HW4B.R

Frank

Tue Oct 4 23:04:00 2016

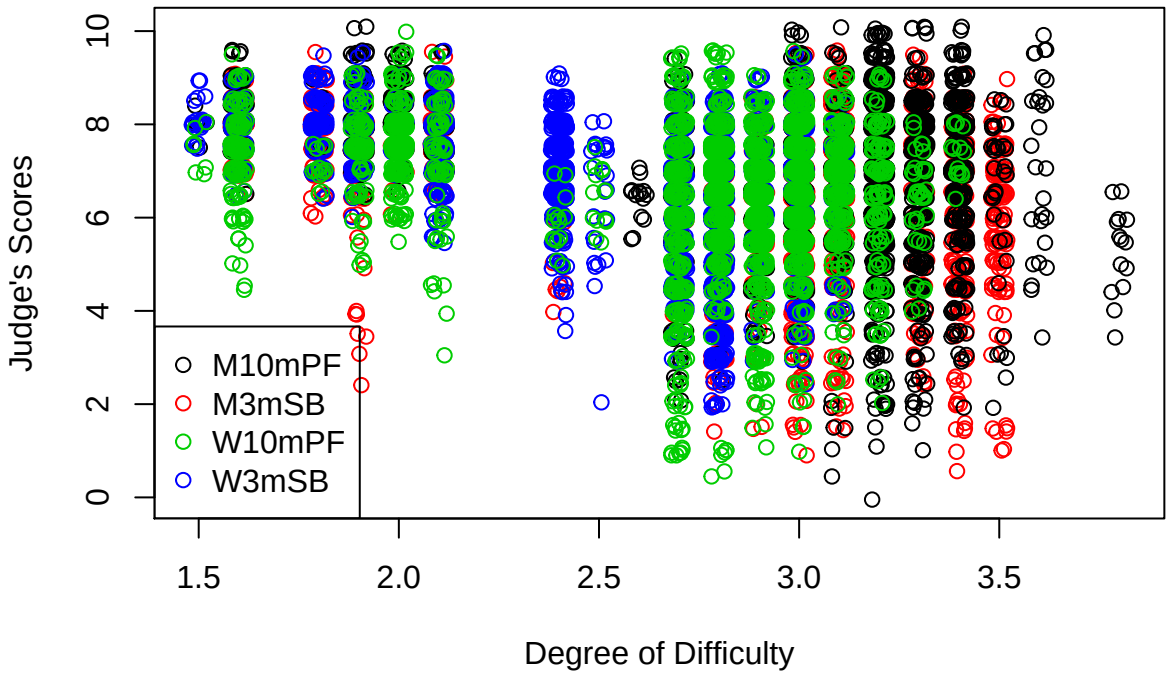
```
#
# Homework 4B          <<<<<<< Dingjue Ji >>>>>>>       LAST INITIAL: J
#
# Due Wednesday 10/5, 1 PM
#
# Please staple this behind Homework 4A.  Here, you don't need to
# delete any of the comments -- they aren't excessive.  But please
# don't include any of your exploratory data analysis code & results
# that don't relate directly to the questions posed.  Thanks!

x <- read.csv("http://www.stat.yale.edu/~jay/diving/Diving2000.csv",
              as.is=TRUE)
dim(x)
```

```
## [1] 10787    10
```

```
names(x)
```

```
##  [1] "Event"      "Round"      "Diver"      "Country"    "Rank"
##  [6] "DiveNo"     "Difficulty" "JScore"     "Judge"      "JCountry"
```

```
# Most of the variables are obvious.  Country is the country of the diver,
# JCountry is the country of the judge.  There are seven judges for each
# dive.  Be careful: JScore has an odd capitalization and this does matter.
# Rank and DiveNo are used only within rounds of events and should be
# ignored (other than perhaps to observe who eventually won medals).

plot(jitter(JScore) ~ jitter(Difficulty), data=x,
     col=factor(x$Event),
     xlab="Degree of Difficulty",
     ylab="Judge's Scores")
legend("bottomleft", legend=levels(factor(x$Event)),
       pch=1, col=1:nlevels(factor(x$Event)))
```



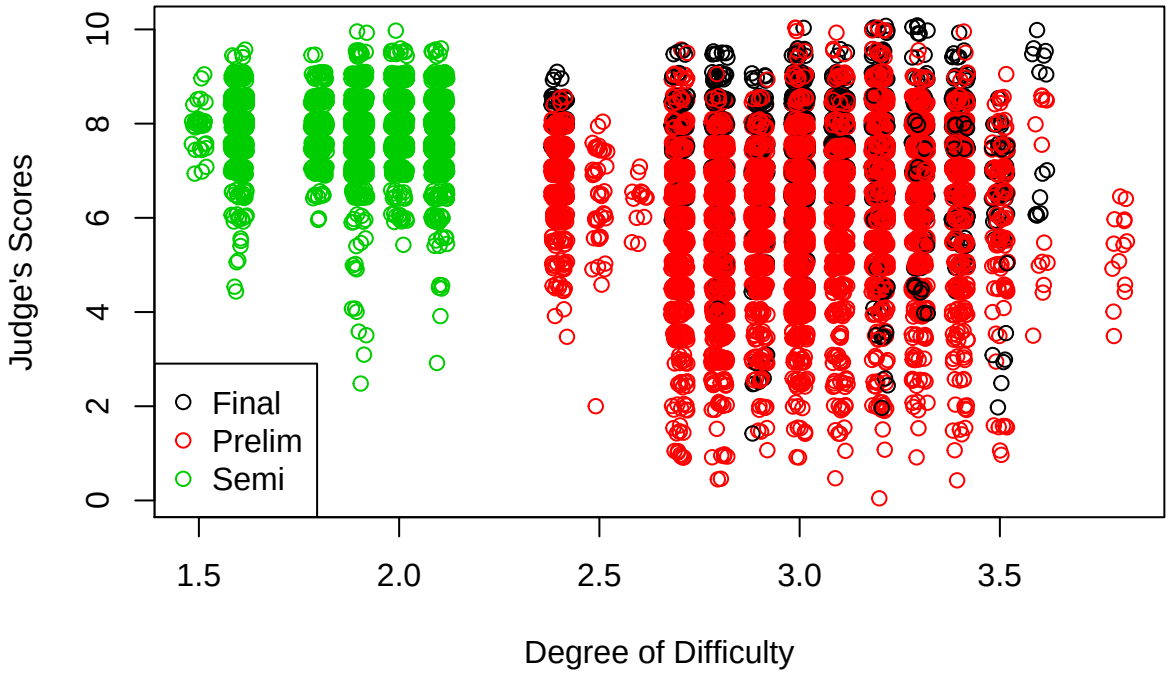

```
# No, the legend isn't beautiful.  But it's fine for basic
# explorations.  It might look just fine on a different
# graphics device, but is kind of bad on my screen at the moment.

# Puzzle: there is an odd left/right division in the plot above.
# I show a few things with graphics which you can observe,
# absorb, and modify for your own use.  There is a lot of overplotting,
# which the jittering partly addresses.  Without use of the jitter()
# the overplotting would have been terrible.  The coloring is a bit of a mess,
# not clearly explaining the left/right groupings (or perhaps it is just
# a gap in the middle).

################################################################################
## Problem 4A.1:
## Using graphical exploration, tables, or other summary statistics,
## try to explain the nature (if any) of the left/right divide
## evident in the plot.  It has nothing to do with the event, I just
## wanted to show you how to add color with an example.

##
## Final code to support your answer here (don't include all
## explorations -- only the essentials).  The code may produce
## a graphic or perhaps graphics, tables, summary statistics, etc...,
## but should related directly to your brief discussion, below.
##

left <- x[x$Difficulty <= 2.1, ]
right <- x[x$Difficulty > 2.1, ]
table(left$Round)
```

```
##
## Semi
## 2303
```

```
table(right$Round)
```

```
##
##  Final Prelim
##   1848   6636
```

```
plot(jitter(JScore) ~ jitter(Difficulty), data=x,
     col=factor(x$Round),
     xlab="Degree of Difficulty",
     ylab="Judge's Scores")
legend("bottomleft", legend=levels(factor(x$Round)),
       pch=1, col=1:nlevels(factor(x$Round)))
```




```
## Tweet out your explanation, or say "I didn't find anything interesting."
## 140 characters or less.  Try it:

# All the divers prefer a low degree (less than 2.2) of difficulty in
# semi-final.
#

################################################################################
## Problem 4A.2:
## Calculate some simple summary statistics: a single line of code for
## each challenge, below.  No discussion other than the last challenge.

## What is the average score in the competition?
mean(x$JScore)
```

```
## [1] 6.832576
```

```
## What is the average score for Chinese divers (a single average)?
mean(x[x$Country == 'CHN', 'JScore'])
```

```
## [1] 8.158986
```

```
## What is the average score for American divers (a single average)?
mean(x[x$Country == 'USA', 'JScore'])
```

```
## [1] 7.477191
```

```
## Using tapply(), a one-line command, calculate the average score
## all at once for all countries separately.  Might require some playing
## around and experimenting, but you can check the answers for CHN and USA
## against what you did above.
tapply(x$JScore, factor(x$Country), mean)
```

```
##      ARG      ARM      AUS      AUT      AZE      BLR      BRA      CAN
## 4.614286 5.238095 7.302885 6.445714 6.226190 6.651786 6.391534 7.440179
##      CHN      COL      CUB      CZE      ESP      FIN      FRA      GBR
## 8.158986 5.903361 6.486711 5.488095 6.243243 6.458333 6.109375 6.363839
##      GEO      GER      GRE      HKG      HUN      INA      ITA      JPN
## 6.000000 7.212798 5.544974 4.666667 6.510823 4.473214 6.811224 7.590909
##      KAZ      KOR      MAS      MEX      PER      PHI      PRK      PUR
## 6.606516 5.844156 6.010204 6.913095 6.017857 5.603896 6.672131 5.831633
##      ROM      RUS      SUI      SWE      THA      TPE      UKR      USA
## 5.662338 7.623894 5.240260 7.647619 5.107143 5.185714 6.824580 7.477191
##      VEN      ZIM
## 5.934783 5.583333
```

```
## Calculate the Semi-Final average score.  Can you explain why it is
## so much higher than the average score in the Preliminary or Final
## rounds of the competition?  The equivalent of a Tweet or two should
## be more than enough:
mean(x[x$Round == 'Semi', 'JScore'])
```

```
## [1] 7.793747
```

```
tapply(x$JS, factor(x$Round), mean)
```

```
##    Final   Prelim     Semi
## 7.474838 6.320148 7.793747
```

```
mean(x[x$Difficulty <= 2.1, 'JScore'])
```

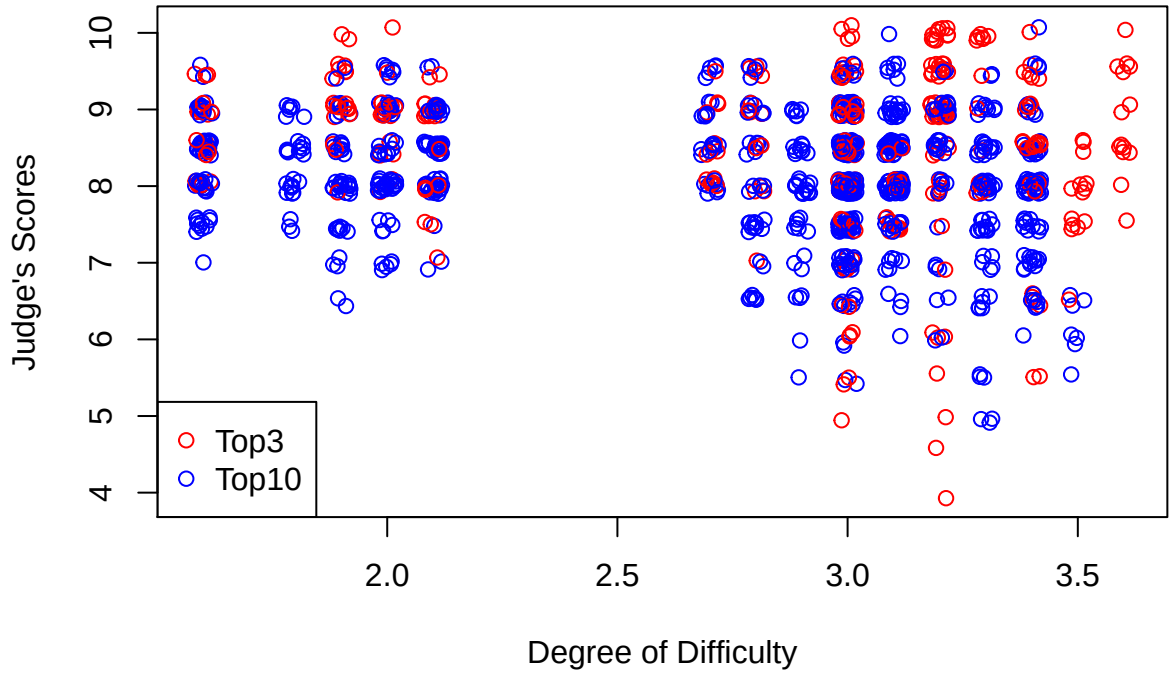
```
## [1] 7.793747
```

```
mean(x[x$Difficulty > 2.1, 'JScore'])
```

```
## [1] 6.571664
```

```
#
# Because during the Semi, divers are more likely to choose low difficulty.
# When the degree of diffculty is low, it will be easier for divers to get a
# a higher score. As a result, the Semi score will be higher than the others.
#
#



################################################################################
## Problem 4A.3:
##
## Open-ended graphical exploration.  Have some fun exploring the data set
## with graphics.  Choose your favorite -- present it well with labels,
```

```
## etc... -- and with the equivalent of a Tweet or two (really, keep it
## short) point out what you find interesting in the plot.  There is no right
## answer, but polish of graphics and code will be noted.  Include only
## your code for the graphic you present.  Use of par(mfrow(...)) is
## allowed as long as the result is clear when printed.

top10 <- sort(tapply(x$JScore, factor(x$Diver), mean), decreasing = TRUE)[1:10]
top10 <- names(top10)
top3 <- top10[1:3]
dat <- x[x$Diver %in% top10, ]
cols <- ifelse(dat$Diver %in% top3, 'red', 'blue')
plot(jitter(JScore) ~ jitter(Difficulty), data=dat,
     col=cols,
     xlab="Degree of Difficulty",
     ylab="Judge's Scores")
legend("bottomleft", legend=c('Top3', 'Top10'),
       pch=1, col=c('red', 'blue'))
```




```
# The Top3 divers are distinguished because they can handle the high difficuty
# better.
```