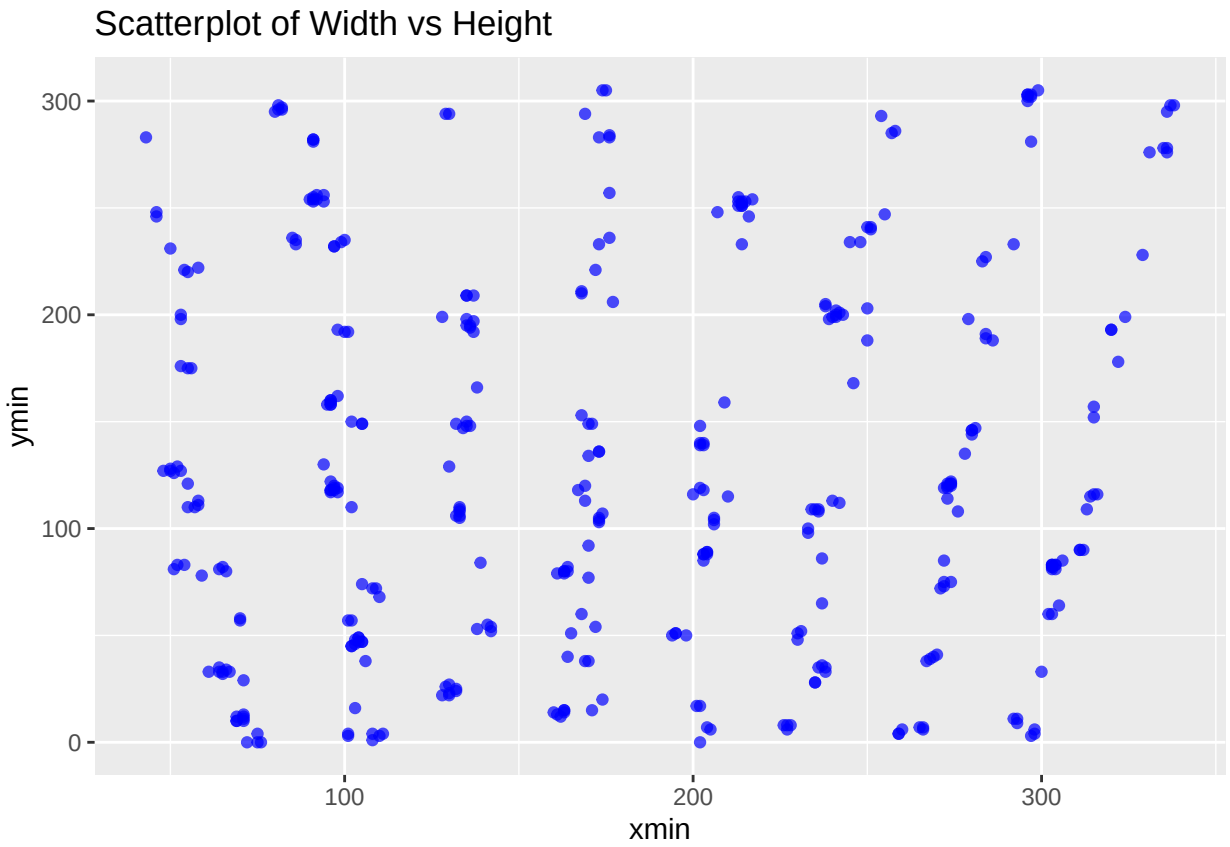
```
## Warning: Removed 1 rows containing non-finite values (`stat_bin()`).
```




```
# Scatterplot of 'width' vs 'height'
scatterplot <- ggplot(chess_data, aes(x = xmin , y = ymin)) +
  geom_point(color = "blue", alpha = 0.7) +
  labs(title = "Scatterplot of Width vs Height", x = "xmin", y = "ymin")

# Display the scatterplot
print(scatterplot)
```

```
## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
```


Scatterplot of Width vs Height
300
200
100
0
ymin
100
200
300
xmin