# MATH 3070 Lab Project 6

Pranav Rajan

October 2, 2020

## Contents

*Remember: I expect to see commentary either in the text, in the code with comments created using* **#**, *or (preferably) both!* ***Failing to do so may result in lost points!***

### Problem 1 (Verzani problem 5.6)

*For the* `batting` *(**UsingR**) data set, make parallel boxplots of the batting average (*`H/AB`*) for each team. Which team had the greatest median average? (Use* ***lattice*** *functions for this problem.)*

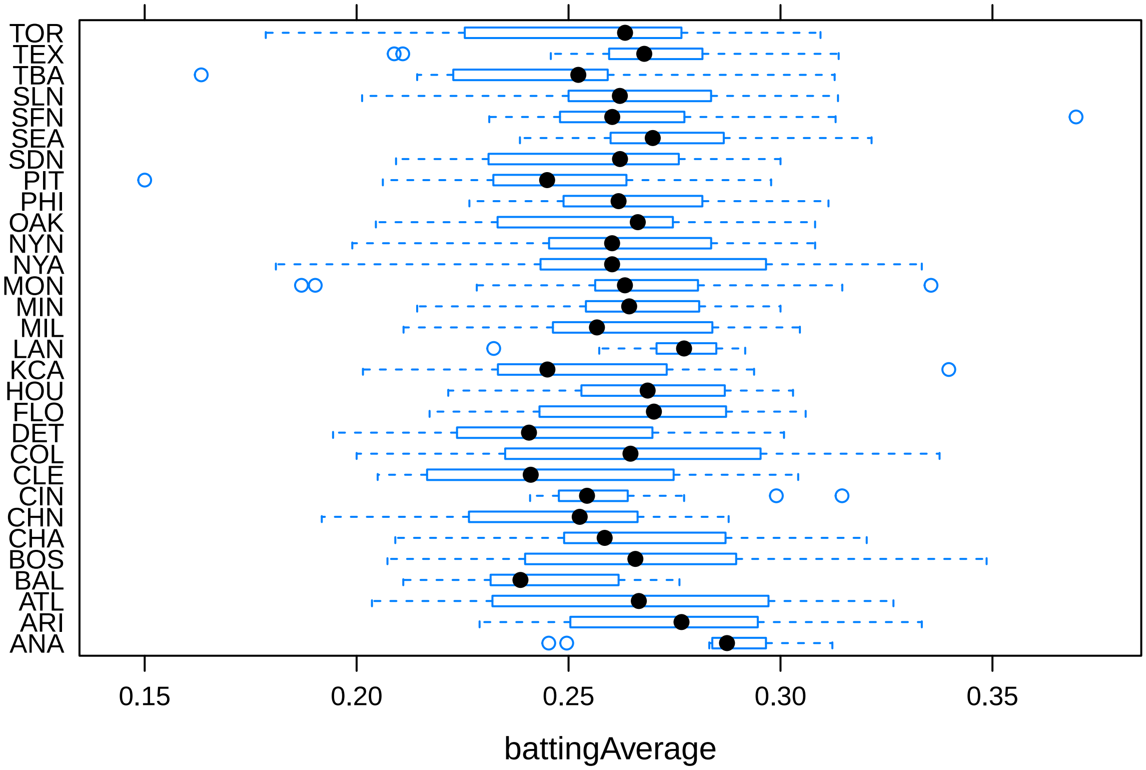
```
# Your code here
require(UsingR)
```

```
## Loading required package: UsingR

## Loading required package: MASS

## Loading required package: HistData

## Loading required package: Hmisc

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
##
## Attaching package: 'UsingR'
```

```
## The following object is masked from 'package:survival':
##
##     cancer
```

```
library(lattice)
battingAverage <- batting$H/batting$AB
bwplot(batting$teamID ~ battingAverage, data=batting)
```



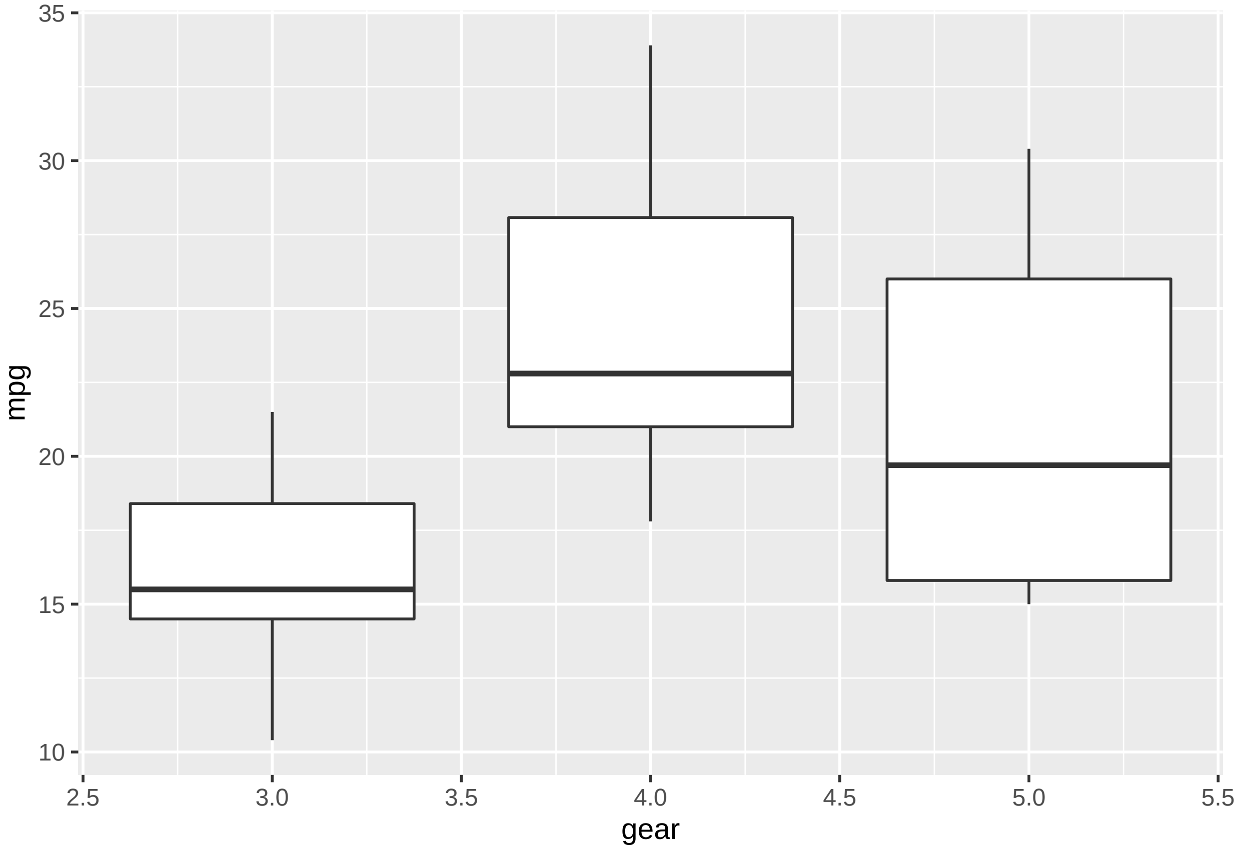
```
# The greatest median average belongs to team TOR
```

## Problem 2 (Verzani problem 5.7)

*For the `mtcars` data set, produce graphics of the following using **ggplot2**:*

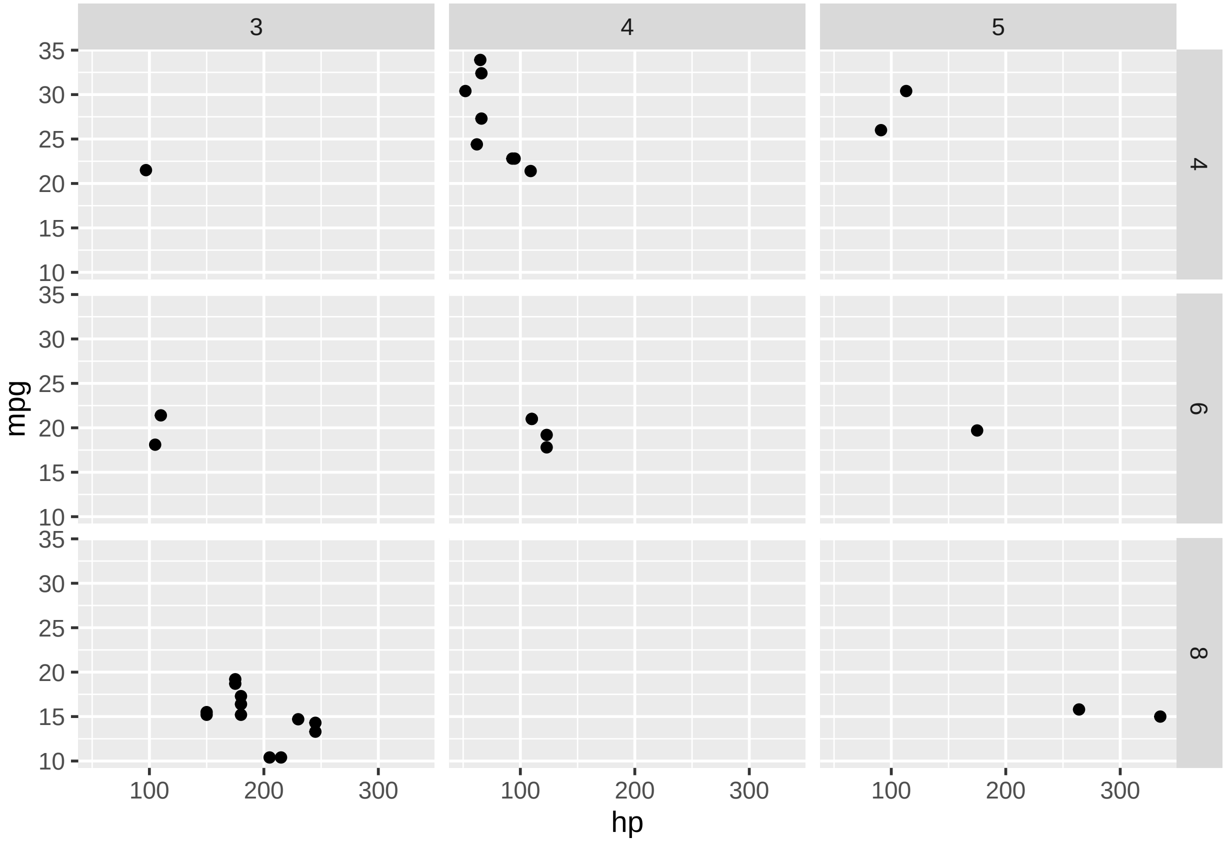
1. *Boxplots for miles per gallon (`mpg`) for groups defined by the number of gears (`gear`).*

```
# Your code here
require(UsingR)
ggplot(mtcars, aes(group=gear, x=gear, y=mpg)) + geom_boxplot()
```



3. *A scatterplot of `mpg` modeled by horsepower (`hp`). Create facets by the number of cylinders (`cyl`) and `gear`.*

```
# Your code here
library(ggplot2)
ggplot(mtcars, aes(x=hp, y=mpg)) + geom_point() + facet_grid(cyl ~ gear)
```



```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

## Problem 3

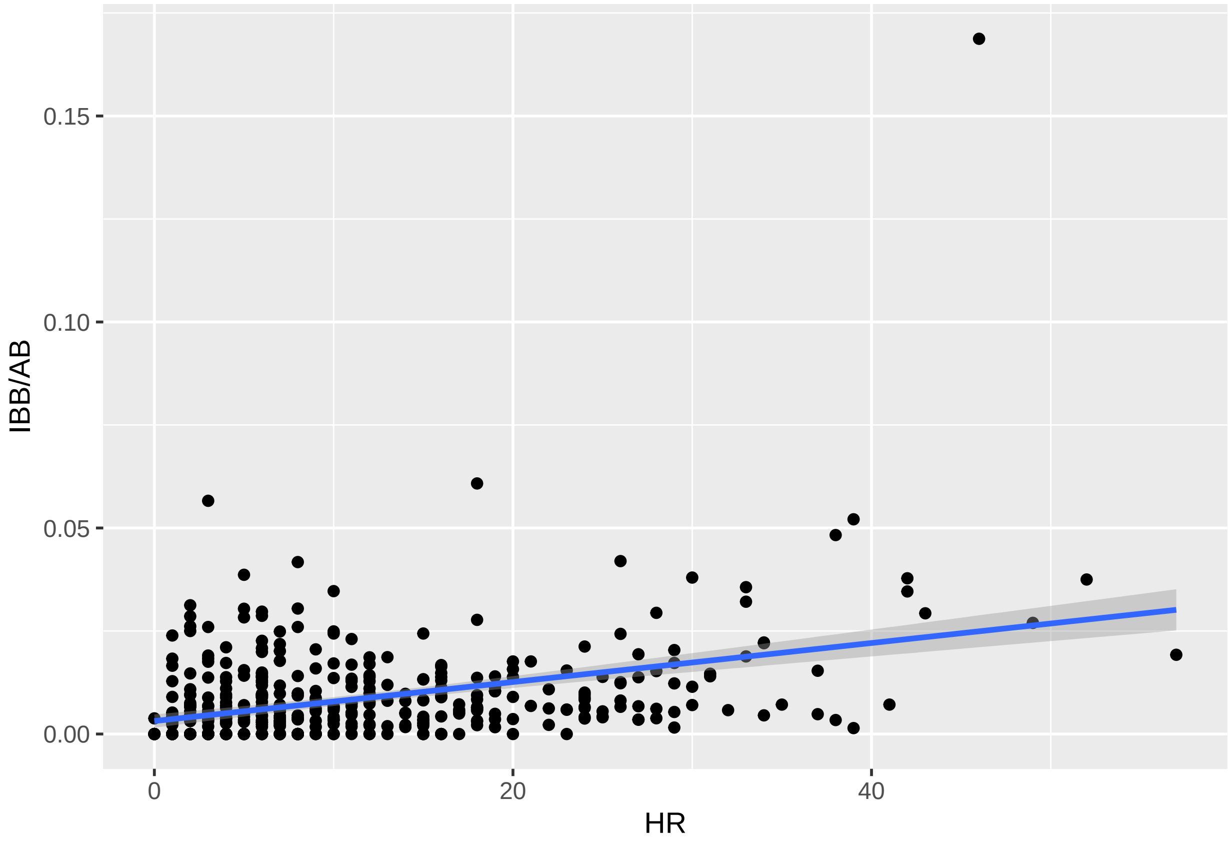
*Using the* `batting` *data set (UsingR), create a visualization that does the following:*

- *Plots the* rate *of intentional walks (that is, the number of intentional walks divided by the number of times a player was at bat; these are the* `IBB` *and* `AB` *variables in the data set, respectively) against the* rate *of home runs (the* `HR` *variable in the data set) as a scatter plot*
- *Draws a trend line for these variables*
- *Identifies and labels the outlier in the data set in these variables (easily spotted once the scatter plot is drawn)*

*(Hint:* `geom`*-type functions can accept data arguments and will use the data set passed rather than the default for the chart. So for the third requirement, consider adding a text layer with* `geom_text(data = ..., aes(...))` *where the argument passed to* `data` *is a subset of the data set consisting of the outlier, and* `aes(...)` *defines how to label that outlier.)*

```
# Your code here
require(UsingR)
intentionalWalkRate <- batting$IBB/batting$AB
ggplot(batting, aes(x=HR, y=IBB/AB)) + geom_point() + geom_smooth(method=lm, fullrange=TRUE)
```

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



## Problem 4

*Reconsider the data set from a previous project containing data about the results of 2012 Olympics. I load the data in for you below:*

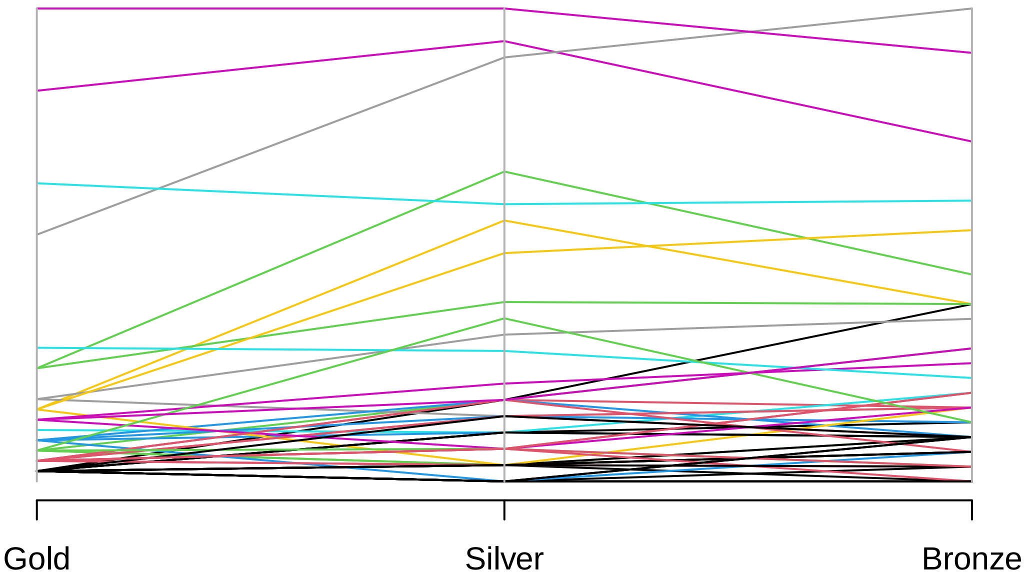
```
olympic2012 <- read.csv("http://introcs.cs.princeton.edu/java/data/olympic-medals2012.csv")
```

*Use any plotting system (base R, **lattice**, **ggplot2**) to create plot involving at least three variables in the `olympic2012` data set, showing a relationship not yet visualized in the lecture, the textbook, or this assignment. Explain the relationship you explored and any interesting findings.* ***Bonus points will be given for plots that I consider exceptionally clean, clear, and insightful, accompanied with good analyses of what you found.***

```
# Your code here
str(olympic2012)
```

```
## ’data.frame’:    204 obs. of  48 variables:
##  $ ISO                                            : chr  "AFG" "ALB" "DZA" "ASM" ...
##  $ GDP.2011                                       : chr  "20,343,461,030.00" "12,959,563,902.00" "188
##  $ pop.2010                                       : chr  "34,385,000" "3,205,000" "35,468,000" "68,42
##  $ Country.name                                   : chr  "Afghanistan" "Albania" "Algeria" "American
##  $ NOC                                            : chr  "AFG" "ALB" "ALG" "ASA" ...
##  $ F.2012                                         : int  1 4 18 1 2 30 2 43 4 1 ...
##  $ M.2012                                         : int  5 7 21 4 4 5 3 99 21 3 ...
##  $ NOC.SIZE                                       : int  6 11 39 5 6 35 5 142 25 4 ...
##  $ NOC.Size.Per.100K.pop                          : num  0.0174 0.3432 0.11 7.3078 7.0701 ...
##  $ Gold                                           : int  0 0 1 0 0 0 0 1 0 0 ...
##  $ Silver                                         : int  0 0 0 0 0 0 0 1 1 0 ...
##  $ Bronze                                         : int  1 0 0 0 0 0 0 2 2 0 ...
##  $ Total                                          : int  1 0 1 0 0 0 0 4 3 0 ...
##  $ Bronze.Per.100K.pop                            : num  0.00291 0 0 0 0 ...
##  $ Silver.Per.100K.pop                            : num  0 0 0 0 0 ...
##  $ Gold.Per.100K.pop                              : num  0 0 0.00282 0 0 ...
##  $ Total.Per.100K.pop                             : num  0.00291 0 0.00282 0 0 ...
##  $ Bronze.Per.1BN.GDP                             : num  0.0492 0 0 0 0 ...
##  $ Silver.Per.1BN.GDP                             : num  0 0 0 0 0 ...
##  $ Gold.Per.1BN.GDP                               : num  0 0 0.0053 0 0 ...
##  $ Total.Per.1BN.GDP                              : num  0.0492 0 0.0053 0 0 ...
##  $ Bronze.Per.Athlete                             : num  0.167 0 0 0 0 ...
##  $ Silver.Per.Athlete                             : num  0 0 0 0 0 ...
##  $ Gold.Per.Athlete                               : num  0 0 0.0256 0 0 ...
##  $ Total.Per.Athlete                              : num  0.1667 0 0.0256 0 0 ...
##  $ Bronze.pop                                     : num  0.4 0 0 0 0 0 0 0.7 9.3 0 ...
##  $ Silver.pop                                     : num  0 0 0 0 0 0 0 0.4 5.3 0 ...
##  $ Gold.pop                                       : num  0 0 0.3 0 0 0 0 0.3 0 0 ...
##  $ Total.pop                                      : num  0.4 0 0.3 0 0 0 0 1.4 14.6 0 ...
##  $ Bronze.GDP                                     : num  5.53 0 0 0 0 ...
##  $ Silver.GDP                                     : num  0 0 0 0 0 ...
##  $ Gold.GDP                                       : num  0 0 0.54 0 0 0 0 0.23 0 0 ...
##  $ Total.GDP                                      : num  5.53 0 0.54 0 0 ...
##  $ Bronze.Athlete                                 : num  17.7 0 0 0 0 ...
##  $ Silver.Athlete                                 : num  0 0 0 0 0 0 0 0.91 5.17 0 ...
##  $ Gold.Athlete                                   : num  0 0 4.23 0 0 0 0 1.16 0 0 ...
##  $ Total.Athlete                                : num  17.72 0 4.23 0 0 ...
##  $ GDP.rank.score                               : num  5.53 0 1.62 0 0 ...
##  $ Population.rank.score                        : num  0.4 0 0.9 0 0 0 0 2.4 19.9 0 ...
##  $ Athlete.rank.score                           : num  17.7 0 12.7 0 0 ...
##  $ Official.medal.ranking                       : int  79 86 58 86 86 86 86 43 53 86 ...
##  $ GDP.rank                                     : int  45 86 68 86 86 86 86 64 8 86 ...
##  $ Pop.rank                                     : int  82 85 78 85 85 85 85 65 29 85 ...
##  $ Team.size.rank                               : int  39 86 58 86 86 86 86 76 37 86 ...
##  $ X                                            : logi  NA NA NA NA NA NA ...
##  $ Total.medal.score..gold.3..silver..2..bronze.1.: int  1 0 3 0 0 0 0 7 4 0 ...
##  $ Model.based.score                            : num  -0.726 -1.174 -5.829 -0.104 -0.413 ...
##  $ Model.based.rank                             : int  107 125 169 56 90 167 67 184 30 82 ...
```

```
library(MASS)
parcoord(olympic2012[c("Gold", "Silver", "Bronze")],
    col = olympic2012$Gold)
```



```
# medals won by country with a parallel coordinates chart. The different columns represent the
# medals one and the different colored lines represent the countries that won.
```