Lesson 5 Boardsheet

Kevin Cummiskey

1/22/2020

```
library(Lahman)
library(tidyverse)
```

Functions

Frequently, we need to repeat the same task many times. Using functions will make your life much easier. (I've seen many instances of lengthy code that could have accomplished the same thing in a few lines.) See page 60 in our text for additional discussion of today's lesson.

Last class, we talked about how teams employ relief pitchers has changed dramatically in the last 40 years. For today's class, we want to find the top five pitchers from each team and season since 1970 with at least 20 starts. Briefly summarize how you would approach coding this task.

Here is a function that returns the pitchers with the most starts in a data frame with at least a minimum number of starts.

```
#this function finds the top n.starters in games started in a
#data frame of pitchers with at least GS.min starts
# Arguments
# d - data frame containing pitchers
# GS.min - minimum number of start; default is 10
# n.starters - the number of starters to return
get_starting_pitchers <- function(d,n.starters,GS.min = 10){
  d = d %>%
    filter(GS >= GS.min) %>%
    arrange(desc(GS)) %>%
    head(n.starters)
  return(d)
}
```

What are the arguments of this function?

What does variable scope mean?

We can use this function on any data frame containing the necessary variables. Write one line of code to find the ten pitchers with the most starts in the Pitching data frame.

More frequently, we want to apply this function over many data frames. Here is how to get the pitchers with the most starts by team and season.

```
top.starters = Pitching %>%
  filter(yearID >= 1970) %>%
  split(list(pull(.,yearID),
             pull(.,teamID)),
        drop = TRUE) %>%
  map_df(get_starting_pitchers,
         n.starters = 5,
         GS.min = 25)
```

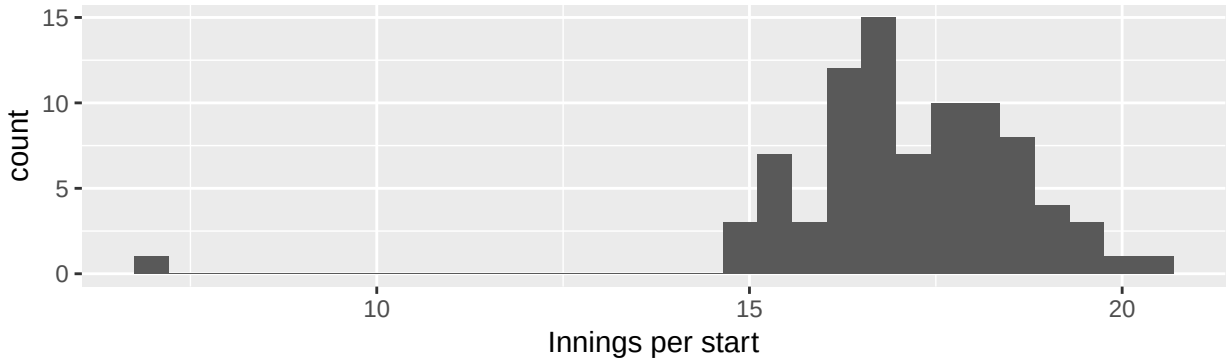
Graphics

Univariate

Histograms and boxplots help us look at a single variable at a time. Let's look at innings pitched per start. (note that we should be excluding the innings these pitchers pitched in relief.)

```
top.starters %>%
  filter(yearID == 2018) %>%
  ggplot(aes(x = IPouts/GS)) +
  geom_histogram() +
  labs(x = "Innings per start")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



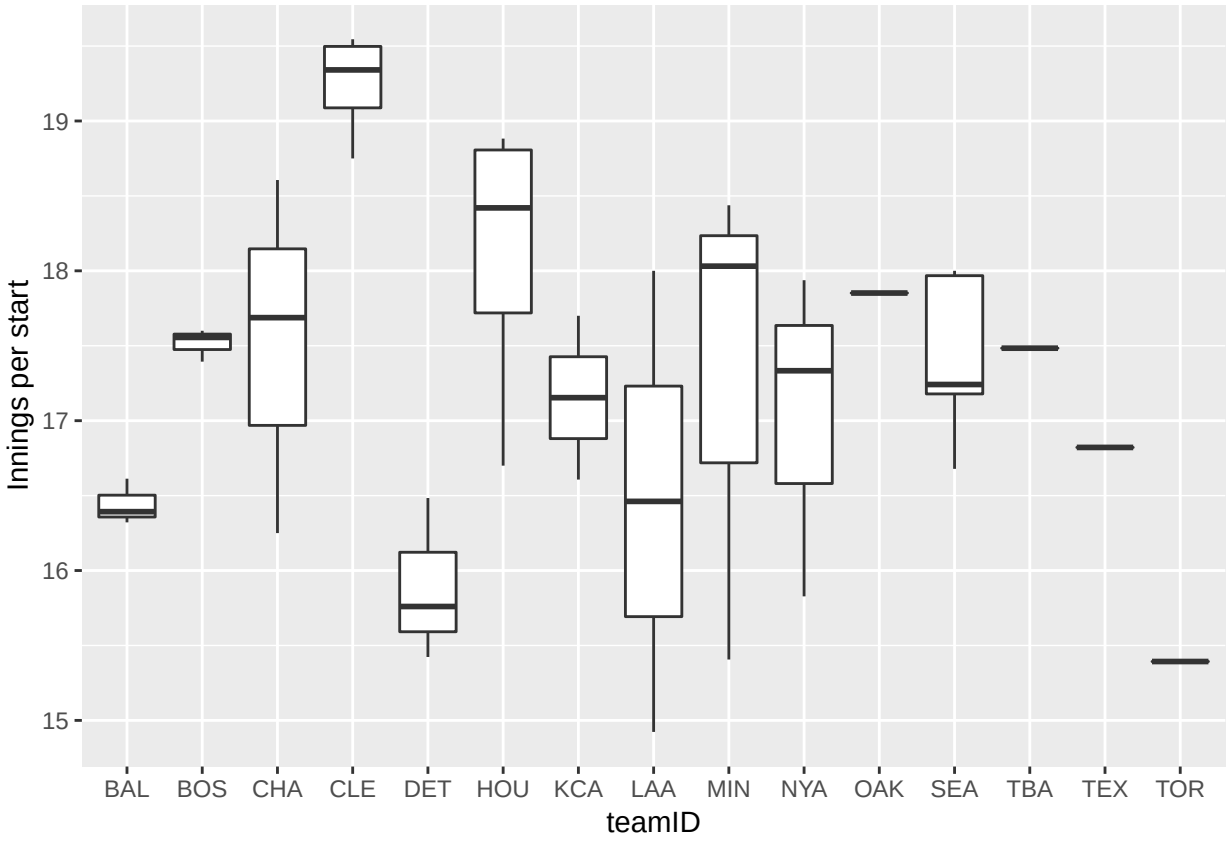
We would want to investigate the pitcher with almost zero innings per start. It turns out this pitcher had many inning in relief.

```
top.starters = top.starters %>%
  filter(IPouts/GS > 10)
```

Categorical and Quantitative Variables

Here is a look at innings pitched per start by team. Side by side boxplots are typically the way to go here.

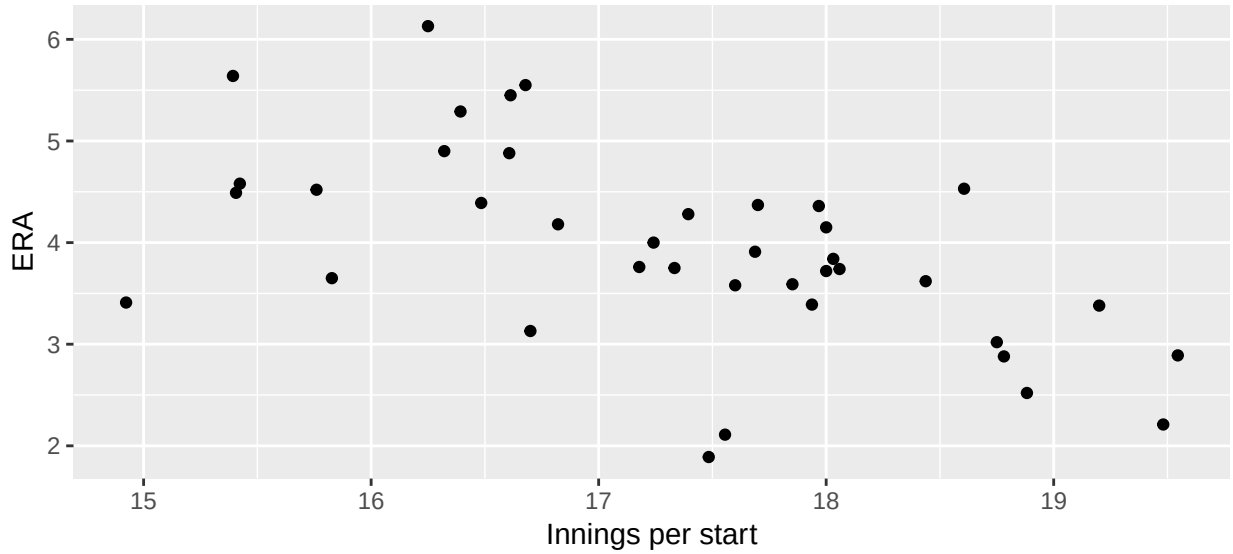
```
top.starters %>%
  filter(yearID == 2018, lgID == "AL") %>%
  ggplot(aes(y = IPouts/GS, x = teamID)) +
  geom_boxplot() +
  labs(y = "Innings per start")
```



Quantitative Variables

Let's check to ensure better pitchers are throwing more innings per start.

```
top.starters %>%
  filter(yearID == 2018, lgID == "AL") %>%
  ggplot(aes(x = IPouts/GS, y = ERA)) +
  geom_point() +
  labs(x = "Innings per start")
```



Let's look at trends over time.

```
top.starters %>%
  filter(teamID %in% c("BOS", "NYA", "TBA", "TOR", "BAL")) %>%
  group_by(yearID, teamID) %>%
  summarize(innings.per.start = sum(IPouts)/sum(GS)) %>%
  ggplot(aes(x = yearID, y = innings.per.start,
             color = teamID)) +
  geom_line()
```