Data Acquisition, Preparation and EDA

Contents

Objectives

Data Science is a field of science connecting statistics, computer science, and domain knowledge. We would like to discover the pattern of differences and changes, as well as the reasons behind the scene. For any well-designed study, we need to first layout the goal of the study. Using domain knowledge we may list possible factors related to the study, i.e., we need to first design what information may help us to achieve the goal. Taking feasibility and cost into account, we will come up with a list of variables and then gather data (from experiments, surveys, or other studies). On the other hand we may want to learn important insights from existing data. Both the quantity and quality of data determine the success of the study. Once we have the data, we proceed to extract useful information. To use the data correctly and efficiently we must understand the data first. In this lecture, we go through some basic data acquisition/preparation and exploratory data analysis to understand the nature of the data, and to explore plausible relationships among the variables. We defer the formal modeling later.

Data mining tools have been expanding dramatically in the past 20 years. We inevitably need to use computing software. R is one of the most popular software among data scientists and in academia. It is an open-source programming language so users can customize existing codes or functions according to their needs. Most state-of-the-art methodologies are implemented as R packages.

While the number of scientific papers is soaring, especially during COVID, a significant amount of the studies can not be reproduced or replicated. This phenomenon is an ongoing crisis termed the replicability crisis. In an effort to produce trustworthy and reproducible results, we use R Markdown. It achieves many goals: 1) Anyone can rerun our study to replicate the results 2) We will be able to run our data analysis and produce reports at the same time. Communication between us and readers/decision-makers is essential.

In this module we will focus on data preparation, data cleaning, and exploratory data analyses (EDA).

R combined with R Markdown will be used. Please go through advanced_R_tutorial.Rmd to learn a set of extremely useful EDA tools such as `dplyr`, `ggplot`, `data.table`, and more.

This lecture is rich and rather involved with both data analyses and R-packages/functions. Perhaps you may read through a compiled file first to grasp the main theme then come back to this .rmd file to run it line by line. That is one way you will turn the coding to yourself!

Contents:

0. Suggested extra readings/doing:
   - run and study Get_staRted.Rmd
   - run and study advanced_R_tutorial.Rmd and advanced_R_tutorial.html
   - Data set: `-MLPayData_Total.csv` `-baseball.csv`
1. Case Study: Billion dollar Billy Beane
2. Study flow:
   - Study design
   - Gathering data
   - Process data (tidy data)
   - Exploratory Data Analysis (EDA)
   - Conclusion/Challenges
3. R functions
   - basic r functions
   - dplyr
   - ggplot

Handy cheat sheets

DPLYR Cheat Sheet

ggplot Cheat Sheet

1 Case Study: Baseball

Baseball is one of the most popular sports in the US. The highest level of baseball teams belong to Major League Baseball, which includes American League and National League with a total of 30 teams. New York Yankees, Boston Red Sox, Philadelphia Phillies and recently rising star team Oakland Athletics are among the top teams. Oakland A's is a low budget team. But the team has been moving itself up in performance mainly due to its General Manager (GM) Billy Beane who is well known to apply statistics in his coaching. In an article Billion dollar Billy Beane, Benjamin Morris studies a regression of performance vs. total payroll among all 30 teams in a 17 years period. He observes that Oakland Athletics's performance is comparable to that of Boston Red Sox, so consequently it explains why Billy Beane is worth 12 million dollars over 5 years. We take this case study for the following purpose:

1. Reproduce Benjamin's study. Is Billy Beane (Oakland A's GM) worth 12.5 million dollars for a period of 5 years, as argued in the article? We challenge Benjamin's reasoning behind his argument.
2. Explore general questions: How does pay and performance relate to each other? Will a team perform better when they are paid more?

1.1 Gathering data

What determines performance is a long arguable topic. Because the original goal of our study is to reproduce the published results, we merely took an easy way out by gathering information on payrolland performance by team from 1998 to 2014. We could have easily **reproduced** all the analyses done in the post only if the data were available! So we reassembled a data set from several websites. Some manual corrections are made. For example we need to consolidate teams with different names due to name change so that. Here is one site among many others that you may find updated information about each team: http://www.stevetheump.com/Payrolls.htm

Data: `MLPayData_Total.csv`, consists of winning records and the payroll of all 30 ML teams from 1998 to 2014 (17 years). There are 162 games in each season.

The variables included are:

- `team name`: team names
- `p2014`: total pay in 2014 in **millions** and other years indicated by year
- `X2014`: number of games won in 2014 and other years labeled
- `X2014.pct`: percent winning in 2014 and other years (We only need one of the two variables from above.)

1.2 Data Preparation

Before we do any analysis, it is a **MUST** that we take a look at the data. In particular, we will try to

Tidy the data:

- Import data/Data preparation
- Data format
- Missing values/peculiarity
- Understand the variables: unit, format, unusual values, etc.
- Put data into a standard data format

Columns: variables Rows: subjects

Read the data: One of the most important aspects of using R is to know how to import data properly. Most of the data is available in a table form as a .csv file already. So the simplest way to do so is to use `read.csv()` (or the faster `fread()` from the `data.table` package).

What is the current working directory? R will find files or save files to the working directory.

```
# getwd()
# dir <- "/Users/lzhao/Dropbox/STAT471/Data"  # my laptop
# setwd(dir)   #same as  setwd("/Users/lzhao/Dropbox/STAT471/Data")
# getwd()
```

Alternatively if the data is in the same folder as the .Rmd file then we can read data directly.

```
datapay <- read.csv("data/MLPayData_Total.csv", header=T, stringsAsFactors = FALSE)
# You can also use the whole path to read in the data. In my case,
# datapay <- read.csv("/Users/lzhao/Dropbox/STAT471/Data/MLPayData_Total.csv", header=T)
# command + return excute the highlighted line(s)
```

What is in the dataset?

Take a quick look at the data. Pay attention to what is in the data, any missing values, and the variable format.

```
names(datapay) # see what variables
```

```
##  [1] "Team.name.2014" "p1998"          "p1999"          "p2000"
##  [5] "p2001"          "p2002"          "p2003"          "p2004"
##  [9] "p2005"          "p2006"          "p2007"          "p2008"
## [13] "p2009"          "p2010"          "p2011"          "p2012"
## [17] "p2013"          "p2014"          "X2014"          "X2013"
## [21] "X2012"          "X2011"          "X2010"          "X2009"
## [25] "X2008"          "X2007"          "X2006"          "X2005"
## [29] "X2004"          "X2003"          "X2002"          "X2001"
## [33] "X2000"          "X1999"          "X1998"          "X2014.pct"
## [37] "X2013.pct"      "X2012.pct"      "X2011.pct"      "X2010.pct"
## [41] "X2009.pct"      "X2008.pct"      "X2007.pct"      "X2006.pct"
## [45] "X2005.pct"      "X2004.pct"      "X2003.pct"      "X2002.pct"
## [49] "X2001.pct"      "X2000.pct"      "X1999.pct"      "X1998.pct"
```

Is anything bothering you? We may want to change names of teams to a shorter, neater name.

```
# hide the results
str(datapay) # data format
summary(datapay) # quick summary. missing values may be shown
```

Everything seems to be OK at the moment other than changing one variable name.

```
datapay <- datapay %>% rename(team = Team.name.2014) # change variable name and also update the data fi
names(datapay)[1:10] # only show 10 names
```

```
##  [1] "team"  "p1998" "p1999" "p2000" "p2001" "p2002" "p2003" "p2004"
##  [9] "p2005" "p2006"
```

Reshape the data

The original format of the dataset `MLPayData_Total.csv` is not in a desirable format. Each row lists multiple results. Also the variable `year` is missing.

```
datapay[1:4, 1:5]  # list a few lines
```

```
##                    team p1998 p1999 p2000 p2001
## 1 Arizona Diamondbacks  31.6  70.5  81.0  81.2
## 2       Atlanta Braves  61.7  74.9  84.5  91.9
## 3    Baltimore Orioles  71.9  72.2  81.4  72.4
```

```
## 4       Boston Red Sox   59.5   71.7   77.9 109.6
```

We would like to reshape the data into the following desirable table format:

- columns (variables) contain all variables
- each row records one result(s)

In our case we have four variables: team, year, pay, win_number and win_percentage. So we would like to rearrange the data into the following form:

team | year | payroll | win_number | win_percentage

Let us do this using `dplyr::pivot_longer()` in the following chunk:

You may skip the detailed reshaping data chunk for the moment. But run the chunk to produce the `datapay_long.csv`

```
payroll <- datapay %>%   # first create variable: payroll and year
  select(team, p1998:p2014) %>%
  pivot_longer(cols = starts_with("p"),
               names_to = "year",
               names_prefix = "p",
               values_to = "payroll")
payroll[1:3, 1:3] # show a few rows

win_num <- datapay %>%  # create variable: win_num and year
  select(team, X1998:X2014) %>%
  pivot_longer(cols = X1998:X2014,
               names_to = "year",
               names_prefix = "X",
               values_to = "win_num")
#win_num[1:3, 1:3]

win_pct <- datapay %>%  # create variable: win_pct and year
  select(team, X1998.pct:X2014.pct) %>%
  pivot_longer(cols = X1998.pct:X2014.pct,
               names_to = "year",
               names_prefix = "X",
               values_to = "win_pct") %>%
  mutate(year = substr(year, 1, 4))
#win_pct[1:3, 1:3]

# join tables into team, year, payrow, win_num, win_pct
datapay_long <- payroll %>%
  inner_join(win_num, by = c("team", "year")) %>%
  inner_join(win_pct, by = c("team", "year"))
head(datapay_long, 2)  # see first 2 rows
```

Take a quick look at the newly formed data file `datapay_long`.

```
names(datapay_long) #names(datapay) new vs. old data files
```

```
## [1] "team"    "year"    "payroll" "win_num" "win_pct"
```

More summaries of the new data:

```
head(datapay_long) # show the first 6 rows
dim(datapay_long)
str(datapay_long)
```

```
summary(datapay_long)
skimr::skim(datapay_long)  # skimr is a package with a func skim that is a summary func with more stati
# It does two things: load package skimr's func skim only
# or specify to use skim() from package skimr
```

Output the cleaned data file

Now we have done the tidy data processing, we will save this cleaned data file into a new table called `baseball.csv` and output this table to the /data folder in our working folder. From now on we will only use the data file `baseball`.

```
write.csv(datapay_long, "data/baseball.csv", row.names = F)
```

Remark: The above data prep process should be put into a separate .r or .rmd file. There is no need to rerun the above data prep portion each time we work on the project. We put the whole project into one file for the purpose of demonstration.

Now we move on to the next part: Data Analysis

2 Exploratory Data Analysis (EDA)

All the analyses done in this lecture will be exploratory. The goal is to see what information we might be able to extract so that it will support the goal of our study. This is an extremely important first step of the data analyses. We try to understand the data, summarize the data, then finally explore the relationships among the variables through useful visualization.

2.1 Part I: Analyze aggregated variables

In this section, we try to use aggregated information such as the total pay for each team and the average performance to see the relationship between performance and the payroll as suggested in Morris's post.

2.1.1 Input the data

Let's first input the clean data `baseball` and do a quick exploration over the data.

```
baseball <- read.csv("data/baseball.csv", header = TRUE, stringsAsFactors = F)
names(baseball)
str(baseball)
summary(baseball)
```

```
## [1] "team"    "year"    "payroll" "win_num" "win_pct"
## 'data.frame':    510 obs. of  5 variables:
##  $ team   : chr  "Arizona Diamondbacks" "Arizona Diamondbacks" "Arizona Diamondbacks" "Arizona Diamo
##  $ year   : int  1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 ...
##  $ payroll: num  31.6 70.5 81 81.2 102.8 ...
##  $ win_num: int  65 100 85 92 98 84 51 77 76 90 ...
##  $ win_pct: num  0.401 0.617 0.525 0.568 0.605 ...
##      team                year         payroll         win_num
##  Length:510         Min.   :1998   Min.   :  8.3   Min.   : 43
##  Class :character   1st Qu.:2002   1st Qu.: 51.3   1st Qu.: 72
##  Mode  :character   Median :2006   Median : 73.3   Median : 81
##                     Mean   :2006   Mean   : 78.1   Mean   : 81
##                     3rd Qu.:2010   3rd Qu.: 95.0   3rd Qu.: 90
```

```
##                       Max.   :2014   Max.   :235.3   Max.   :116
##     win_pct
##  Min.   :0.265
##  1st Qu.:0.444
##  Median :0.500
##  Mean   :0.500
##  3rd Qu.:0.556
##  Max.   :0.716
```

Everything seems to be fine: no missing values, names of variables are good. The class of each variable matches its nature. (numeric, factor, characters…)

2.1.2 Create a new data table

For convenience, we create a new table which only contains the total payroll and average winning percentage for each team. We name them `team`, `payroll_total`, and `win_pct_ave`. We will change the unit of `payroll_total` from million to billion.

```r
# create total and average winning percentage for each team
data_agg <-baseball %>%
  group_by(team) %>%
  summarise(
    payroll_total = sum(payroll)/1000,
    win_pct_ave = mean(win_pct))
str(data_agg)
summary(data_agg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    30 obs. of  3 variables:
##  $ team         : chr  "Arizona Diamondbacks" "Atlanta Braves" "Baltimore Orioles" "Boston Red Sox"
##  $ payroll_total: num  1.22 1.52 1.31 2.1 1.55 ...
##  $ win_pct_ave  : num  0.492 0.563 0.459 0.553 0.476 ...
##      team           payroll_total     win_pct_ave
##  Length:30          Min.   :0.698   Min.   :0.433
##  Class :character   1st Qu.:1.022   1st Qu.:0.473
##  Mode  :character   Median :1.264   Median :0.492
##                     Mean   :1.328   Mean   :0.500
##                     3rd Qu.:1.517   3rd Qu.:0.526
##                     Max.   :2.857   Max.   :0.594
```

2.1.3 Descriptive statistics

To summarize a continuous variable such as `payroll_total` or `win_pct_ave`, we use the following measurements:

- **Center**: sample mean/median
- **Spread**: sample standard deviation
- **Range**: minimum and maximum
- **Distribution**: quantiles

Let us first take a look at `payroll_total`.

Base `R` way:

```r
mean(data_agg$payroll_total)
sd(data_agg$payroll_total)
quantile(data_agg$payroll_total, prob = seq(0, 1, 0.25))
```

```
median(data_agg$payroll_total)
max(data_agg$payroll_total)
min(data_agg$payroll_total)
summary(data_agg$payroll_total)
```

```
## [1] 1.33
## [1] 0.45
##    0%   25%   50%   75%  100%
## 0.698 1.022 1.264 1.517 2.857
## [1] 1.26
## [1] 2.86
## [1] 0.698
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.698   1.022   1.264   1.328   1.517   2.857
```

`dplyr` way:

```
data_agg %>% select(payroll_total) %>%
  summarise(
    mean = mean(payroll_total),
    sd   = sd(payroll_total),
    max = max(payroll_total),
    min = min(payroll_total),
    "0%" = quantile(payroll_total)[1],
    "25%" = quantile(payroll_total)[2],
    "50%" = quantile(payroll_total)[3],
    "75%" = quantile(payroll_total)[4],
    "100%" = quantile(payroll_total)[5]
  )
```

```
## # A tibble: 1 x 9
##    mean    sd   max   min  `0%` `25%` `50%` `75%` `100%`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1  1.33 0.450  2.86 0.698 0.698  1.02  1.26  1.52   2.86
```

Find the team with the max/min payroll:

Base `R` way:

```
data_agg$team[which.max(data_agg$payroll_total)]
```

```
## [1] "New York Yankees"
```

```
data_agg$team[which.min(data_agg$payroll_total)]
```

```
## [1] "Miami Marlins"
```

Rearrange the data to see the ranks of team by `payroll`

But we can easily rearrange the whole data set `data_agg` by ordering one variable, say `payroll_total`.

Base `R` way:

```
#To rank teams by payroll in decreasing order
arrange(data_agg, desc(payroll_total))[1:6,] #default decs=T
```

```
## # A tibble: 6 x 3
##   team                  payroll_total win_pct_ave
##   <chr>                         <dbl>       <dbl>
## 1 New York Yankees               2.86       0.594
```

```
## 2 Boston Red Sox                2.10       0.553
## 3 Los Angeles Dodgers           1.87       0.529
## 4 New York Mets                 1.72       0.502
## 5 Philadelphia Phillies         1.69       0.519
## 6 Los Angeles Angels            1.66       0.540
```

dplyr way:

```
data_agg %>% select(team,payroll_total) %>% filter(payroll_total == max(payroll_total))
data_agg %>% select(team,payroll_total) %>% filter(payroll_total == min(payroll_total))
```

```
## # A tibble: 1 x 2
##   team             payroll_total
##   <chr>                    <dbl>
## 1 New York Yankees          2.86
## # A tibble: 1 x 2
##   team          payroll_total
##   <chr>                 <dbl>
## 1 Miami Marlins         0.698
```

dplyr way:

```
data_agg %>%
  arrange(payroll_total) %>%
  slice(1:6)  # select first 6 rows
```

```
## # A tibble: 6 x 3
##   team               payroll_total win_pct_ave
##   <chr>                      <dbl>       <dbl>
## 1 Miami Marlins              0.698       0.468
## 2 Pittsburgh Pirates         0.772       0.443
## 3 Tampa Bay Rays             0.776       0.462
## 4 Kansas City Royals         0.870       0.433
## 5 Oakland Athletics          0.888       0.539
## 6 San Diego Padres           0.940       0.483
```

2.1.4 Displaying variables

payroll_totals are clearly different for different teams. How does it vary? We use the distribution to describe the variability.

A **histogram** shows the distribution of the payroll.

Base R plots:

```
hist(data_agg$payroll_total, breaks=5, freq = F) # default: freq =T -> count
```

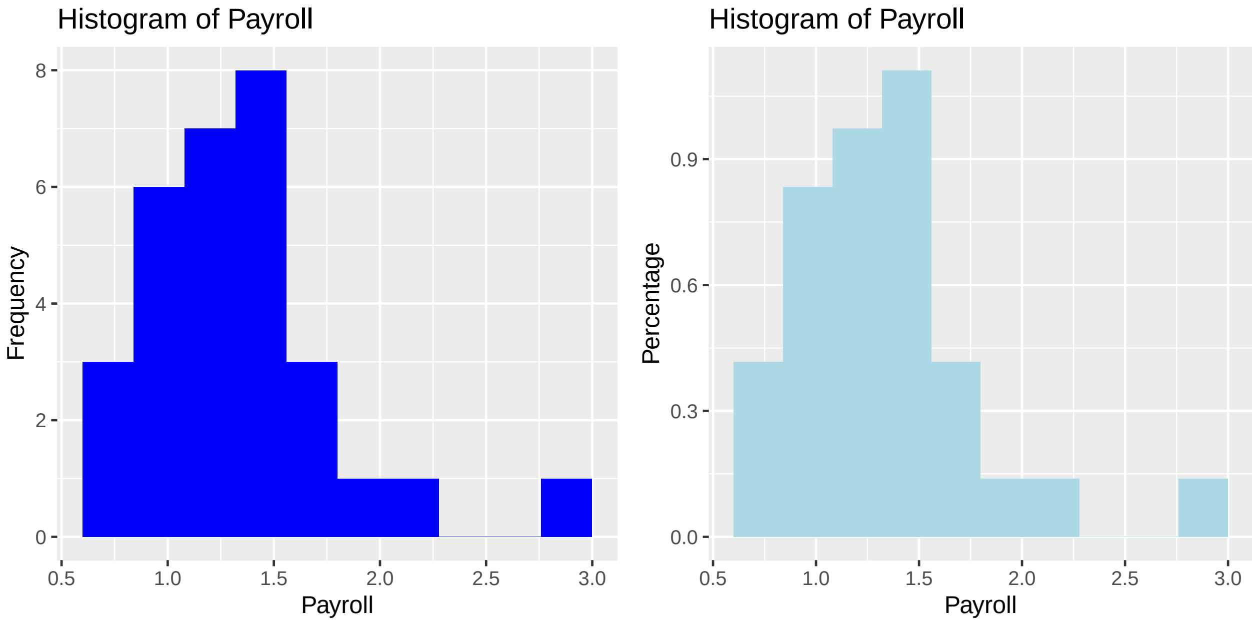
```
hist(data_agg$payroll_total, breaks=10, col="blue") # make larger number of classes to see the details
```

ggplot plots:

```
p1 <- ggplot(data_agg) +
  geom_histogram(aes(x = payroll_total), bins = 10, fill = "blue") +
  labs( title = "Histogram of Payroll", x = "Payroll" , y = "Frequency")

p2 <- ggplot(data_agg) +
  geom_histogram(aes(x = payroll_total, y = ..density..), bins = 10, fill = "light blue") +
  labs( title = "Histogram of Payroll", x = "Payroll" , y = "Percentage")
```

```r
grid.arrange(p1, p2, ncol = 2) # facet the two plots  side by side
```



Notice, the two plots above look identical but with different `y-scale`.

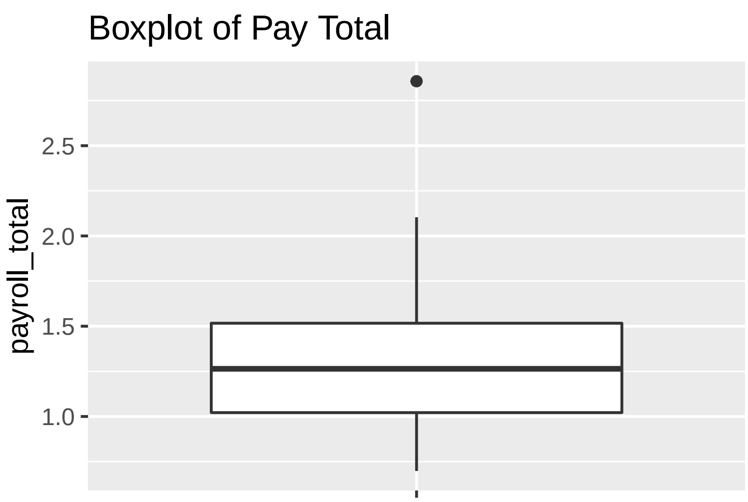
A **boxplot** captures the spread by showing median, quantiles and outliers:

Base `R` plots:

```r
boxplot(data_agg$payroll_total,
        main = "Boxplot of Payroll",
        ylab = "payroll")
```

`ggplot` plots:

```r
ggplot(data_agg) +
  geom_boxplot(aes(x="", y=payroll_total)) +
  labs(title="Boxplot of Pay Total", x="")
```




```r
# theme_bw() #default is gray
```

2.1.5 Normal variables

When would the sample mean and sample standard deviation help us to describe the distribution of a variable? As an exercise, let us summarize the variable `win_pct_ave`.

```
mean(data_agg$win_pct_ave) # sort(data_agg$win_pct_ave)
sd(data_agg$win_pct_ave)
```

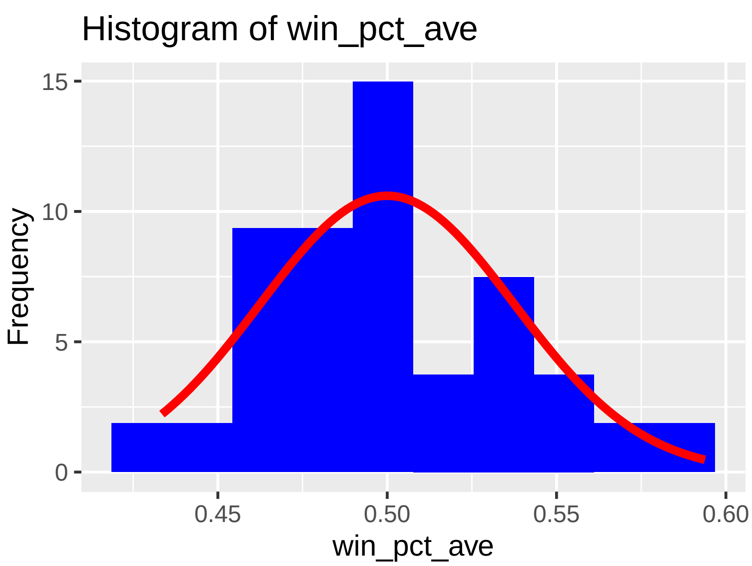
```
## [1] 0.5
## [1] 0.0376
```

We see that `win` on average is 0.5 with a SD being 0.038. How would the mean and sd be useful in describing the distribution of `win`? **Only if the histogram looks like a bell curve**!

Take a look at the histogram of `win`. Here we impose a **normal curve** with the center being 0.5 and the spread, sd = 0.038.

```
# create a sequence of (0, 0.001, 0.002, ..., 1)
x <- seq(0, 1, .001)
# density along the sequence
y <- dnorm(x, mean(data_agg$win_pct_ave), sd(data_agg$win_pct_ave))
# plot the histogram of "win" and normal curve N(0.5, 0.04^2)
hist(data_agg$win_pct_ave, freq = F, col="red")
lines(x, y, col="blue", lwd=5) # lwd: line width
```

```
ggplot(data_agg) +
  geom_histogram(aes(x=win_pct_ave, y = ..density..), bins=10, fill= "blue" ) +
  stat_function(fun = dnorm, args = list(mean = mean(data_agg$win_pct_ave),
                                         sd = sd(data_agg$win_pct_ave)), colour = "red",
  labs( title = "Histogram of win_pct_ave", x = "win_pct_ave" , y = "Frequency")
```




```
# another way to do it
# ggplot(data_agg) +
# geom_histogram(aes(x=win_pct_ave, y = ..density..), bins=10, fill= "light blue" )+
# geom_density(aes(x=win_pct_ave), kernel = "gaussian", color = "blue", size = 3) # kernel smoothing
```

The smoothed normal curve captures the shape of the histogram of win. Or we will say that the variable `win` follows a normal distribution approximately. Then we can describe the distribution of `win` using the two numbers: mean and sd.

Roughly speaking

- 68% of teams with `win` to be within one sd from the mean.

$$0.5 \pm 0.038 = [0.462, 0.538]$$

- 95% of the teams with `win` to be within 2 sd from the mean:

$$0.5 \pm 2 * 0.038 = [0.425, 0.575]$$

- 2.5% of the teams with `win` to be higher 2.5 times of sd above the mean:

$$> 0.5 + 2 * 0.038 = 0.575$$

2.1.6 Explore the relationship between `payroll_total` and `win_pct_ave`.

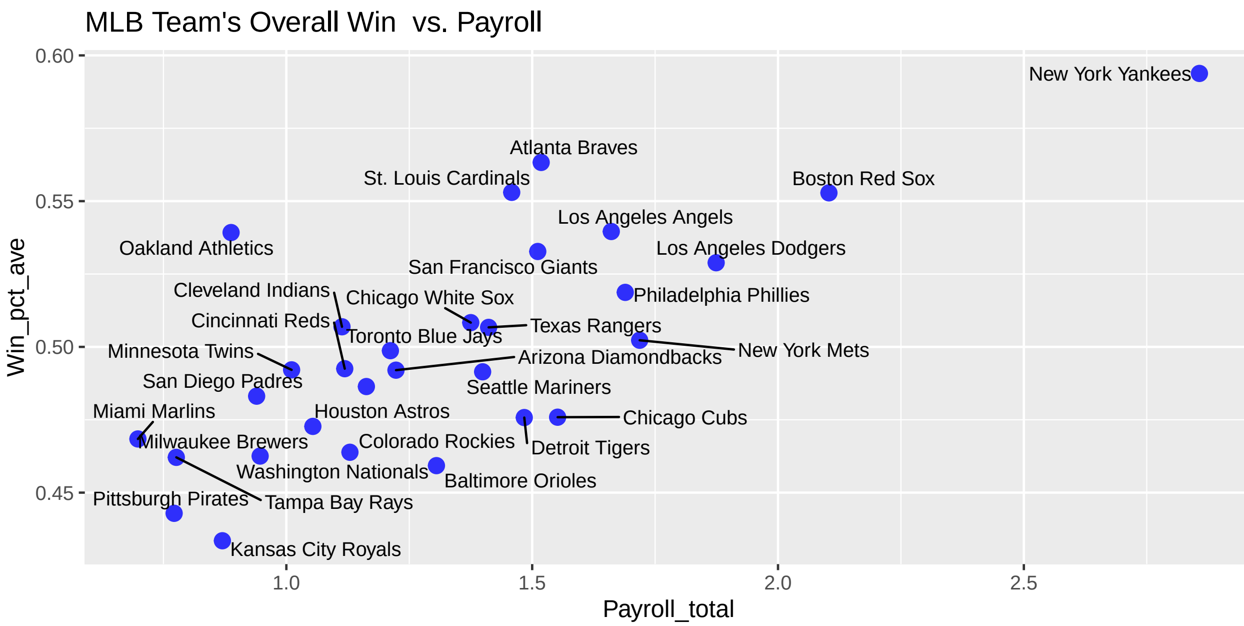
Scatter plots show the relationship between x variable `payroll_total` and y variable `win_pct_ave`. We are looking for patterns between the two variables, such as a linear or quadratic relationship.

```
plot(x = data_agg$payroll_total,
     y = data_agg$win_pct_ave,
     pch  = 16,     # "point character": shape/character of points
     cex  = 0.8,    # size
     col  = "blue", # color
     xlab = "Payroll_total",  # x-axis
     ylab = "Win_ave",  # y-axis
     main = "MLB Team's Overall Win vs. Payroll") # title
text(data_agg$payroll_total, data_agg$win_pct_ave, labels=data_agg$team, cex=1, pos=1) # label all poin
```

We notice the positive association: when `payroll_total` increases, so does `win_pct_ave`.

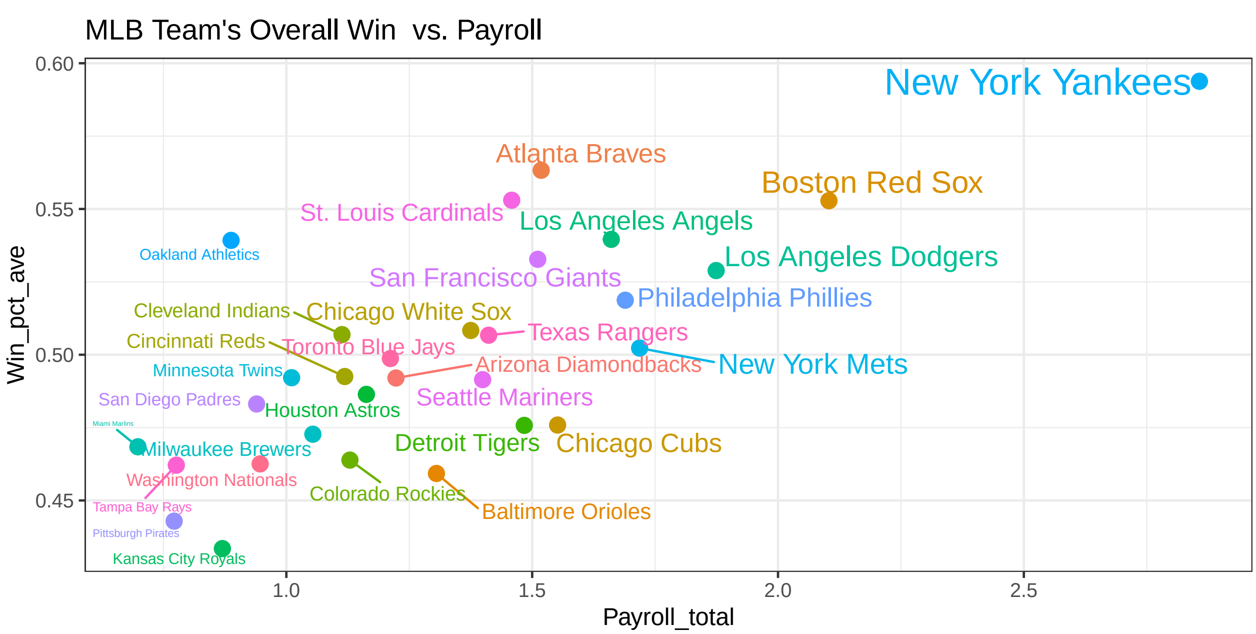
ggplot

```
data_agg %>%
  ggplot(aes(x = payroll_total, y = win_pct_ave)) +
  # geometric options: color, size, shape, alpha: transparency (range: 0 to 1)
  geom_point(color = "blue", size= 3, alpha = .8) +
  geom_text_repel(aes(label = team), size = 3) +
  labs(title = "MLB Team's Overall Win  vs. Payroll",
       x = "Payroll_total",
       y = "Win_pct_ave")
```



Compare with the previous plot. We can bring in other variables to adjust the color, size, and alpha of the scatter plot via aesthetic mapping.

```
data_agg %>%
  ggplot(aes(x = payroll_total, y = win_pct_ave)) +
  # geometric options with aes mapping:
  # color, size, alpha as a function of a variable
  geom_point(aes(color = team), size = 3) +
  geom_text_repel(aes(color = team, label = team, size = payroll_total)) +
  labs(title = "MLB Team's Overall Win  vs. Payroll",
       x = "Payroll_total",
       y = "Win_pct_ave") +
  theme_bw() +
  theme(legend.position = "none")
```

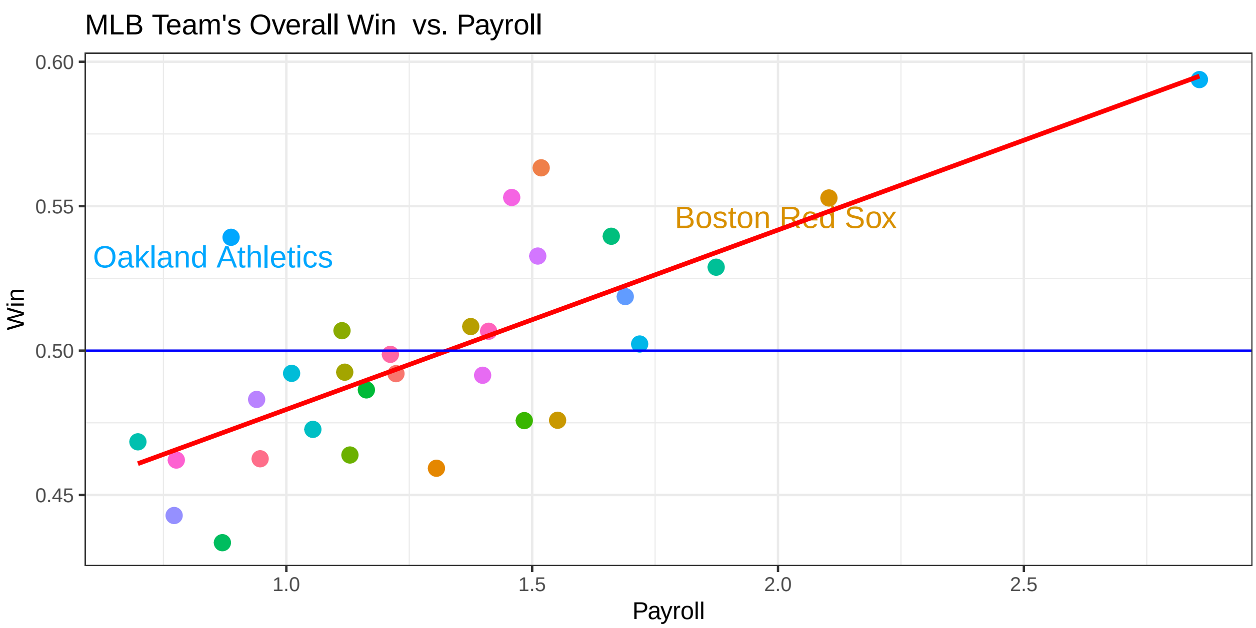


Least Squared Lines

The simplest function to capture the relationship between pay and performance is through the linear model. We impose the least squared equation on top of the scatter plot using `ggplot()` with `geom_smooth()`. We also annotate the two teams `Oakland Athletics` and `Boston Red Sox`.

```
selected_teams <- c("Oakland Athletics", "Boston Red Sox")

data_agg %>%
  ggplot(aes(x = payroll_total, y = win_pct_ave)) +
  geom_point(aes(color = team), size = 3) +
  # only show names of selected_teams
  geom_text_repel(data = subset(data_agg, team %in% selected_teams),
                  aes(label = team, color = team), size = 5) +
  geom_smooth(method = "lm", formula = y ~ x, se = F,color = "red") +
  geom_hline(aes(yintercept = mean(win_pct_ave)), color = "blue") +
  labs(title = "MLB Team's Overall Win  vs. Payroll",
       x = "Payroll",
       y = "Win") +
  theme_bw() +
  theme(legend.position = "none")
```



Conclusions/Discussions

Answer to Question 1:

HERE is how the article concludes that Beane is worth as much as the GM in Red Sox. By looking at the above plot, Oakland A's win pct is more or less the same as that of Red Sox, so based on the LS equation, the team should have paid 2 billion!

Do you agree with this argument? Why or why not?

Answer to Question 2:

From this regression line, we see a clear upward trend. Or precisely the least squared equation has a positive coefficient. Consequently, the more a team is paid the better performance we expect the team has.

Questions for you:

Do you agree with the conclusions made based on a regression analysis shown above? How would you carry out a study which may have done a better job? In what way?

2.2 Part II: Analyze pay and winning percent over time and by team

Payroll and performance vary depending on teams and years. We investigate changes over time and by teams to see how payroll relates to performance.

2.2.1 Compare payroll and performance

We can compare summary statistics of payrolls and performance among teams.

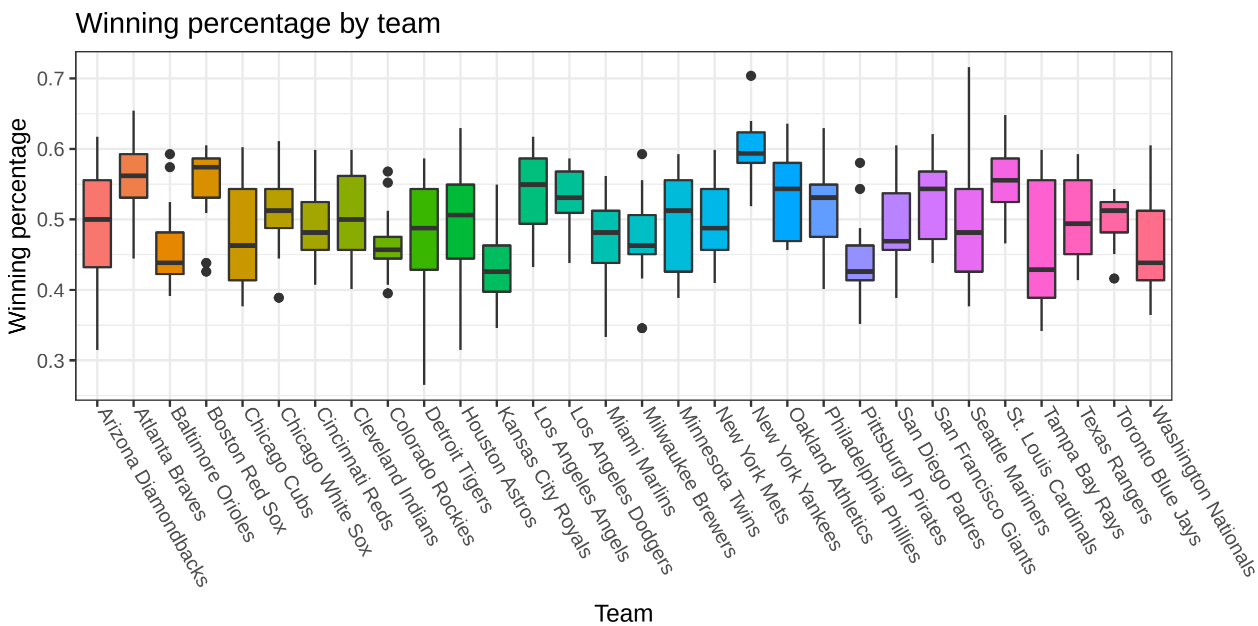
```
baseball %>% group_by(team) %>%
  summarise(payroll_mean = mean(payroll),
            win_pct_mean = mean(win_pct)) %>%
  arrange(-payroll_mean) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##    team                  payroll_mean win_pct_mean
##    <chr>                        <dbl>        <dbl>
##  1 New York Yankees             168.         0.594
##  2 Boston Red Sox               124.         0.553
##  3 Los Angeles Dodgers          110.         0.529
##  4 New York Mets                101.         0.502
##  5 Philadelphia Phillies         99.4        0.519
##  6 Los Angeles Angels            97.7        0.540
##  7 Chicago Cubs                  91.3        0.476
##  8 Atlanta Braves                89.3        0.563
##  9 San Francisco Giants          88.9        0.533
## 10 Detroit Tigers                87.3        0.476
```

We see that New York Yankees has the highest payroll. Boston Red Sox is the next highest paid team. But the time effect is not included here.

Summary statistics can not describe the distributions of either payroll or performances. Back to back boxplots of payroll or winning percentage would capture the variability in details.

```
baseball %>%
  ggplot(aes(x = team, y = win_pct, fill = team)) +
  geom_boxplot() +
  xlab("Team") +
  ylab("Winning percentage") +
  ggtitle("Winning percentage by team") +
  theme_bw() +
  theme(legend.position = "none",
        # adjust for margins around the plot; t: top; r: right; b: bottom; l: left
        plot.margin = margin(t = 5, r = 50, b = 5, l = 0, unit = "pt"),
        axis.text.x = element_text(angle = -60, vjust = 0, hjust = 0))
```

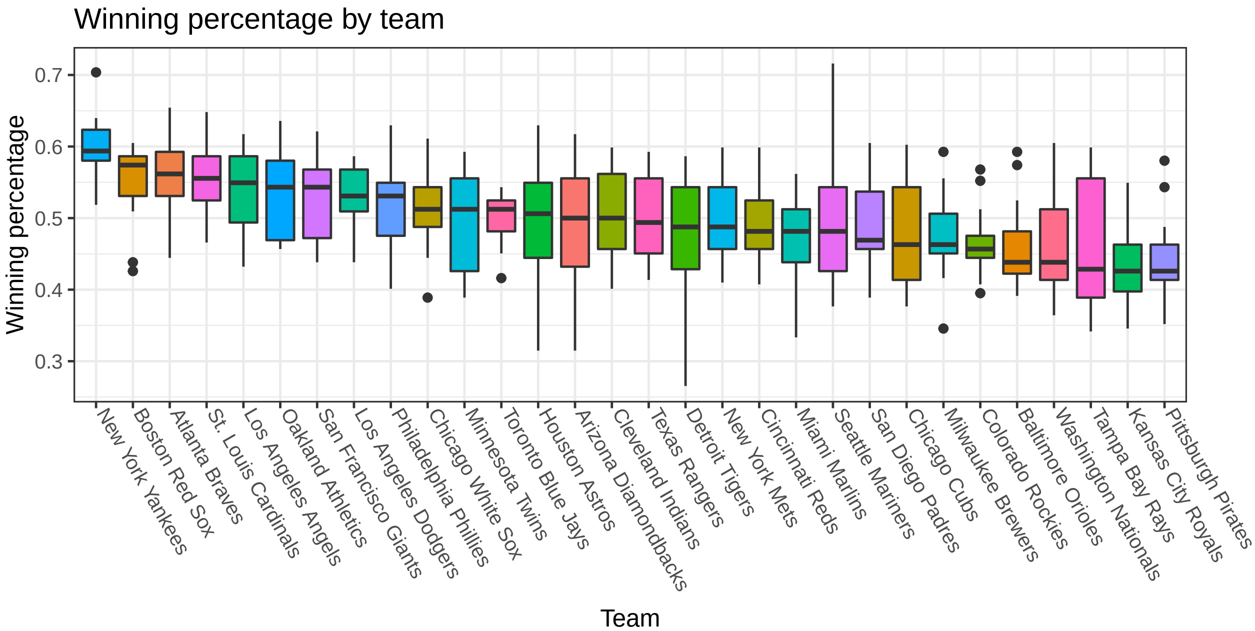


We see clearly that the medians/means and spreads are very different. Is there a more informative way to display this?

We probably want to display the comparison by ranking the median for example:

```
boxplot_theme <-
  theme_bw() +
  theme(legend.position = "none",
        # adjust for margins around the plot; t: top; r: right; b: bottom; l: left
        plot.margin = margin(t = 5, r = 50, b = 5, l = 0, unit = "pt"),
        axis.text.x = element_text(angle = -60, vjust = 0, hjust = 0))
```

```
baseball %>%
  ggplot(aes(x = forcats::fct_reorder(team, -win_pct, .fun = median), #order win_pct in a decreasing or
             y = win_pct, fill = team)) +
  geom_boxplot() +
  xlab("Team") +
  ylab("Winning percentage") +
  ggtitle("Winning percentage by team") +
  boxplot_theme
```



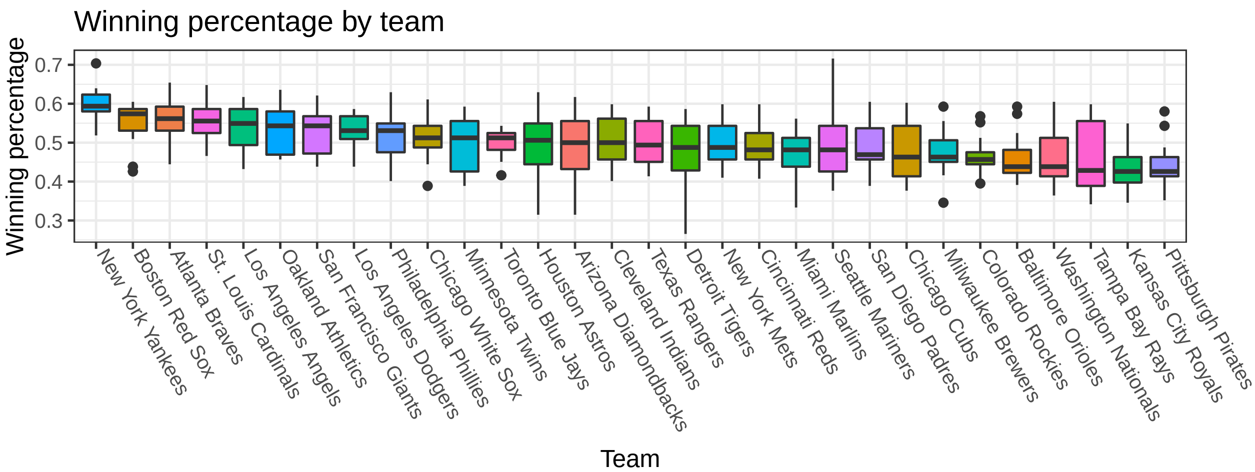
We see that `NY Yankees` and `Red Sox` are consistently good teams while `Oakland A's` has a good overall team performance but the performance varies.

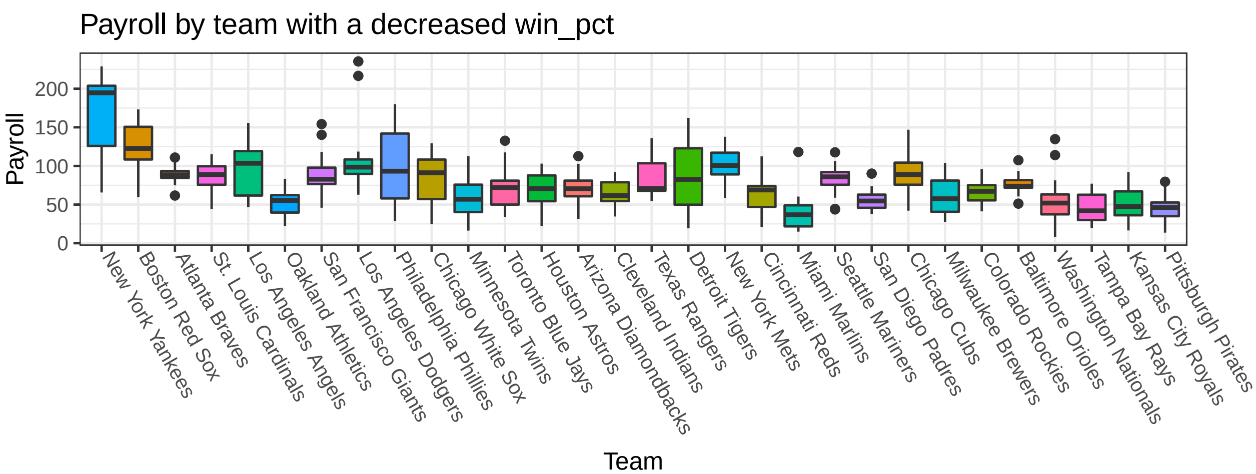
Next we would like to compare both `payroll` and `win_pct` by teams. Let us try to line up two back to back boxplots together. Notice that we tried to rank one variable while carrying the other variable in the same order. The hope is to reveal the relationship between `payroll` and `performance`.

```
# use reorder_within() and scale_x_reordered() from tidytext to order boxplot within each facet
library(tidytext)
p_win_pct <- baseball %>%
  ggplot(aes(x = forcats::fct_reorder(team, -win_pct, .fun = median), #order win_pct in a decreasing or
             y = win_pct, fill = team)) +
  geom_boxplot() +
  xlab("Team") +
  ylab("Winning percentage") +
  ggtitle("Winning percentage by team") +
  boxplot_theme


p_payroll <- baseball %>%
  ggplot(aes(x = forcats::fct_reorder(team, -win_pct, .fun = median), #order win_pct in a decreasing or
             y = payroll, fill = team)) +
  geom_boxplot() +
  xlab("Team") +
  ylab("Payroll") +
  ggtitle("Payroll by team with a decreased win_pct") +
  boxplot_theme

gridExtra::grid.arrange(p_win_pct, p_payroll, ncol=1)
```






```r
# ggpubr::ggarrange(p_win_pct, p_payroll, ncol = 1)
```

Bingo! While `Oakland A's` payroll are consistently lower than that of `Red Sox`, they have similar performance!!!

```r
# use reorder_within() and scale_x_reordered() from tidytext to order boxplot within each facet
library(tidytext)
# facet names
facet_names <- c("payroll" = "Payroll",
                 "win_pct" = "Winning percentage")
baseball %>%
  select(-win_num) %>%
  pivot_longer(cols = c("payroll", "win_pct"),
               names_to = "variable") %>%
  ggplot(aes(x = reorder_within(team, -value, variable, fun = median),
             y = value, fill = team)) +
  geom_boxplot() +
  scale_x_reordered() +
  facet_wrap(~ variable, ncol = 1, scales = "free",
             labeller = as_labeller(facet_names)) +
  xlab("Team") + ylab("") +
  ggtitle("Payroll and winning percentage by team") +
  boxplot_theme
```

2.2.2 Comparing performance as a function of time

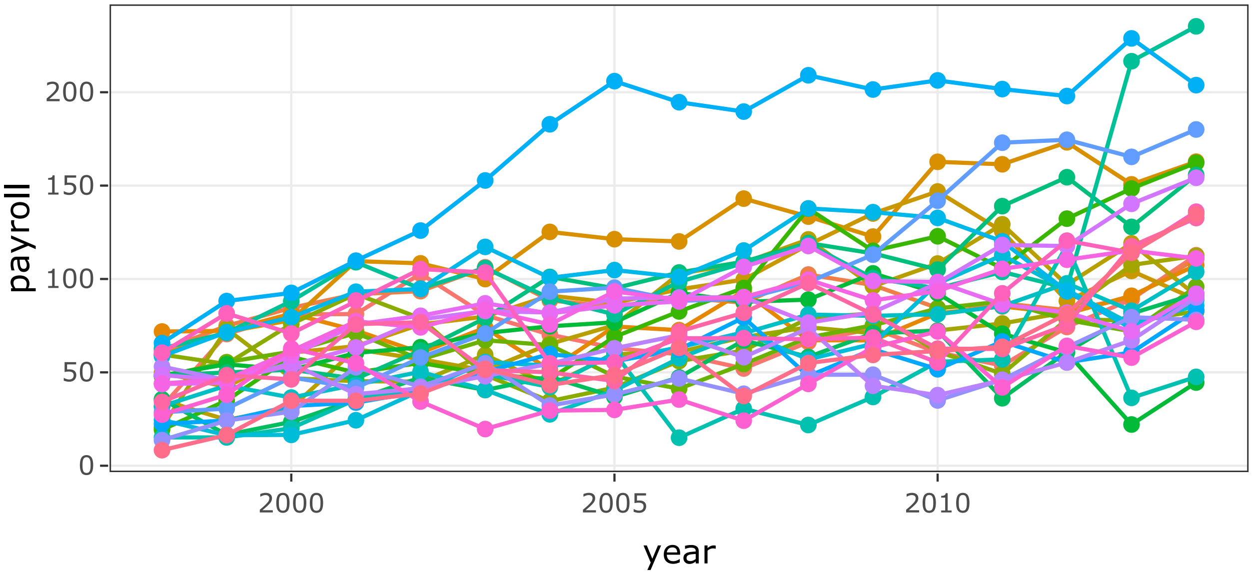
A time series of performance may reveal patterns of performance over the years to see if some teams are consistently better or worse.

Payroll plot

```
payroll_plot <- baseball %>%
  ggplot(aes(x = year, y = payroll, group = team, col = team)) +
  geom_line() +
  geom_point() +
  theme_bw()
ggtitle("Winning percentage over years")
```

```
## $title
## [1] "Winning percentage over years"
##
## attr(,"class")
## [1] "labels"
```

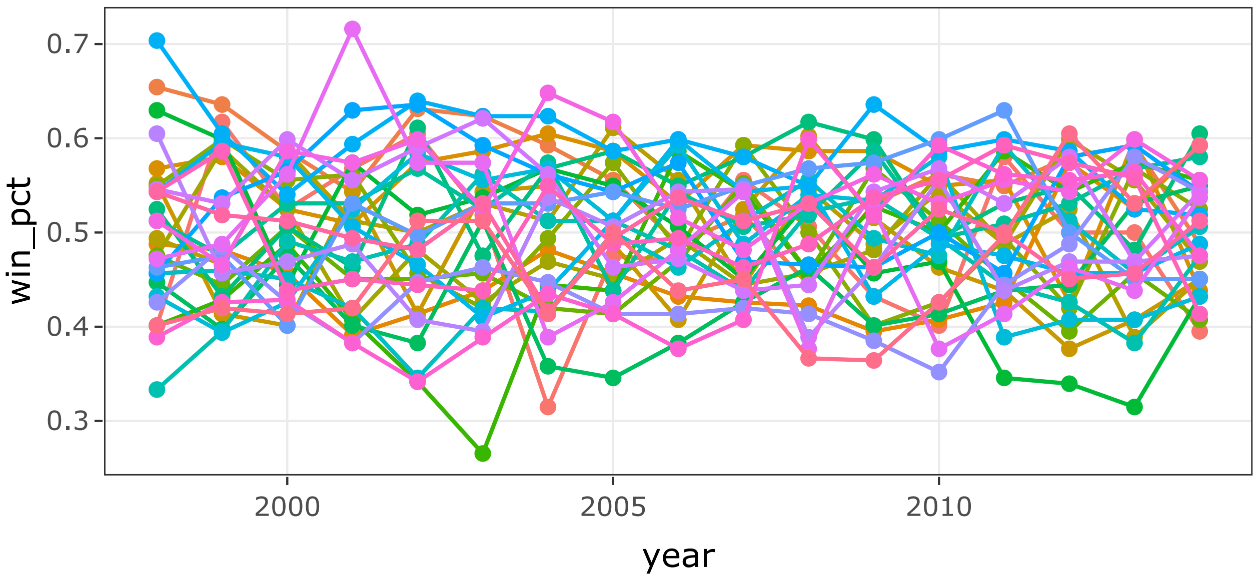
```
ggplotly(payroll_plot +
           theme(legend.position = "none"))
```



Winning pct vs year:

```
win_pct_plot <- baseball %>%
  ggplot(aes(x = year, y = win_pct, group = team, col = team)) +
  geom_line() +
  geom_point() +
  theme_bw()

ggplotly(win_pct_plot +
           theme(legend.position = "none"))
```

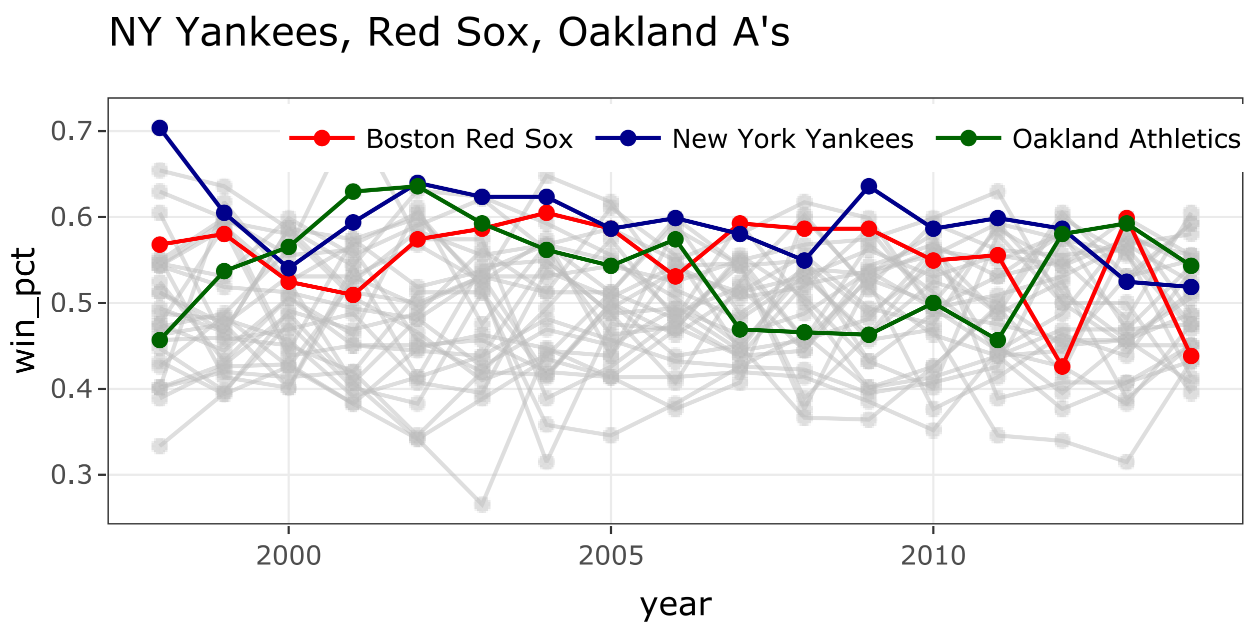


Winning pct plot with only `NY Yankees` (blue), `Boston Red Sox` (red) and `Oakland Athletics` (green) while keeping all other teams as background in gray.

```
selected_teams <- c("New York Yankees", "Boston Red Sox", "Oakland Athletics")

win_pct_plot <- baseball %>%
  ggplot(aes(x = year, y = win_pct, group = team)) +
  geom_line(col = "grey", alpha = .5) +
  geom_point(col = "grey", alpha = .5) +
  geom_line(data = subset(baseball, team %in% selected_teams),
            aes(col = team)) +
  geom_point(data = subset(baseball, team %in% selected_teams),
            aes(col = team)) +
  scale_color_manual(values = c("red", "darkblue", "darkgreen")) +
  theme_bw() +
  ggtitle("NY Yankees, Red Sox, Oakland A's")

ggplotly(win_pct_plot) %>%
  # ggplotly use layout to adjust legend
  layout(legend = list(x = 0.35, y = 0.99, orientation = 'h'))
```



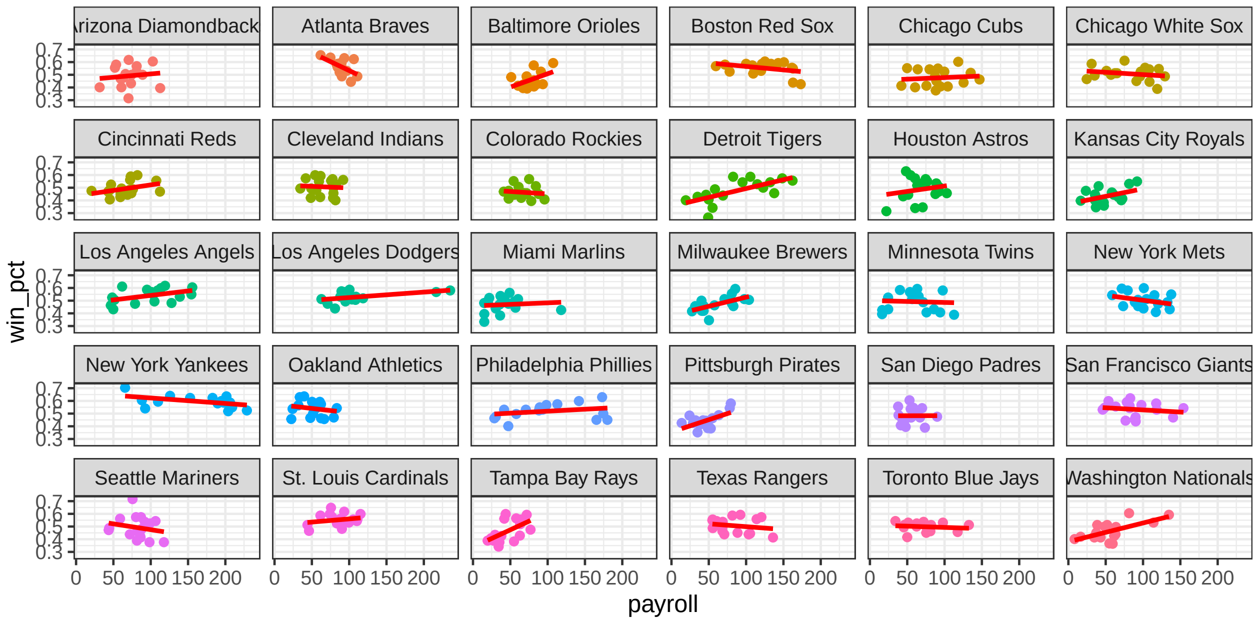
Now we see that `Red Sox` seems to perform better most of the time compared to the `Oakland A's`.

2.2.3 Performance, Payroll and Year

We are trying to reveal the relationship between `performance` and `payroll`. But it depends on which team at a given year.

Scatter plots of `payroll` v.s. `win_pct` by `team`

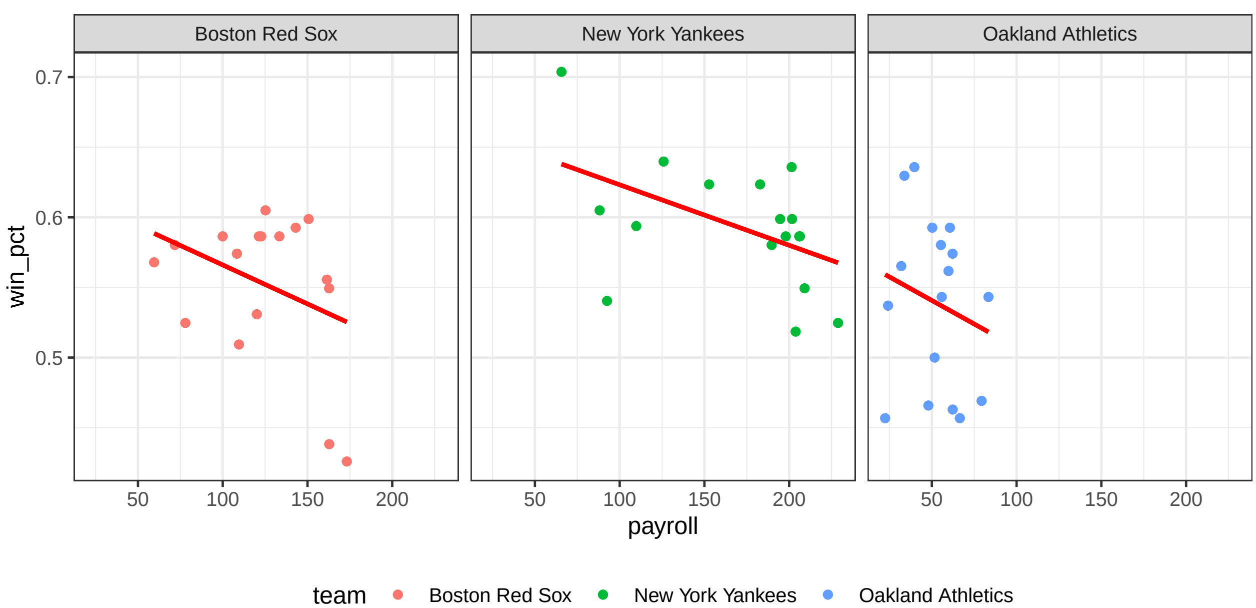
```
baseball %>%
  ggplot(aes(x=payroll, y=win_pct, group = team, color=team)) +
  geom_point()+
  geom_smooth(method="lm", formula=y~x, se=F,color = "red")+
  facet_wrap(~team) +
  theme_bw() +
  theme(legend.position = "none")
```



We see a discrepancy among teams for the relationship between `payroll` and `performance`. The positive trends vary from very positive to even negatively correlated.

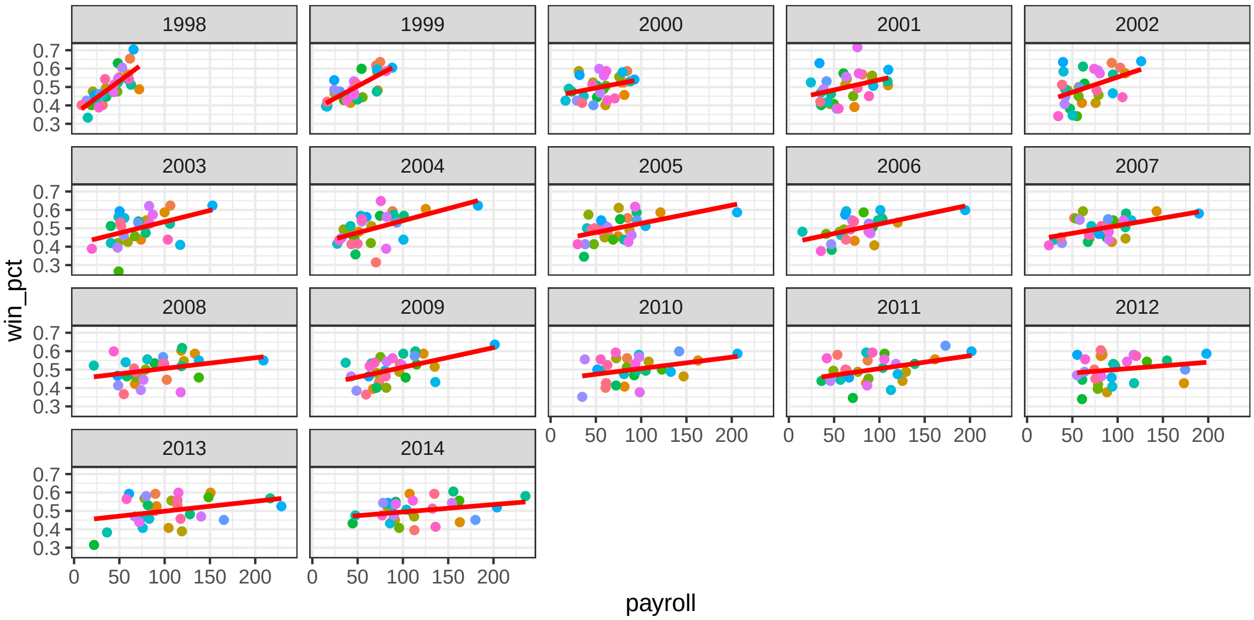
If we zoom in on a few teams we see a clear negative correlation between payroll and performance. What is missing here?

```
#unique(baseball$team)
baseball %>%
  filter(team %in% c("New York Yankees", "Boston Red Sox", "Oakland Athletics")) %>%
  ggplot(aes(x=payroll, y=win_pct, group = team, color=team)) +
  geom_point()+
  geom_smooth(method="lm", formula= y~x,  se=F,color = "red")+
  facet_wrap(~team) +
  theme_bw() +
  theme(legend.position = "bottom")
```



We have seen before, `payroll` increases over years. It will be better to examine `payroll` v.s. `win_pct` by `year`:

```
baseball %>%
  ggplot(aes(x=payroll, y=win_pct, group = year, color=team)) +
  geom_point()+
  geom_smooth(method="lm", formula=y~x, se=F,color = "red")+
  facet_wrap(~year) +
  theme_bw() +
  theme(legend.position = 0)
```



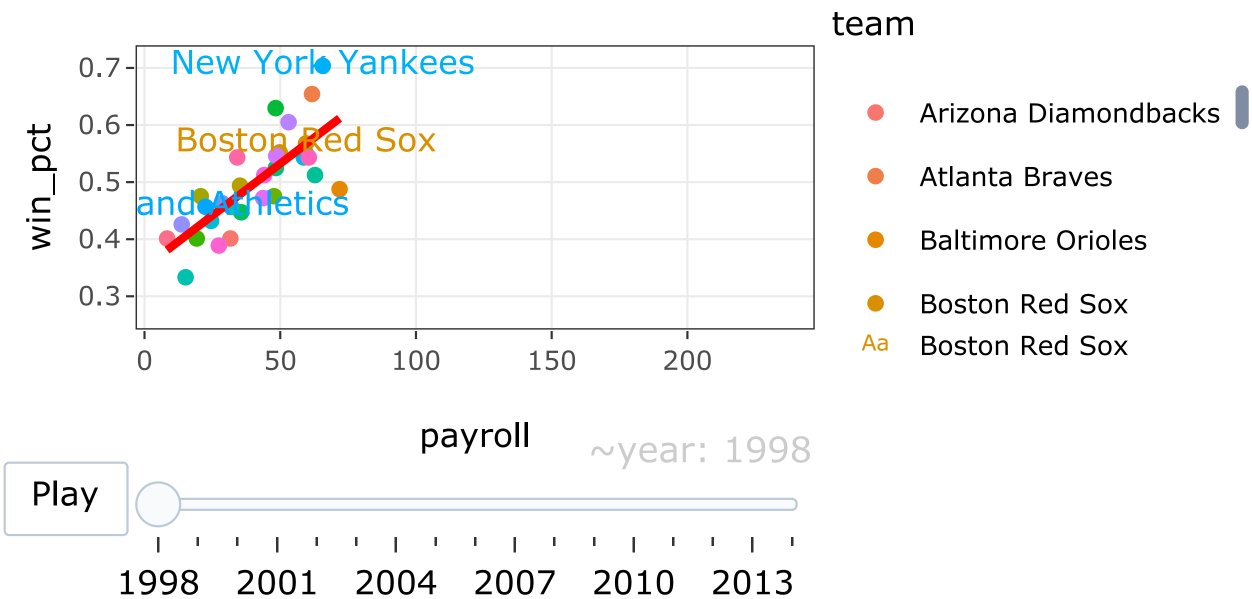
Now it seems to agree with our intuition, payroll and performance are indeed positively related for a given year. But the degree of relationship seems to change depending on which year and they are heavily controlled by some teams.

We can summarize the above three dimension plots via a movie that tracks dynamic changes!

```
selected_team <- c("Oakland Athletics", "New York Yankees", "Boston Red Sox")

p <- baseball %>%
  ggplot(aes(x=payroll, y=win_pct, color=team, frame = year)) +
   theme(legend.position = 0) +
  geom_point() +
  geom_smooth(method="lm", formula=y~x, se=F, color = "red") +
  geom_text(data = subset(baseball, team %in% selected_team),
            aes(label = team),
            show.legend = FALSE) +
  theme_bw()


ggplotly(p)
```



Perhaps we do not see strong evidence that `Oakland A's` is comparable to `Red Sox` in performance.

3 Conclusions and Discussion

We have shown the power of exploratory data analysis to reveal correlation between payroll and performance. However, team performance changes. Is payroll an important factor affecting the team performance if taking more factors into account? While this is a much more complex question of interest in general, we here only assembled a data set containing performance, payroll at team level over a span of 17 years. We have seen the analysis via aggregated statistics can be misleading. In addition to the variation among teams, there can be also substantial variation within each team as well. For example, the payroll distribution within each team is drastically different. See this article on MLB income inequality.

Questions remain:

1. Based on our current data,
   a) what model will you consider to capture effects of payroll, year and team over the performace?
   b) would you use other measurements as dependent variable, e.g. annual payroll increase?
2. If you are asked to run the study to find out what are the main factors affecting performance, how would you do it? To narrow down the scope of the first step of the study, what information you may gather?

4 Appendix: Sample Statistics

We remind readers of the definition of sample statistics here.

- Sample mean:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- Sample variance:

$$s^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}$$

- Sample Standard Deviation:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1}}$$

- Sample correlation

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}$$