Warm Up 4

Frankie Lin

2/22/2019

1) Importing Data

```
#loading
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(readr)

# assembling url so it fits on the screen
github <- 'https://raw.githubusercontent.com/ucb-stat133/stat133-hws/'
repo <- 'master/data/nba2018-players.csv'
datafile <- paste0(github, repo)
```

a.

```
cols = c("ccfiiiicdiiiiii")
```

b.

```
dat <- read_csv(datafile, col_names = TRUE,col_types = cols)
summary(dat)
```

```
##     player              team           position     height
##  Length:477         Length:477         C : 97   Min.   :69.00
##  Class :character   Class :character   PF: 98   1st Qu.:77.00
##  Mode  :character   Mode  :character   SG:102   Median :79.00
##                                        PG: 96   Mean   :79.09
##                                        SF: 84   3rd Qu.:82.00
##                                                 Max.   :87.00
##      weight           age          experience       college
##  Min.   :150.0   Min.   :19.00   Min.   : 0.000   Length:477
##  1st Qu.:200.0   1st Qu.:23.00   1st Qu.: 1.000   Class :character
##  Median :220.0   Median :26.00   Median : 4.000   Mode  :character
##  Mean   :219.9   Mean   :26.39   Mean   : 4.662
```

```
##  3rd Qu.:240.0   3rd Qu.:29.00   3rd Qu.: 7.000
##  Max.   :290.0   Max.   :40.00   Max.   :18.000
##      salary            games          minutes         points
##  Min.   :    5145   Min.   : 1.00   Min.   :   1   Min.   :   0.0
##  1st Qu.: 1050961   1st Qu.:25.00   1st Qu.: 381   1st Qu.: 124.0
##  Median : 3000000   Median :60.00   Median :1123   Median : 403.0
##  Mean   : 5804697   Mean   :50.71   Mean   :1164   Mean   : 510.3
##  3rd Qu.: 8269663   3rd Qu.:74.00   3rd Qu.:1843   3rd Qu.: 756.0
##  Max.   :30963450   Max.   :82.00   Max.   :3048   Max.   :2558.0
##     points3         points2         points1
##  Min.   :  0.0   Min.   :  0.0   Min.   :  0.00
##  1st Qu.:  2.0   1st Qu.: 30.0   1st Qu.: 15.00
##  Median : 26.0   Median :100.0   Median : 50.00
##  Mean   : 46.4   Mean   :142.3   Mean   : 86.49
##  3rd Qu.: 73.0   3rd Qu.:208.0   3rd Qu.:116.00
##  Max.   :324.0   Max.   :730.0   Max.   :746.00
```

c.

```
class(dat)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

2) Technical Questions about "readr"

a.

After doing a little googling and self research, we see that a tibble is a more modern reimagining of data.frame. They are basically dataframes but they tweak a few of the functions in order to to basically be a bit more modern. Some distinct differences is that the tibble does not change the type of inputs and can have column titles that are unacceptable for standard base R dataframes. However, the two main differences come from the fact that:

-1. A tibble prints only the first 10 rows of the dataframe and that all the columns fit one screen when called upon with additional options to adjust what and how things are printed

-2. When you subset using a tibble, it will only subset the data based off full matching rather than partial matching. Thus they are simply a more strict function and will give you a warning if something does not exist.

b.

You can indeed only import a few columns with by the internal specification of col_types. In this case we would set col_types = c("cc_____d__d___"). This would allow us to omit the unwanted columns.

```
read_csv(datafile, col_names = TRUE,col_types = c("cc______d__d___"))
```

```
## # A tibble: 477 x 4
##    player            team    salary points
##    <chr>             <chr>    <dbl>  <dbl>
##  1 Al Horford        BOS   26540100    952
##  2 Amir Johnson      BOS   12000000    520
##  3 Avery Bradley     BOS    8269663    894
```

```
##  4 Demetrius Jackson BOS    1450000     10
##  5 Gerald Green      BOS    1410598    262
##  6 Isaiah Thomas     BOS    6587132   2199
##  7 Jae Crowder       BOS    6286408    999
##  8 James Young       BOS    1825200     68
##  9 Jaylen Brown      BOS    4743000    515
## 10 Jonas Jerebko     BOS    5000000    299
## # ... with 467 more rows
```

c.

- header = col_names
- col.names = col_names
- na.strings = na
- colClasses = col_classes

3) Salaries by Team

a.

```
team_salaries <- arrange(
  summarise(
  group_by(dat, team),
  total_salary = sum(salary / 1000000),
  mean_salary = mean(salary / 1000000),
  median_salary = median(salary / 1000000)
  ),
  desc(total_salary)
  )
```

b.

```
as.data.frame(team_salaries)
```

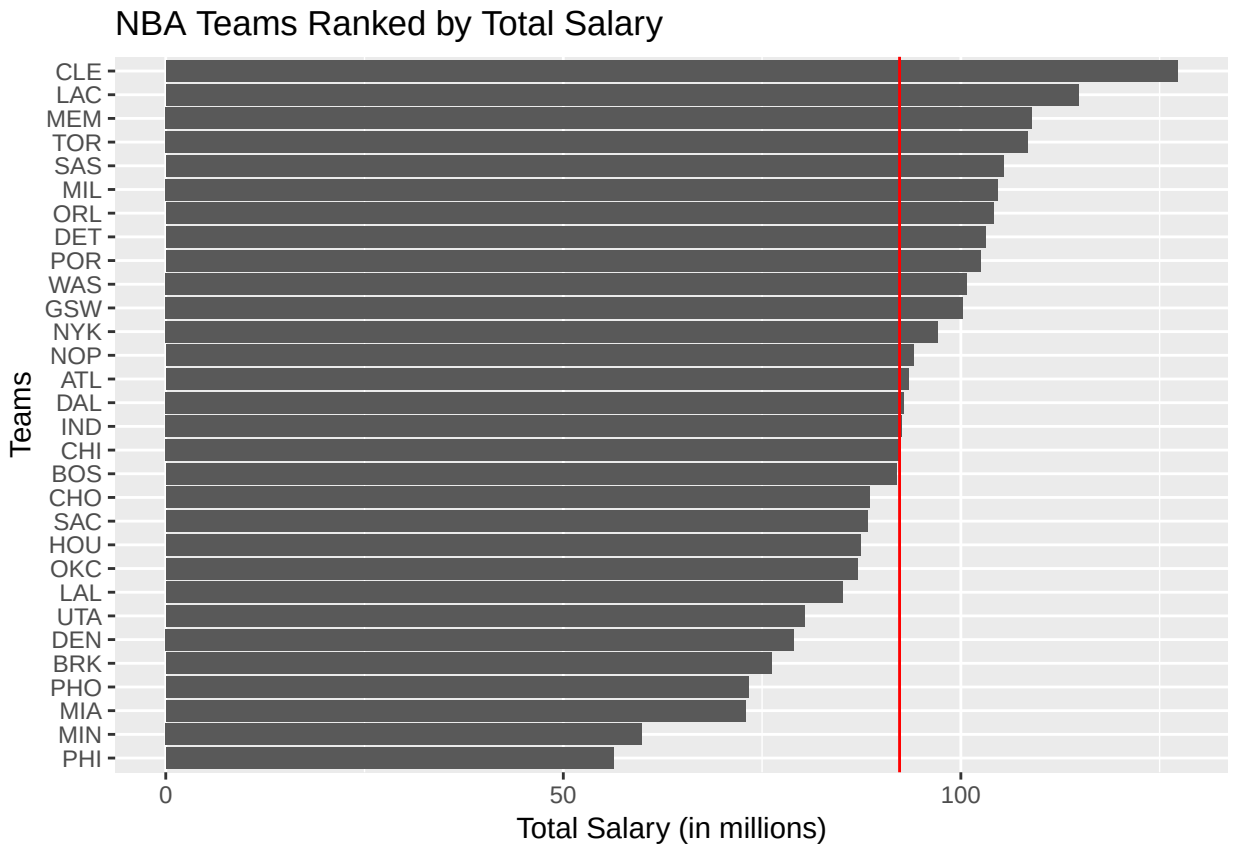
```
##    team total_salary mean_salary median_salary
## 1   CLE    127.25458    7.069699      2.025829
## 2   LAC    114.77662    7.651775      3.500000
## 3   MEM    108.94584    6.809115      3.115470
## 4   TOR    108.45847    7.230565      5.300000
## 5   SAS    105.39553    6.587221      2.224830
## 6   MIL    104.64657    5.507714      2.568600
## 7   ORL    104.11034    5.783908      4.130580
## 8   DET    103.07449    6.871632      4.625000
## 9   POR    102.48876    7.883751      6.666667
## 10  WAS    100.78591    6.719061      3.730653
## 11  GSW    100.24256    6.265160      1.551659
## 12  NYK     97.10692    6.473794      2.898000
## 13  NOP     94.03547    5.877217      2.989125
## 14  ATL     93.40559    5.494447      2.500000
## 15  DAL     92.82830    5.157128      0.945166
```

```
## 16  IND     92.62084    5.788802      4.000000
## 17  CHI     92.50189    5.781368      2.102340
## 18  BOS     91.91509    6.127673      4.743000
## 19  CHO     88.50477    5.531548      4.024157
## 20  SAC     88.27720    5.517325      4.604441
## 21  HOU     87.39233    6.242309      2.309280
## 22  OKC     86.98136    5.798758      3.140517
## 23  LAL     85.12544    6.080389      5.307240
## 24  UTA     80.32319    5.354880      2.433334
## 25  DEN     79.02822    4.648719      3.241800
## 26  BRK     76.21567    4.011351      1.914544
## 27  PHO     73.28258    4.310740      2.223600
## 28  MIA     72.94438    5.210313      3.449000
## 29  MIN     59.87827    4.277020      3.650000
## 30  PHI     56.29336    3.311374      1.514160
```

c.

```
library(ggthemes)

ggplot(team_salaries, aes(x=reorder(team, total_salary), y=total_salary)) +
  geom_bar(stat='identity') +
  geom_hline(yintercept = mean(team_salaries$total_salary), color = "red") +
  coord_flip() +
  ggtitle("NBA Teams Ranked by Total Salary") +
  ylab("Total Salary (in millions)") +
  xlab("Teams")
```



4) Points by Team

a.

```
team_points <-
  arrange(
  summarise(
  group_by(dat, team),
  total_points = sum(points),
  mean_points = mean(points),
  median_points = median(points)
  ),
  desc(total_points)
  )
```

b.

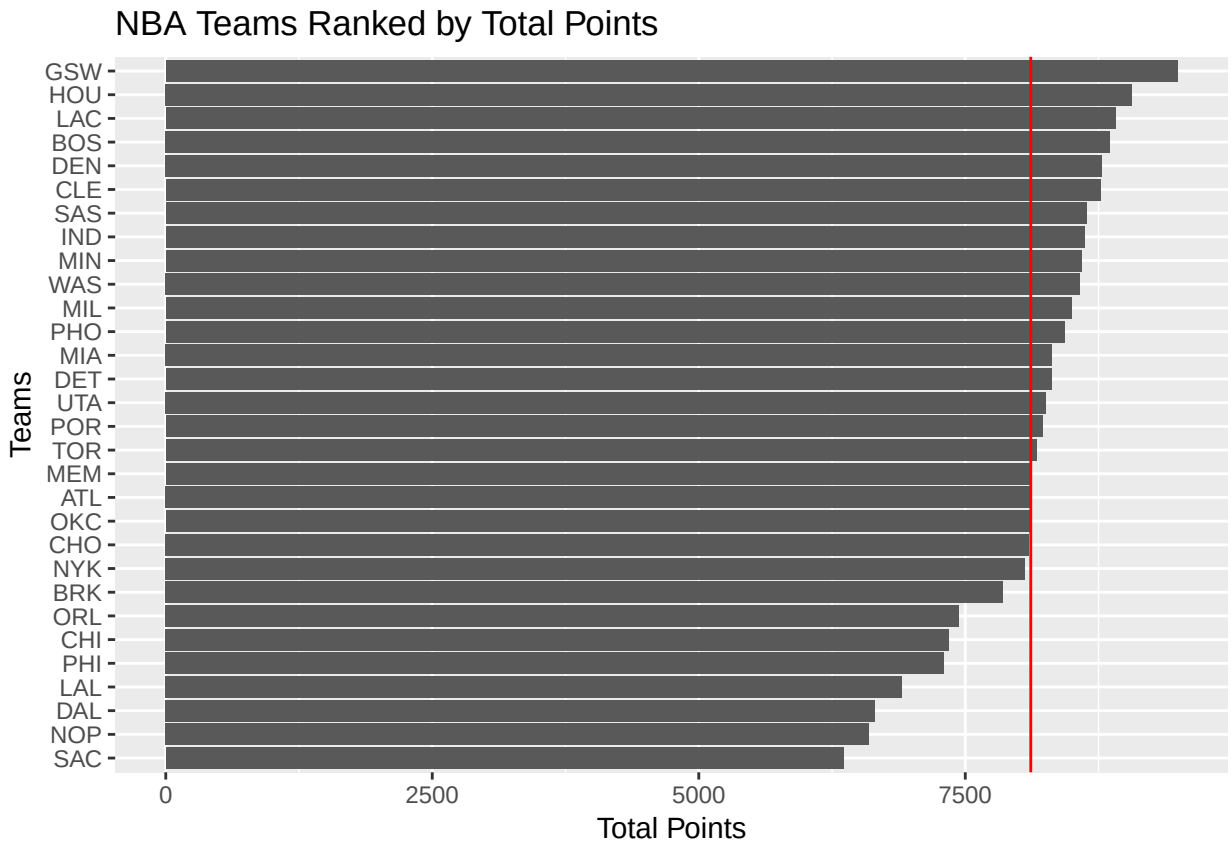
```
as.data.frame(team_points)
```

```
##    team total_points mean_points median_points
## 1   GSW         9491    593.1875         407.5
## 2   HOU         9065    647.5000         568.0
## 3   LAC         8911    594.0667         538.0
```

```
## 4   BOS         8857    590.4667         515.0
## 5   DEN         8783    516.6471         587.0
## 6   CLE         8770    487.2222         265.0
## 7   SAS         8637    539.8125         490.0
## 8   IND         8618    538.6250         370.5
## 9   MIN         8591    613.6429         348.0
## 10  WAS         8574    571.6000         330.0
## 11  MIL         8497    447.2105         392.0
## 12  PHO         8430    495.8824         419.0
## 13  MIA         8312    593.7143         518.0
## 14  DET         8309    553.9333         365.0
## 15  UTA         8258    550.5333         440.0
## 16  POR         8223    632.5385         401.0
## 17  TOR         8166    544.4000         327.0
## 18  MEM         8112    507.0000         427.0
## 19  ATL         8107    476.8824         335.0
## 20  OKC         8104    540.2667         406.0
## 21  CHO         8099    506.1875         457.5
## 22  NYK         8060    537.3333         425.0
## 23  BRK         7855    413.4211         428.0
## 24  ORL         7442    413.4444         308.0
## 25  CHI         7349    459.3125         306.5
## 26  PHI         7299    429.3529         530.0
## 27  LAL         6905    493.2143         392.0
## 28  DAL         6651    369.5000         200.5
## 29  NOP         6597    412.3125         237.0
## 30  SAC         6360    397.5000         465.5
```

c.

```
ggplot(team_points, aes(x=reorder(team, total_points), y=total_points)) +
  geom_bar(stat='identity') +
  geom_hline(yintercept = mean(team_points$total_points), color = "red") +
  coord_flip() +
  ggtitle("NBA Teams Ranked by Total Points") +
  ylab("Total Points") +
  xlab("Teams")
```



5) Cost of Scored Points

a.

```
points_salary <- left_join(team_points, team_salaries, by = "team")
```

b.

```
summary(points_salary)
```

```
##      team            total_points   mean_points    median_points
##  Length:30          Min.   :6360   Min.   :369.5   Min.   :200.5
##  Class :character   1st Qu.:7906   1st Qu.:463.7   1st Qu.:338.2
##  Mode  :character   Median :8240   Median :527.0   Median :406.8
##                     Mean   :8114   Mean   :515.6   Mean   :406.6
##                     3rd Qu.:8611   3rd Qu.:567.2   3rd Qu.:463.5
##                     Max.   :9491   Max.   :647.5   Max.   :587.0
##   total_salary     mean_salary    median_salary
##  Min.   : 56.29   Min.   :3.311   Min.   :0.9452
##  1st Qu.: 85.59   1st Qu.:5.390   1st Qu.:2.2459
##  Median : 92.72   Median :5.794   Median :3.1280
##  Mean   : 92.29   Mean   :5.846   Mean   :3.2476
##  3rd Qu.:102.93   3rd Qu.:6.559   3rd Qu.:4.0181
```

```
##  Max.   :127.25   Max.   :7.884   Max.   :6.6667
```

c.

```
points_salary <- mutate(points_salary, cost_point = (total_salary * 1000000)/total_points)
```

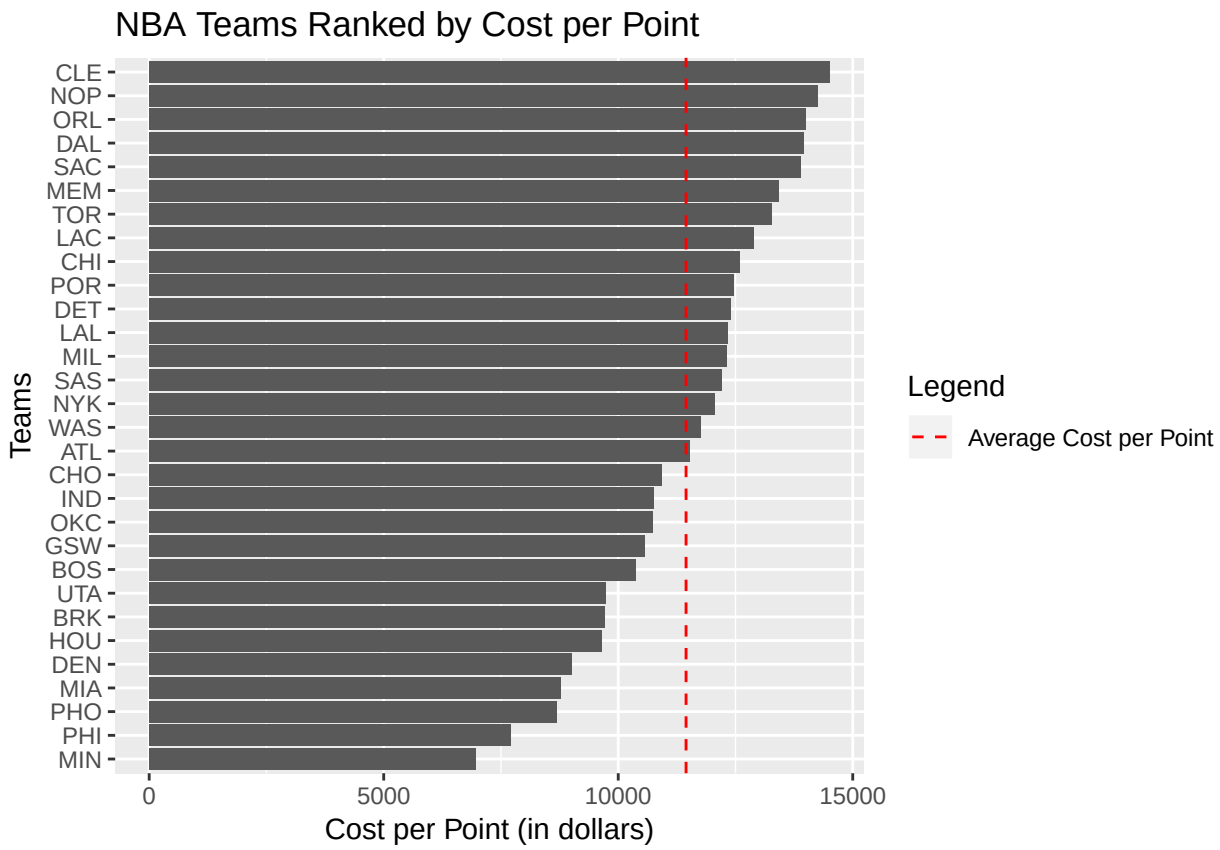
d.

```
summary(points_salary$cost_point)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6970    9889   11901   11446   12807   14510
```

e.

```
ggplot(points_salary, aes(x=reorder(team, cost_point), y=cost_point)) +
  geom_bar(stat='identity') +
  geom_hline(aes(yintercept = mean(points_salary$cost_point), linetype = "Average Cost per Point"), col
  coord_flip() +
  ggtitle("NBA Teams Ranked by Cost per Point") +
  ylab("Cost per Point (in dollars)") +
  xlab("Teams") +
  scale_linetype_manual(name = "Legend", values = c(2), guide = guide_legend(override.aes = list(color =
```



f.

```
qf <- mutate(points_salary, cost_point_quartile = factor(ntile(cost_point, 4)))
levels(qf$cost_point_quartile) <- c("First Quantile", "Second Quantile", "Third Quantile", "Fourth Quant

ggplot(qf, aes(x=mean_points, y=mean_salary, col = cost_point_quartile)) +
  geom_point() +
  ggtitle("NBA Team Mean Cost versus Mean Points Scatter Plot") +
  ylab("Mean Cost (in millions)") +
  xlab("Mean Points Scored") +
  labs(color='Cost Points Quantile')
```