ex3_summer_winter.r

joris

2021-12-06

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
# Part 1

## Question 1a & Question 1b
swo <- read.csv("datasets_exam/summer_winter_olympics.csv")

dim(swo)
```

```
## [1] 146  17
```

```
nrow(swo)
```

```
## [1] 146
```

```
ncol(swo)
```

```
## [1] 17
```

```
head(swo)
```

```
##   X.9    Team..IOC.code. X..Summer   X X.1 X.2 Total X..Winter X.3 X.4 X.5
## 1   1  Afghanistan (AFG)        13   0   0   2     2         0   0   0   0
## 2   2      Algeria (ALG)        12   5   2   8    15         3   0   0   0
## 3   3    Argentina (ARG)        23  18  24  28    70        18   0   0   0
## 4   4      Armenia (ARM)         5   1   2   9    12         6   0   0   0
## 5   5  Australasia (ANZ)         2   3   4   5    12         0   0   0   0
```

```
## 6   6    Australia (AUS)        25 138 153 177   468        18   5   3   4
##   Total.1 X..Games X.6 X.7 X.8 Combined.total
## 1       0       13   0   0   2              2
## 2       0       15   5   2   8             15
## 3       0       41  18  24  28             70
## 4       0       11   1   2   9             12
## 5       0        2   3   4   5             12
## 6      12       43 143 156 181            480
```

```
colnames(swo) <- c(
    "index",
    "NOC",
    "summer_played",
    "summer_gold",
    "summer_silver",
    "summer_bronze",
    "summer_total",
    "winter_played",
    "winter_gold",
    "winter_silver",
    "winter_bronze",
    "winter_total",
    "both_played",
    "both_gold",
    "both_silver",
    "both_bronze",
    "both_total"
)

## Question 1c
table(swo$summer_played)
```

```
##
##  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
##  3  2  6  1 17  3  1  7  8  2  7 10 13  5  8 11  4  2  3  5  4  5  3  2  5  5
## 27
##  4
```

```
## Question 1d
for (column in tail(colnames(swo), -2)) {
    print(column)
    print("FREQUENCY TABLE")
    print(table(swo[[column]]))
}
```

```
## [1] "summer_played"
## [1] "FREQUENCY TABLE"
##
##  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
##  3  2  6  1 17  3  1  7  8  2  7 10 13  5  8 11  4  2  3  5  4  5  3  2  5  5
## 27
##  4
## [1] "summer_gold"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   5   6   7   8   9  12  13  14  15  16  17  18  21  23  25
```

```
##  47  16   9   8   2   3   6   3   1   3   1   1   2   1   1   1   2   1   2   1
##  26  28  30  33  37  39  42  43  45  47  49  51  56  59  64  72  77  81  88 101
##   1   1   1   1   2   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1
## 130 133 138 143 153 167 174 198 201 202 236 395 976
##   1   1   1   1   1   1   1   1   1   1   1   1   1
## [1] "summer_silver"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  15  17  18  20  21  24  25
##  22  27  16   5   5   7   4   4   3   4   1   1   1   1   1   1   1   1   2   1
##  26  27  29  30  32  33  38  42  49  52  54  59  67  68  73  82  84  85  94  99
##   1   1   1   2   1   1   1   1   2   1   1   1   2   1   1   2   1   2   1   1
## 122 126 129 144 153 164 166 182 223 272 319 758
##   1   1   1   2   1   1   1   1   1   1   1   1
## [1] "summer_bronze"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  17  19  20  21
##  23  20  13   5   6   5   2   1   3   2   4   8   2   1   1   3   1   1   1   1
##  24  25  27  28  29  35  36  38  39  40  43  45  53  55  65  68  69  78  80  81
##   1   1   1   3   2   2   1   1   1   1   1   1   1   2   1   1   1   1   1   1
## 104 117 119 120 125 127 128 142 165 176 177 185 217 246 272 296 666
##   1   1   1   1   1   1   1   2   1   1   1   1   1   1   1   1   1
## [1] "summer_total"
## [1] "FREQUENCY TABLE"
##
##    0    1    2    3    4    5    6    7    8    9   10   12   13   15   17   18
##    1   26   12    7    9    1    2    4    3    2    3    4    1    1    1    1
##   19   20   21   22   23   24   25   26   27   28   33   44   45   47   52   60
##    3    1    2    1    3    3    1    3    1    1    1    1    1    1    1    1
##   62   67   70   76   83   86   88   99  108  110  112  115  118  131  142  143
##    1    1    1    2    1    2    1    1    1    1    1    1    1    1    1    1
##  148  179  185  204  208  214  243  266  271  278  301  302  397  398  409  468
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  473  476  483  549  573  671  780 1010 2400
##    1    1    1    1    1    1    1    1    1
## [1] "winter_played"
## [1] "FREQUENCY TABLE"
##
##  0  1  2  3  4  5  6  7  8  9 10 11 13 14 15 16 17 18 19 20 22
## 45 11  6  6  4  3 19  7  4  3  4  2  2  1  1  4  2  4  2  4 12
## [1] "winter_gold"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   4   5   6   7   8   9  10  11  12  26  31  37  39  42  49  50  59
## 109   5   5   2   1   2   1   1   1   2   1   1   1   1   2   1   1   1   2   1
##  62  78  96 118
##   1   2   1   1
## [1] "winter_silver"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   6   7   8   9  15  17  22  31  34  36  38  40  55  57  62
## 104   5   6   3   4   3   1   1   1   1   2   1   1   1   1   1   3   1   1   1
##  78 102 111
```

```
##   2   1   1
## [1] "winter_bronze"
## [1] "FREQUENCY TABLE"
##
##   0   1   3   4   5   7   8   9  10  12  13  15  18  19  35  43  47  48  53  54
## 105   7   4   3   3   1   2   1   1   1   1   1   1   1   3   1   1   1   2   1
##  57  59  81  83 100
##   1   1   1   1   1
## [1] "winter_total"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   4   5   6   7   9  11  12  15  19  20  23  24  25  26  39  45  53
## 101   4   3   1   2   2   4   1   1   1   2   1   1   1   1   1   1   1   1   2
## 109 110 114 124 138 144 161 170 194 209 218 281 329
##   1   2   1   1   1   1   1   1   1   1   1   1   1
## [1] "both_played"
## [1] "FREQUENCY TABLE"
##
##  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
##  2  2  3  2  2  3  1  5  5  4 17  7  8  8  5  5  1  5  8  3  3  1  2  4  2  2
## 28 30 32 33 34 36 37 38 39 40 41 42 43 45 46 47 48 49
##  1  4  3  1  1  1  2  2  1  1  3  1  1  3  2  2  4  3
## [1] "both_gold"
## [1] "FREQUENCY TABLE"
##
##    0    1    2    3    4    5    6    7    8    9   10   13   14   15   17   18
##   46   16   10    8    1    2    7    2    1    3    1    2    1    1    2    2
##   21   23   25   26   30   35   36   38   39   42   43   51   52   54   67   70
##    2    2    1    1    1    1    1    2    1    1    1    1    1    1    1    1
##   72   77   88   97  107  114  121  140  143  167  174  182  192  193  213  233
##    1    1    1    1    1    1    1    1    2    1    1    1    1    1    1    1
##  235  246  252  473 1072
##    1    1    1    1    1
## [1] "both_silver"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   5   6   7   8   9  10  11  13  15  19  20  21  24  25  26
##  21  26  17   6   5   7   3   3   3   2   2   2   2   1   1   2   1   2   1   1
##  28  30  32  42  44  53  57  59  60  67  69  82  87  89  94  99 111 113 123 143
##   2   2   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 146 154 156 160 162 165 166 200 204 254 260 276 376 860
##   2   1   1   1   1   1   1   1   1   1   1   1   1   1
## [1] "both_bronze"
## [1] "FREQUENCY TABLE"
##
##   0   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  20
##  22  20  13   5   6   5   2   1   3   2   3   9   2   1   1   1   1   1   1   1
##  22  23  24  25  27  28  29  36  37  38  39  41  45  55  56  59  60  68  69  81
##   2   1   1   1   1   2   2   1   1   1   1   1   1   1   1   1   1   1   1   1
##  90  94 113 116 120 132 139 143 147 160 162 169 173 174 177 181 228 230 270 284
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 293 355 749
##   1   1   1
## [1] "both_total"
```

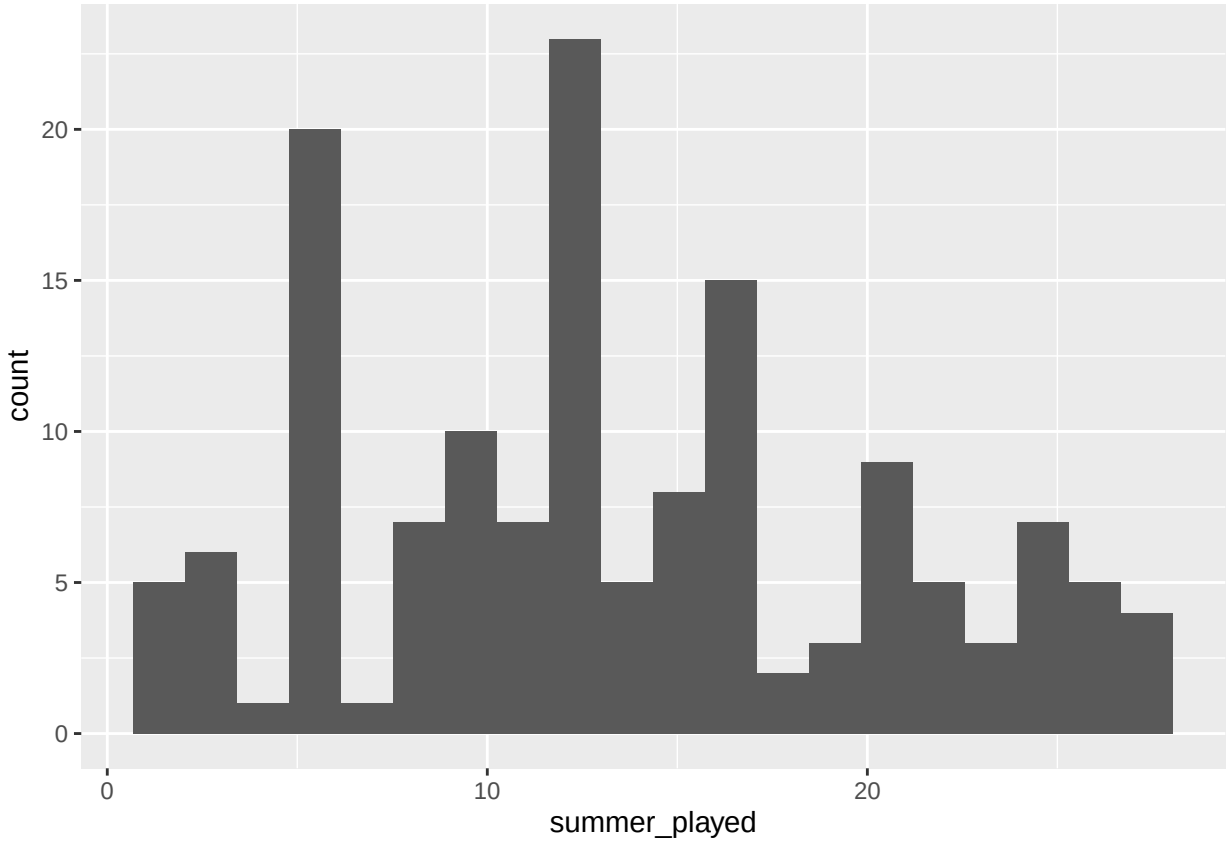
```
## [1] "FREQUENCY TABLE"
##
##    1    2    3    4    5    6    7    8    9   10   12   13   15   17   18   19
##   26   11    7   10    1    2    4    3    3    3    4    1    1    1    1    1
##   21   22   23   24   25   26   27   28   29   34   40   45   49   59   60   62
##    3    1    2    2    1    4    1    1    1    2    1    1    1    1    1    1
##   67   68   70   76   86   87   88   91  100  108  110  122  133  135  137  147
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  168  180  208  220  243  291  296  302  304  323  376  443  448  463  477  480
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
##  482  519  521  526  627  663  780  782  806 1204 2681
##    1    1    1    1    1    1    1    1    1    1    1
```

```
summary(swo)
```

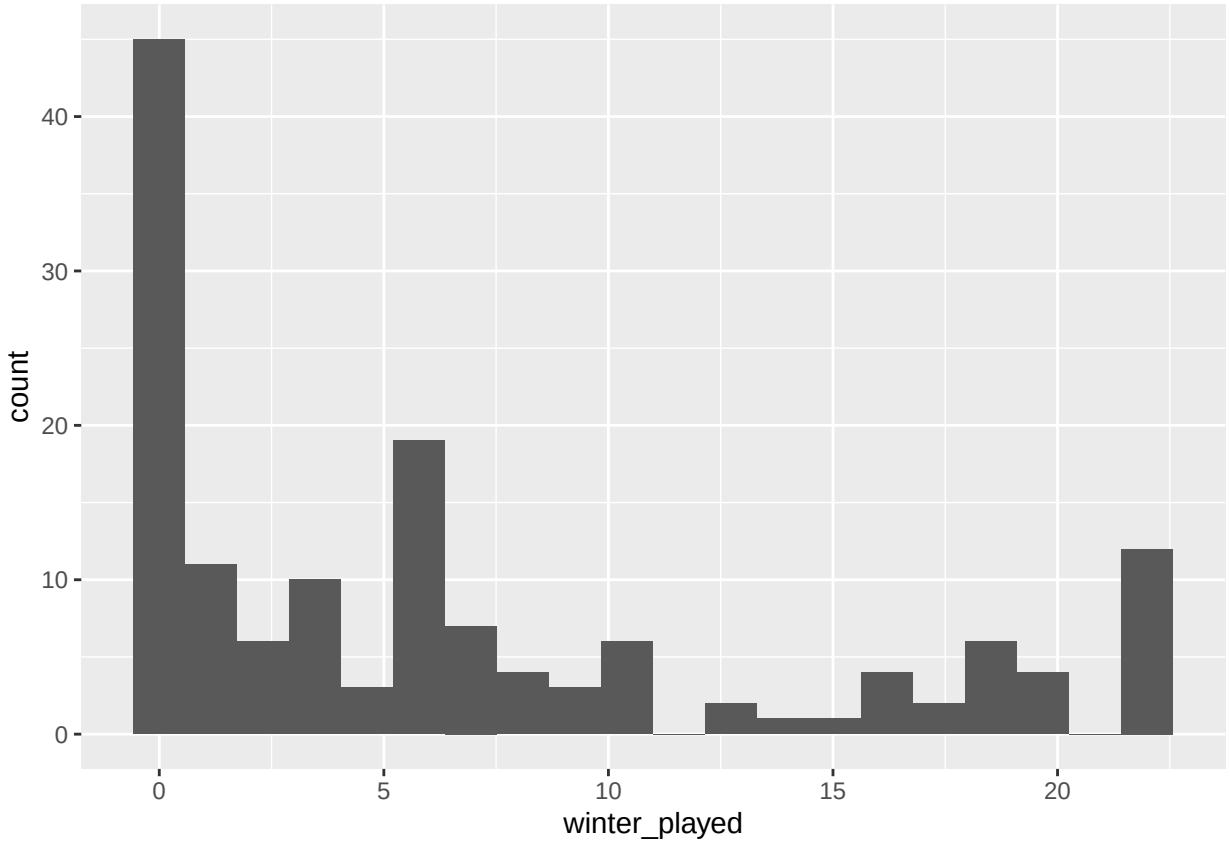
```
##      index            NOC            summer_played    summer_gold
##  Min.   :  1.00   Length:146         Min.   : 1.00   Min.   :  0.00
##  1st Qu.: 37.25   Class :character   1st Qu.: 8.00   1st Qu.:  0.00
##  Median : 73.50   Mode  :character   Median :13.00   Median :  3.00
##  Mean   : 73.50                      Mean   :13.38   Mean   : 32.94
##  3rd Qu.:109.75                      3rd Qu.:18.00   3rd Qu.: 23.00
##  Max.   :146.00                      Max.   :27.00   Max.   :976.00
##  summer_silver    summer_bronze     summer_total     winter_played
##  Min.   :  0.00   Min.   :  0.00   Min.   :   0.00   Min.   : 0.000
##  1st Qu.:  1.00   1st Qu.:  1.00   1st Qu.:   2.00   1st Qu.: 0.000
##  Median :  4.00   Median :  6.00   Median :  12.00   Median : 5.000
##  Mean   : 32.71   Mean   : 35.13   Mean   : 100.78   Mean   : 6.596
##  3rd Qu.: 26.75   3rd Qu.: 28.75   3rd Qu.:  85.25   3rd Qu.:10.000
##  Max.   :758.00   Max.   :666.00   Max.   :2400.00   Max.   :22.000
##   winter_gold      winter_silver     winter_bronze      winter_total
##  Min.   :  0.000   Min.   :  0.000   Min.   :  0.000   Min.   :  0.00
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.00
##  Median :  0.000   Median :  0.000   Median :  0.000   Median :  0.00
##  Mean   :  6.568   Mean   :  6.555   Mean   :  6.493   Mean   : 19.62
##  3rd Qu.:  0.750   3rd Qu.:  1.750   3rd Qu.:  1.000   3rd Qu.:  4.75
##  Max.   :118.000   Max.   :111.000   Max.   :100.000   Max.   :329.00
##   both_played      both_gold        both_silver      both_bronze
##  Min.   : 1.00   Min.   :   0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:11.00   1st Qu.:   0.00   1st Qu.:  1.00   1st Qu.:  1.00
##  Median :15.00   Median :   3.00   Median :  4.00   Median :  6.50
##  Mean   :19.98   Mean   :  39.51   Mean   : 39.27   Mean   : 41.62
##  3rd Qu.:26.00   3rd Qu.:  24.50   3rd Qu.: 28.00   3rd Qu.: 29.00
##  Max.   :49.00   Max.   :1072.00   Max.   :860.00   Max.   :749.00
##    both_total
##  Min.   :   1.00
##  1st Qu.:   2.25
##  Median :  12.00
##  Mean   : 120.40
##  3rd Qu.:  87.75
##  Max.   :2681.00
```

```
# Part 4
# Question 4a
hist_summer_played <- swo %>%
    ggplot(aes(summer_played)) +
```

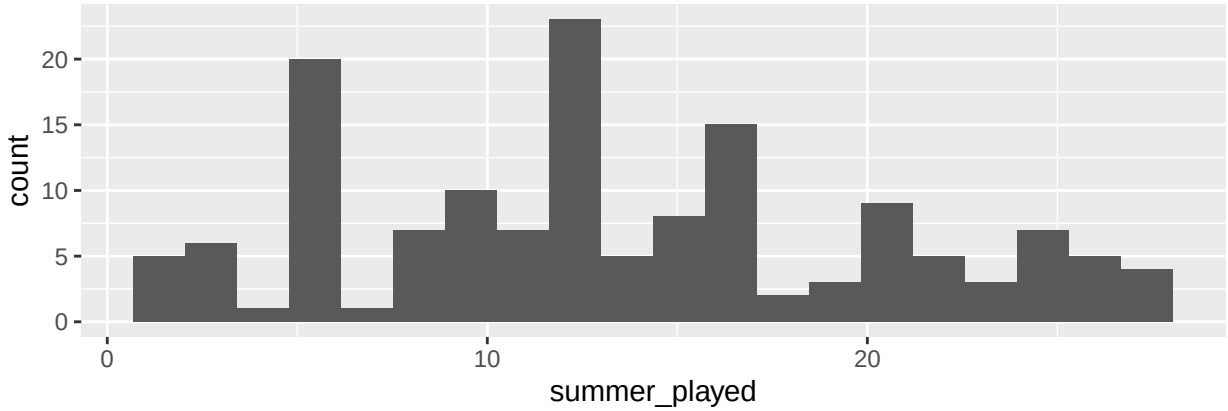
```
    geom_histogram(bins = 20)
hist_summer_played
```

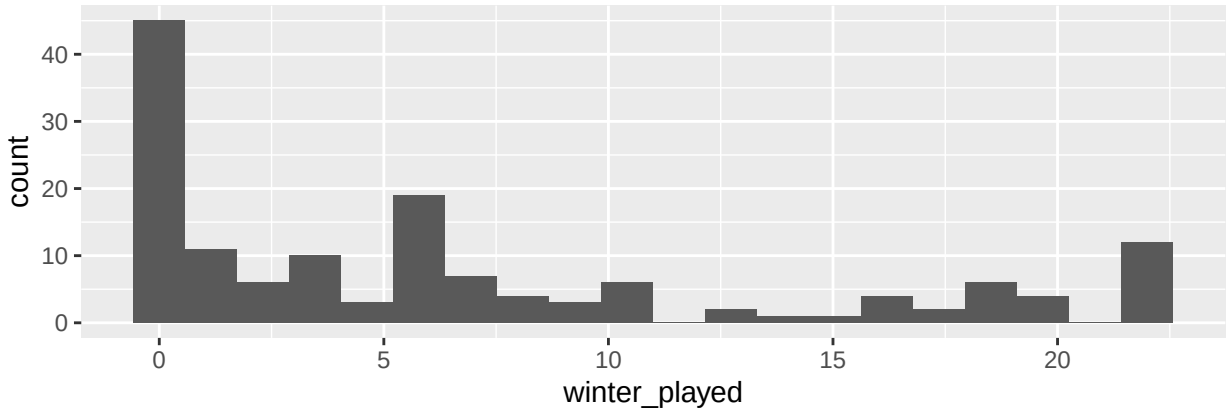



```
# Question 4b
hist_winter_played <- swo %>%
    ggplot(aes(winter_played)) +
    geom_histogram(bins = 20)
hist_winter_played
```

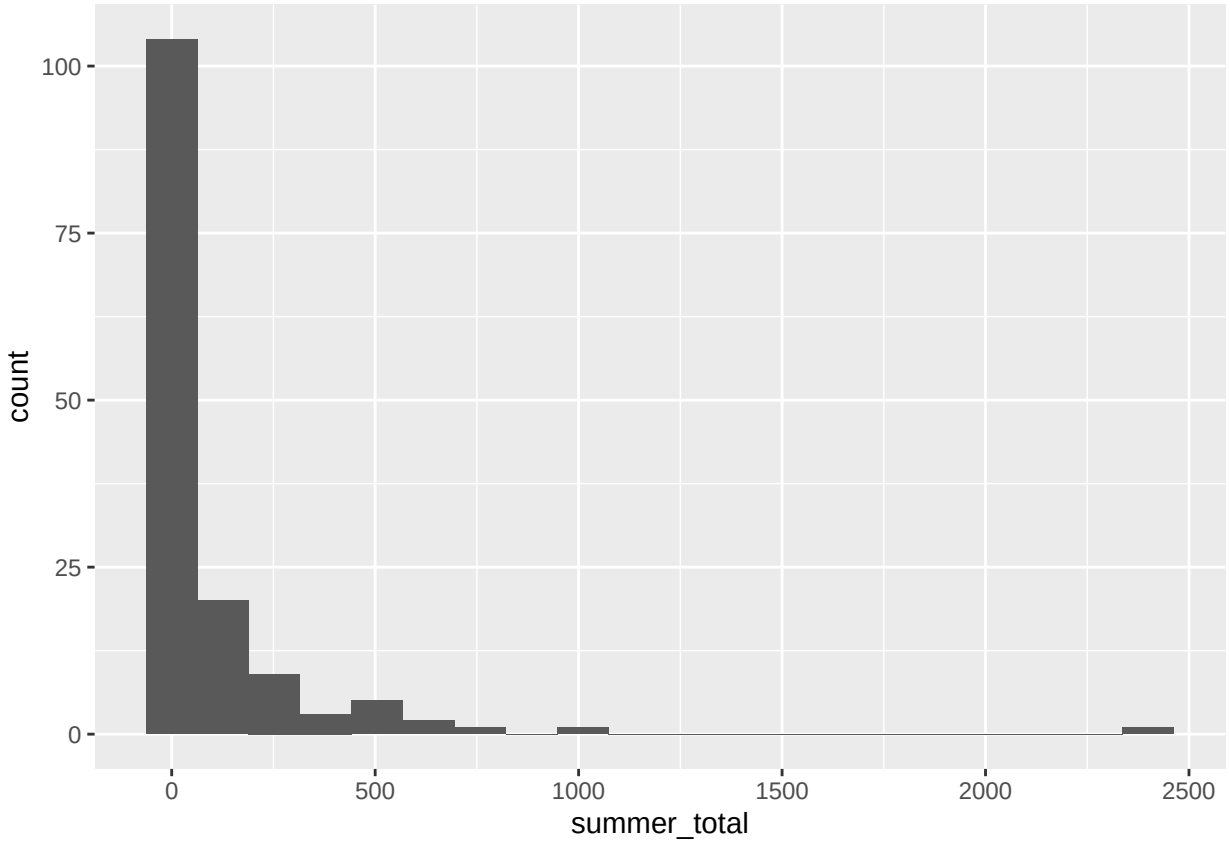



```
# Question 4c
grid.arrange(hist_summer_played, hist_winter_played)
```

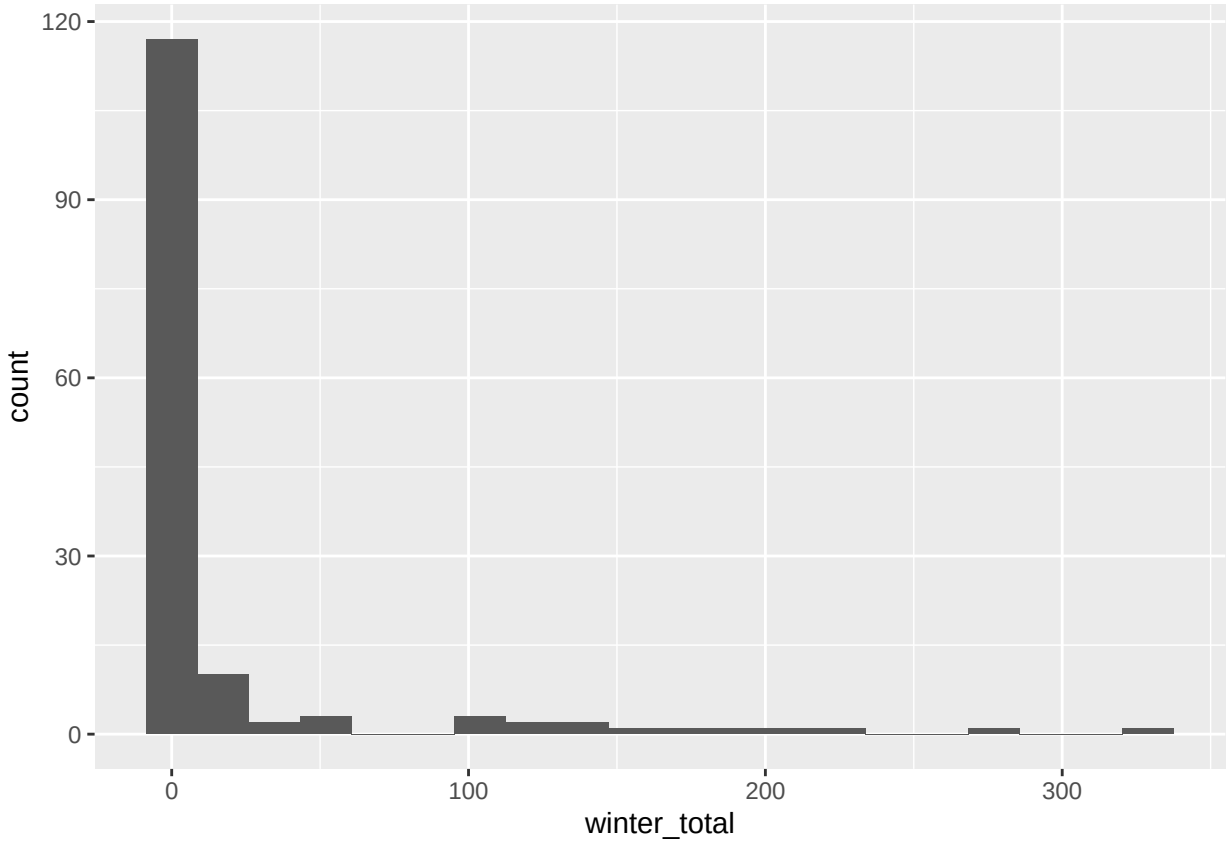




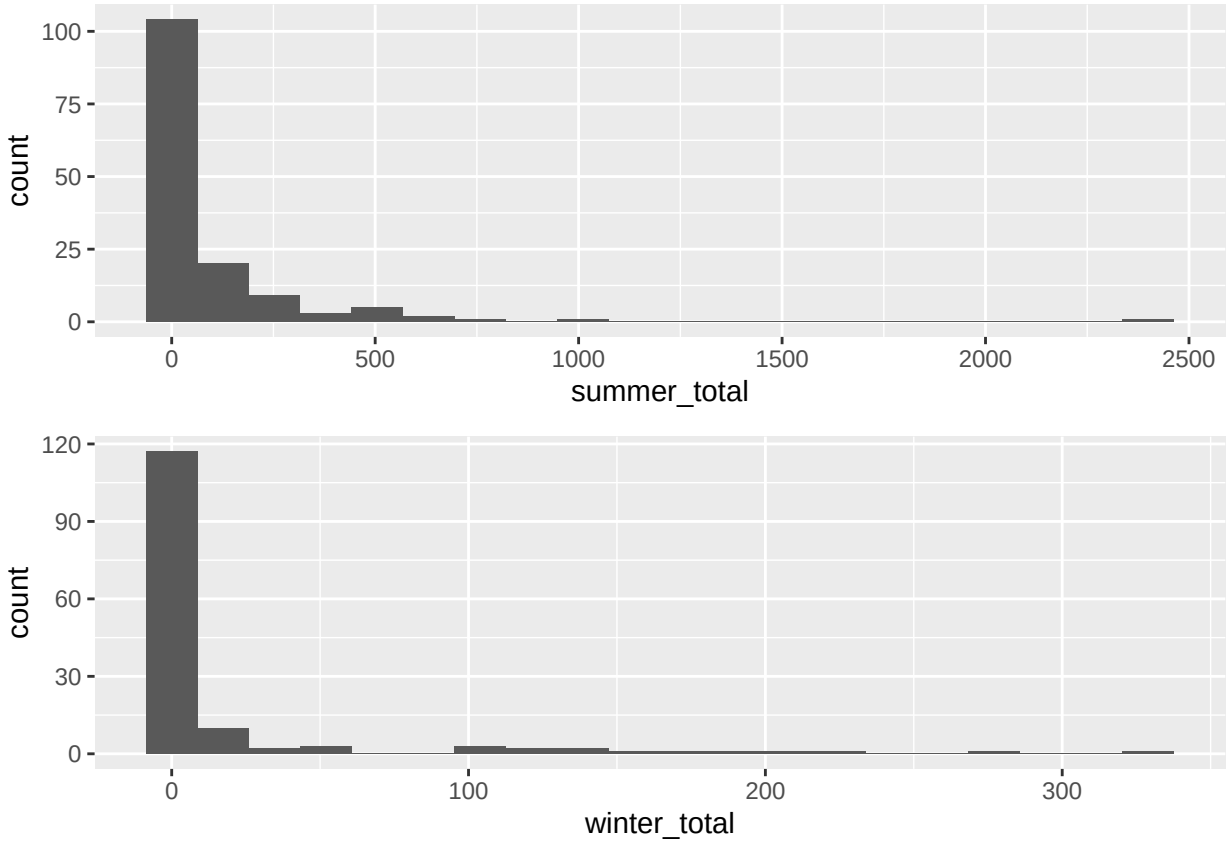

```
# Question 4d

hist_summer_total <- swo %>%
    ggplot(aes(summer_total)) +
    geom_histogram(bins = 20)
hist_summer_total
```



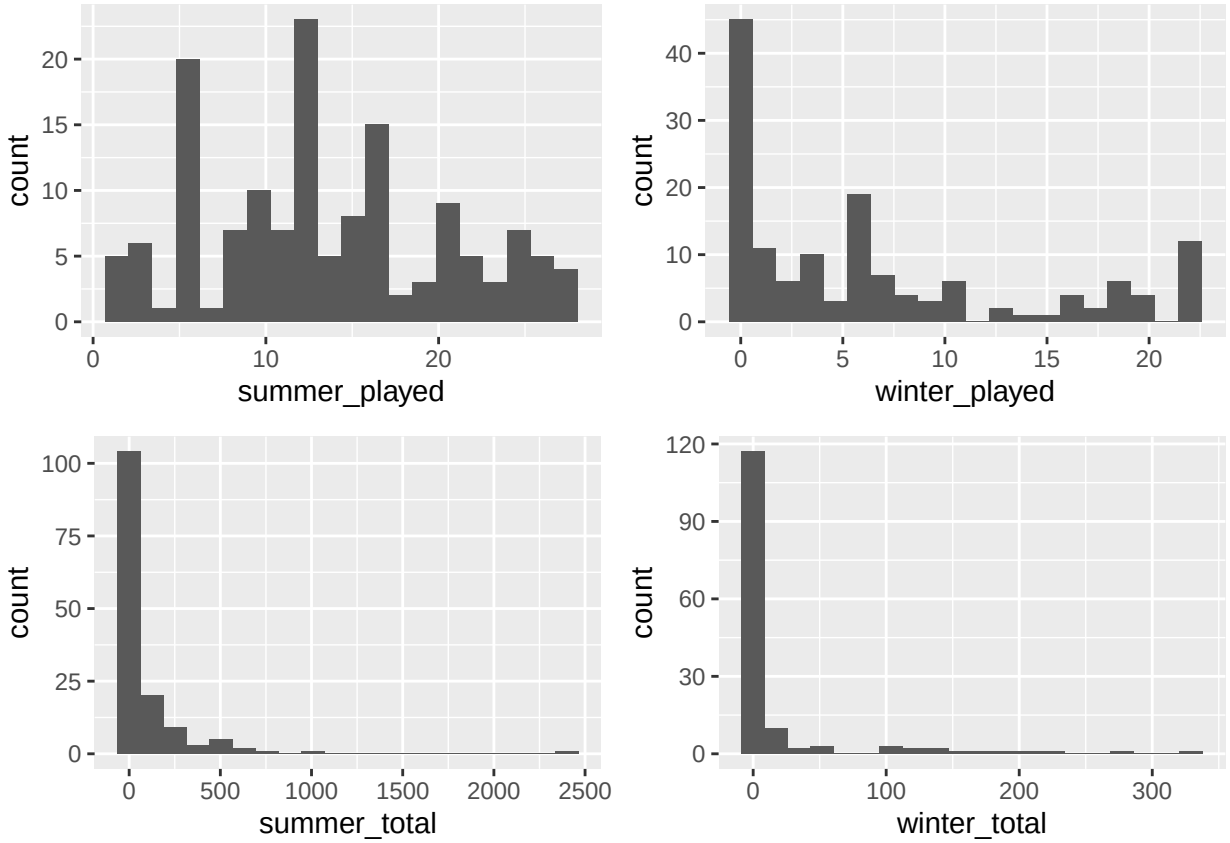

```
hist_winter_total <- swo %>%
    ggplot(aes(winter_total)) +
    geom_histogram(bins = 20)
hist_winter_total
```




```
grid.arrange(hist_summer_total, hist_winter_total)
```




```
grid.arrange(
    hist_summer_played, hist_winter_played,
    hist_summer_total, hist_winter_total
)
```

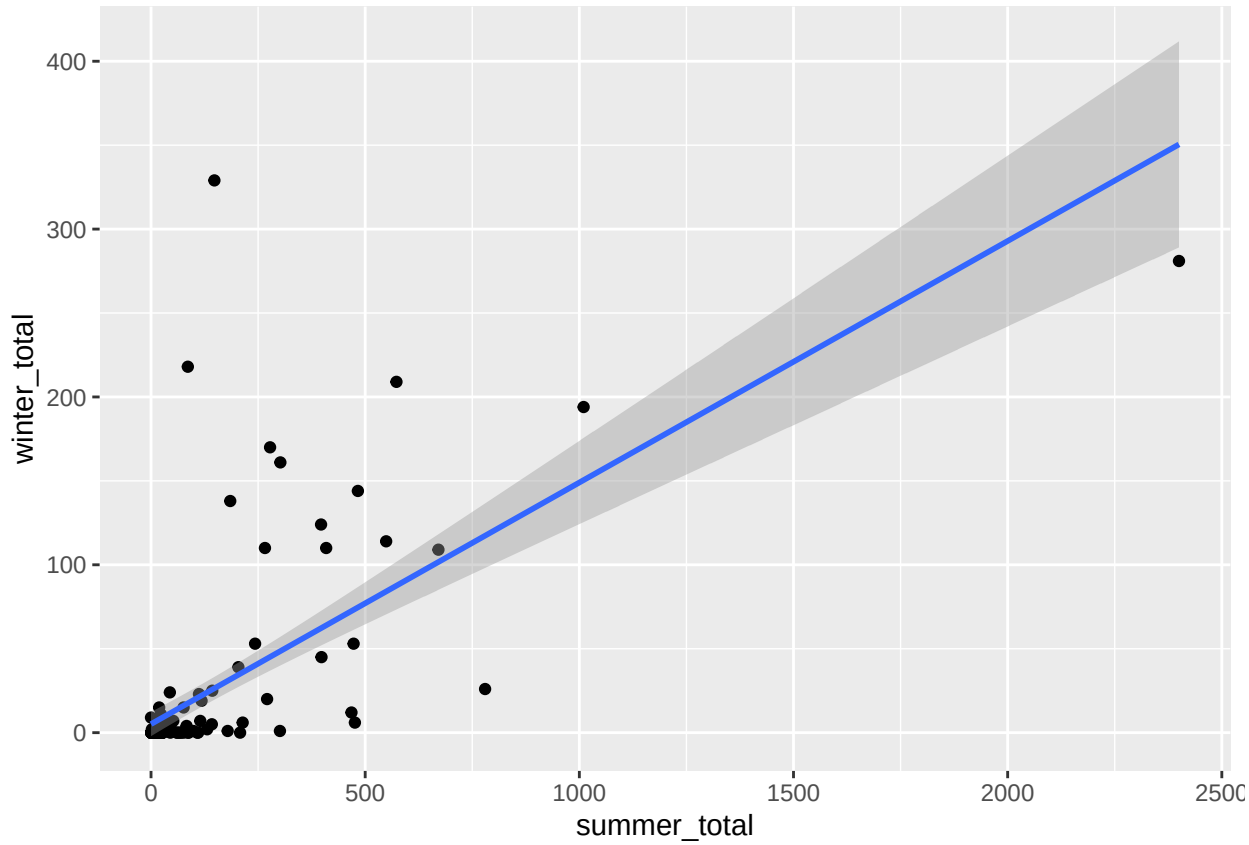



```
# Question 4e
print(
    paste(
        "The correlation between total number of",
        "medals won in summer and in winter is:",
        cor(swo$summer_total, swo$winter_total)
    )
)
```

```
## [1] "The correlation between total number of medals won in summer and in winter is: 0.666063927423379
```

```
swo %>%
    ggplot(aes(summer_total, winter_total)) +
    geom_point() +
    stat_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

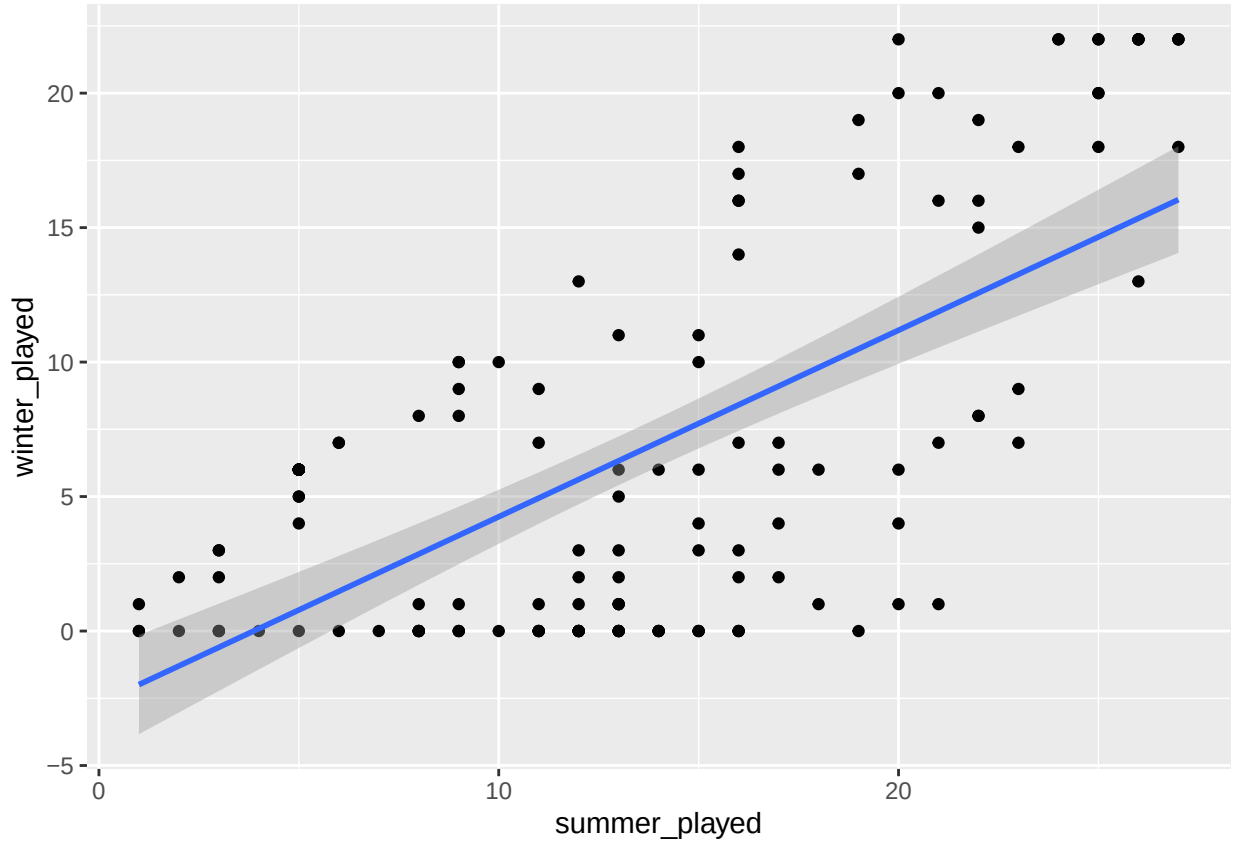



```
# Question 4f
print(
    paste(
        "The correlation between total number of",
        "games played in summer and in winter is:",
        cor(swo$summer_played, swo$winter_played)
    )
)
```

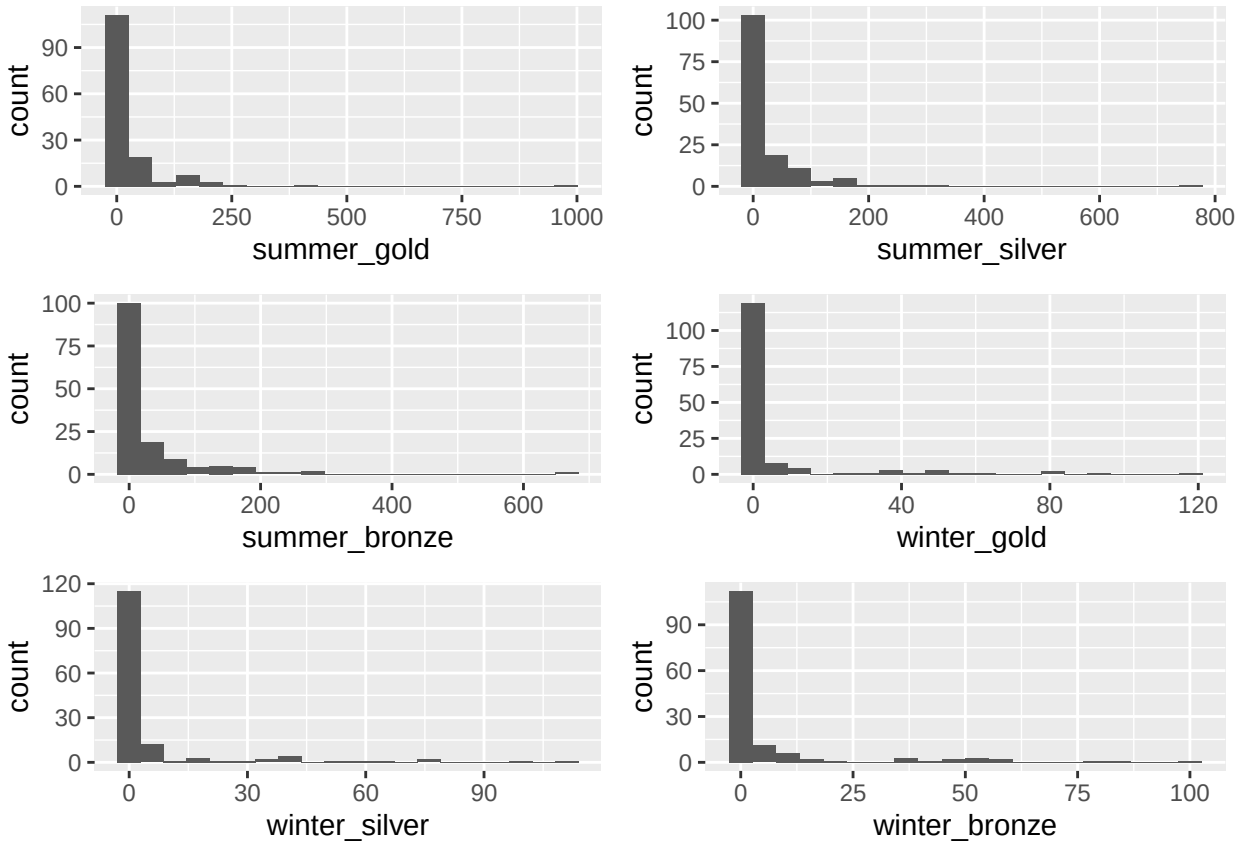
```
## [1] "The correlation between total number of games played in summer and in winter is: 0.661184613384
```

```
swo %>%
    ggplot(aes(summer_played, winter_played)) +
    geom_point() +
    stat_smooth(method = "lm")
```

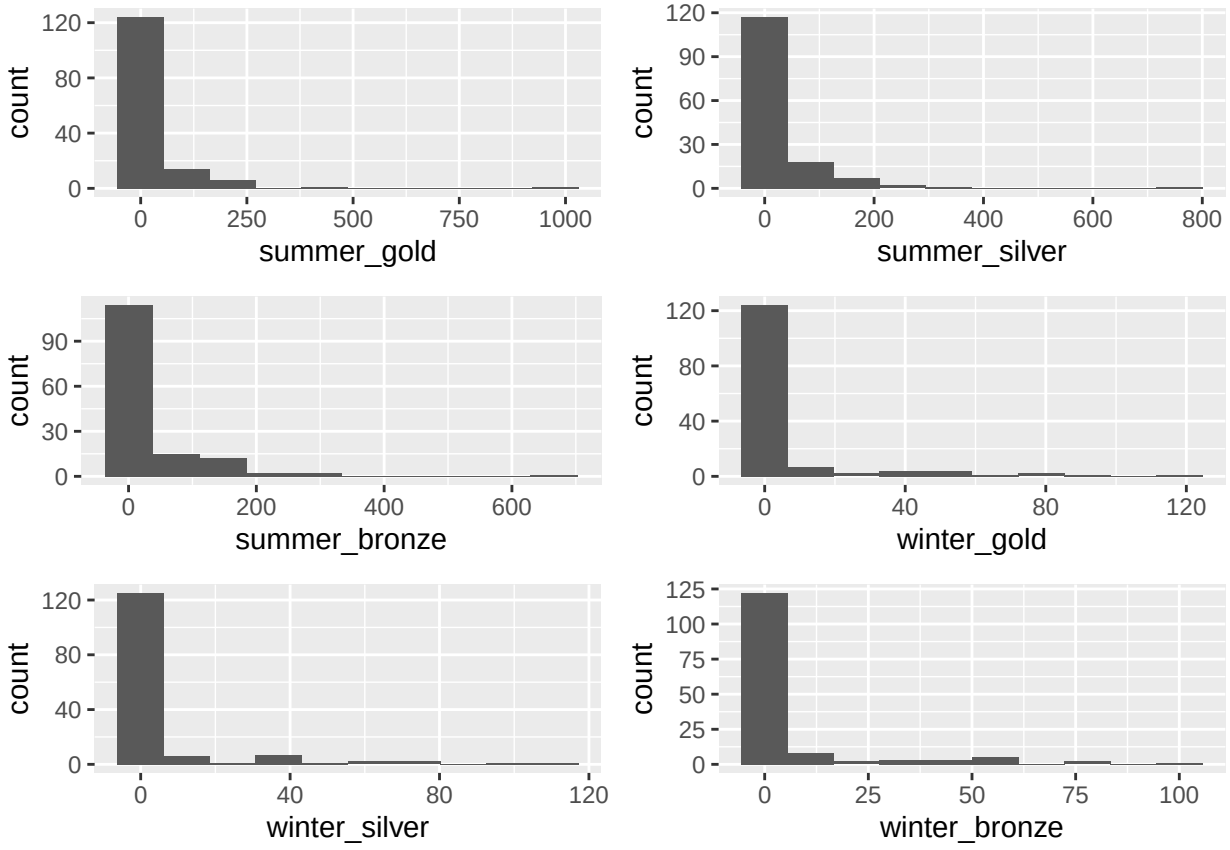
```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Question 4g
hist_summer_gold <- swo %>% ggplot(aes(summer_gold)) +
    geom_histogram(bins = 20)
hist_summer_silver <- swo %>% ggplot(aes(summer_silver)) +
    geom_histogram(bins = 20)
hist_summer_bronze <- swo %>% ggplot(aes(summer_bronze)) +
    geom_histogram(bins = 20)
hist_winter_gold <- swo %>% ggplot(aes(winter_gold)) +
    geom_histogram(bins = 20)
hist_winter_silver <- swo %>% ggplot(aes(winter_silver)) +
    geom_histogram(bins = 20)
hist_winter_bronze <- swo %>% ggplot(aes(winter_bronze)) +
    geom_histogram(bins = 20)

grid.arrange(
    hist_summer_gold,
    hist_summer_silver,
    hist_summer_bronze,
    hist_winter_gold,
    hist_winter_silver,
    hist_winter_bronze
)
```




```
# Question 4h
hist_summer_gold <- swo %>% ggplot(aes(summer_gold)) +
    geom_histogram(bins = 10)
hist_summer_silver <- swo %>% ggplot(aes(summer_silver)) +
    geom_histogram(bins = 10)
hist_summer_bronze <- swo %>% ggplot(aes(summer_bronze)) +
    geom_histogram(bins = 10)
hist_winter_gold <- swo %>% ggplot(aes(winter_gold)) +
    geom_histogram(bins = 10)
hist_winter_silver <- swo %>% ggplot(aes(winter_silver)) +
    geom_histogram(bins = 10)
hist_winter_bronze <- swo %>% ggplot(aes(winter_bronze)) +
    geom_histogram(bins = 10)

grid.arrange(
    hist_summer_gold,
    hist_summer_silver,
    hist_summer_bronze,
    hist_winter_gold,
    hist_winter_silver,
    hist_winter_bronze
)
```




```
# Question 4i
install.packages("ggcorrplot")
```

```
## Installing package into '/home/joris/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(ggcorrplot)

install.packages("GGally")
```

```
## Installing package into '/home/joris/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
install.packages("wordcloud")
```

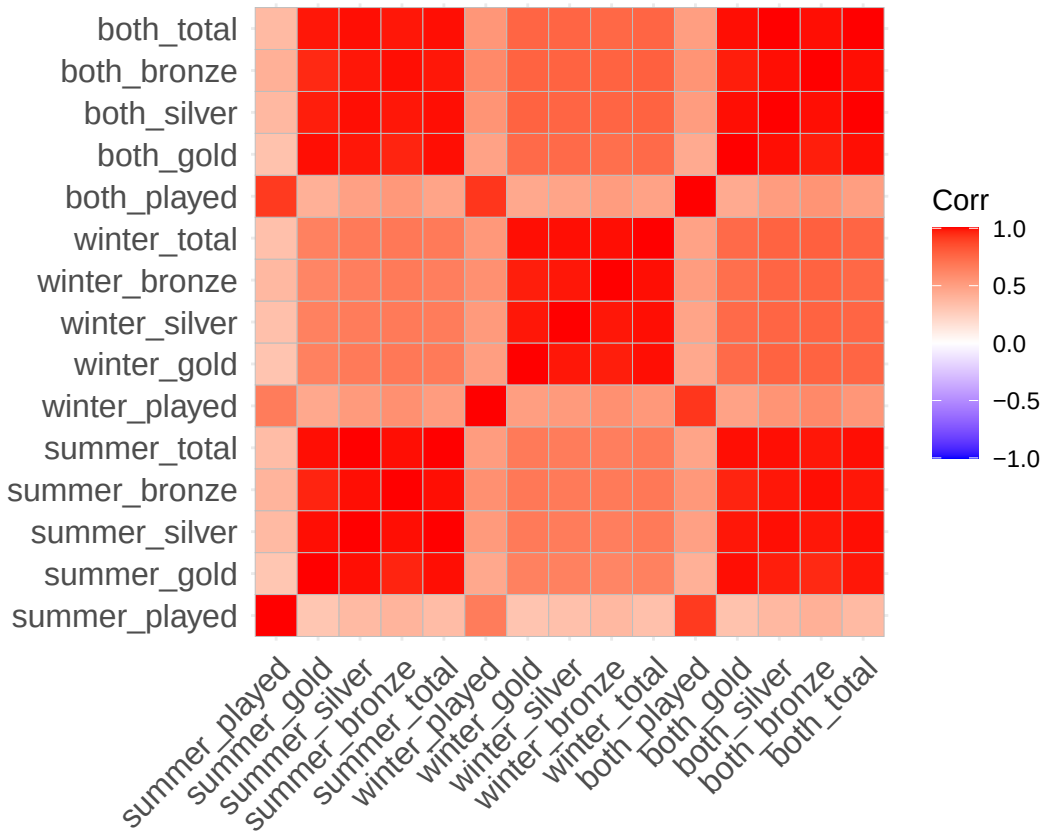
```
## Installing package into '/home/joris/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(wordcloud)
```

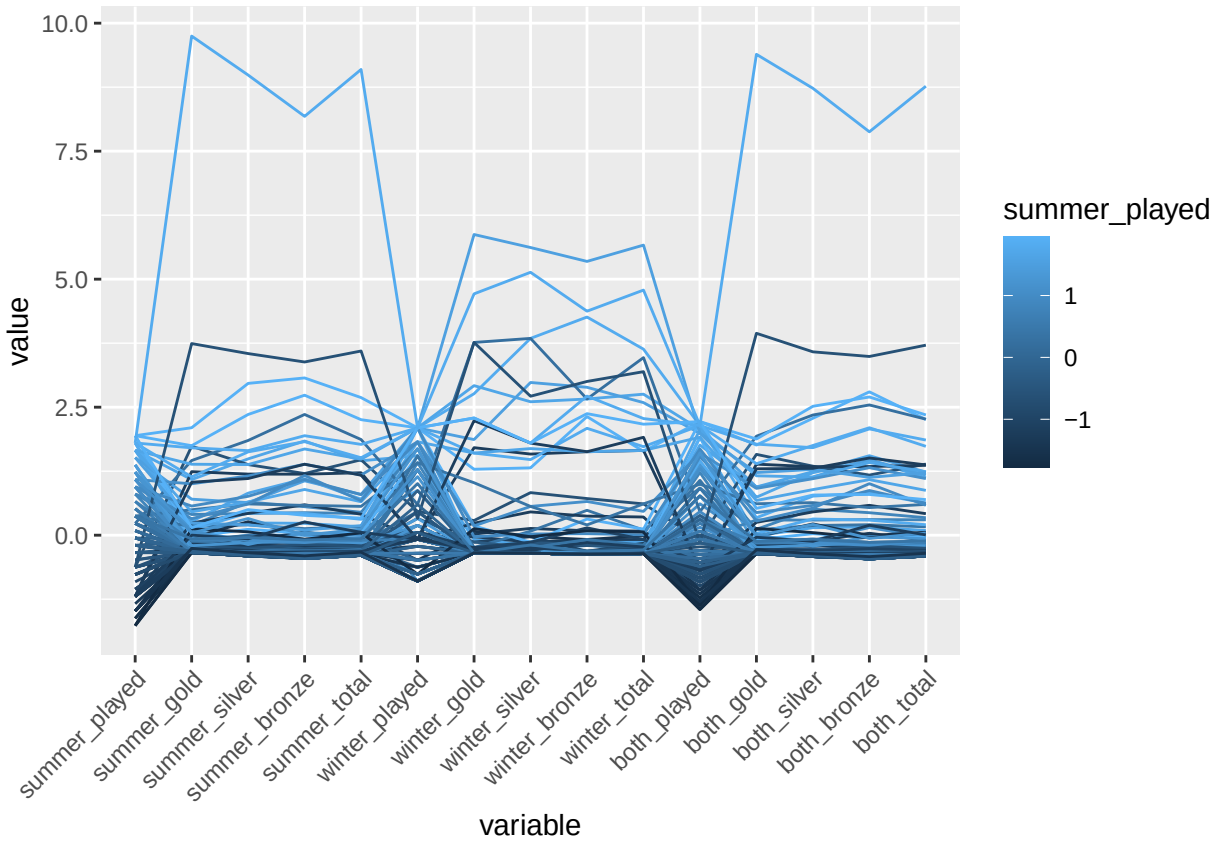
```
## Loading required package: RColorBrewer
```

```
numcol <- swo %>%
    colnames() %>%
```

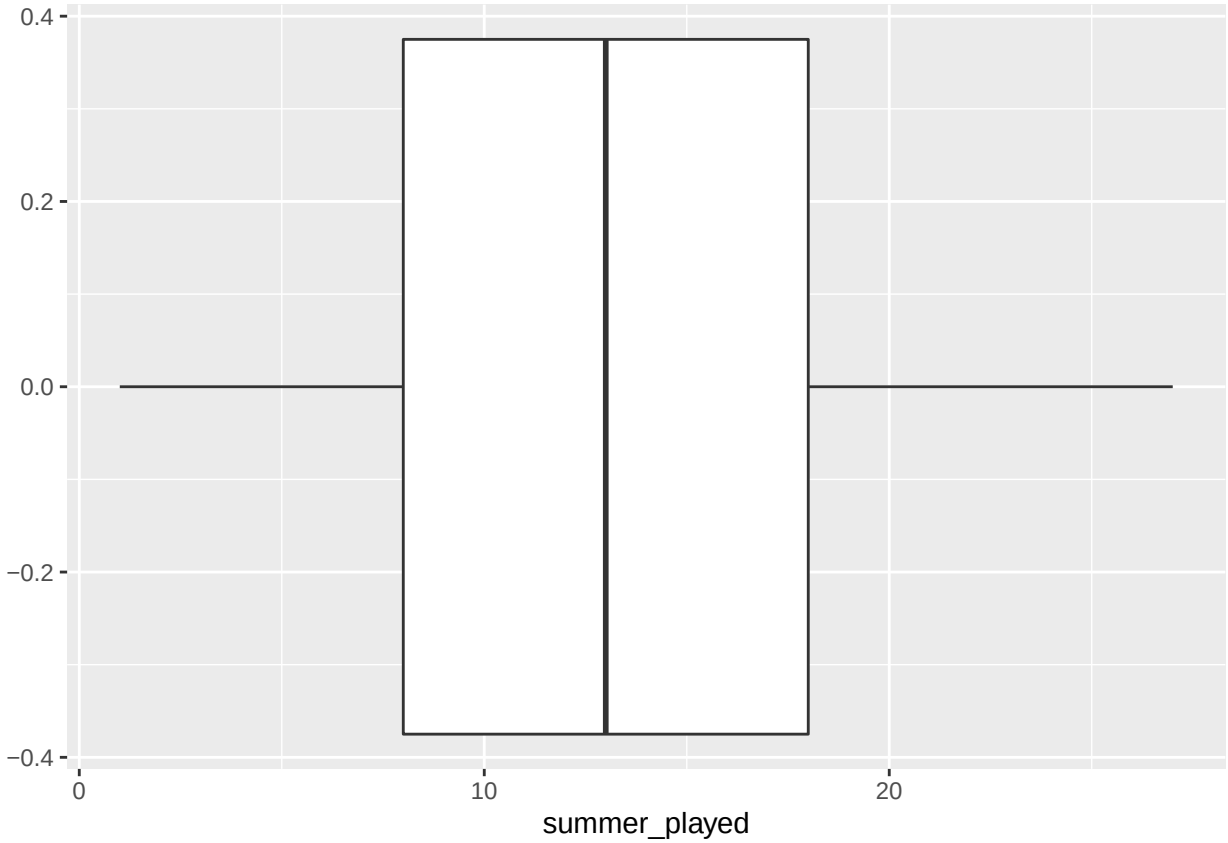
```
    tail(-2)

swo %>%
    select(all_of(numcol)) %>%
    cor() %>%
    ggcorrplot()
```




```
swo %>%
    ggparcoord(columns = 3:17, groupColumn = 3) +
    scale_x_discrete(guide = guide_axis(angle = 45))
```




```
swo %>% ggplot(aes(summer_played)) +
    geom_boxplot()
```




```
wordcloud(
    swo$NOC,
    swo$summer_played,
    max.words = 50,
    rot.per = .35,
    min.freq = 10,
    random.order = FALSE,
    colors = brewer.pal(8, "Dark2")
) # I tried, but the wordcloud doesn't seem to work well
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  France (FRA) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Great Britain (GBR) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Greece (GRE) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Switzerland (SUI) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Denmark (DEN) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Italy (ITA) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
```

```
## 0.35, :  Sweden (SWE) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  United States (USA) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Australia (AUS) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Belgium (BEL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Canada (CAN) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Hungary (HUN) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Netherlands (NED) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Argentina (ARG) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Luxembourg (LUX) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  New Zealand (NZL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Spain (ESP) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Egypt (EGY) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Japan (JPN) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Turkey (TUR) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Ireland (IRL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Philippines (PHI) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Poland (POL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Romania (ROU) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Uruguay (URU) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Bulgaria (BUL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Cuba (CUB) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Iceland (ISL) could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Colombia (COL) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  South Africa (RSA) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Bermuda (BER) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Peru (PER) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Puerto Rico (PUR) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Venezuela (VEN) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Czechoslovakia (TCH) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Guyana (GUY) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Jamaica (JAM) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Lebanon (LIB) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Liechtenstein (LIE) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Pakistan (PAK) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Sri Lanka (SRI) could not be fit on page. It will not be plotted.

## Warning in wordcloud(swo$NOC, swo$summer_played, max.words = 50, rot.per =
## 0.35, :  Trinidad and Tobago (TRI) could not be fit on page. It will not be
## plotted.
```


Portugal (POR)
Chile (CHI)
Norway (NOR)
Austria (AUT)
Finland (FIN)
India (IND)
Mexico (MEX)
Brazil (BRA)