Communicating Variation

Dr Austin R Brown

Kennesaw State University

Introduction

- One of the fundamental concepts in data science and statistics is that of variability.
- For instance, not every Major League Baseball team wins the same number of games during a given regular season. The value, total number of wins in this case, varies from team to team.
- A goal in data science and statistics is to identify assignable reasons (not necessarily causal reasons) which can help us explain the variability we are observing.
  - We generally call this “modeling”

Introduction

- However, the conclusions drawn from analytical models of whatever type may not necessarily be easily disseminated to a broad audience.

- This is to say, visualizations may be effective tools we can use to visually communicate variability.

- As usual, let's start with a research scenario and data source before moving into the specific type of visualizations we have available.

Introduction

- Suppose I want to see how the average number of strikeouts per nine innings pitched (SO/9) MLB pitchers have thrown has changed, if at all, from the 1910 season to the 2023 season.
    - Let's also focus on pitchers who have thrown more than 15 innings to avoid including position players who may occassionally be called on to pitch in lopsided games.
- We can make use of the `Lahman` package to help us get the right data into the right format.

Introduction

- First, we need to subset the full `Lahman::Pitching` dataframe to only include the years between (and inclusive of) 1910 and 2020 and the players who pitched at minimum 15 innings:

```
library(tidyverse)
so9 <- Lahman::Pitching |>
  filter(between(yearID,1910,2023) & (IPouts/3) >= 15)
```

Introduction

- Then, we need to create a new variable called “SO9” to calculate the number of strikeouts a given pitcher had during a given season (note, the variable `IPouts` is the number of innings pitched times 3):

```r
so9 <- Lahman::Pitching |>
  filter(between(yearID,1910,2023) & (IPouts/3) >= 15) |>
  mutate(SO9 = SO/(IPouts/3)*9)
```

Communicating Variation: Time Series Plot

- Now that we have the SO9 value for each pitcher for all of the included seasons, we need to decide exactly how this information is going to be visualized.

- One particular way may be by plotting a time series graph, where we simply plot the individual mean values for each year and connect them with a line.
  - This is a very common type of visualization for financial data.

- To do so, we need to aggregate out present dataframe to find the mean SO9 value for each season.

Communicating Variation: Time Series Plot

```r
so9 <- Lahman::Pitching |>
  filter(between(yearID,1910,2023) & (IPouts/3) >= 15) |>
  mutate(SO9 = SO/(IPouts/3)*9) |>
  group_by(yearID) |>
  summarize(Mean_SO9 = mean(SO9,na.rm=T))
```

Communicating Variation: Time Series Plot

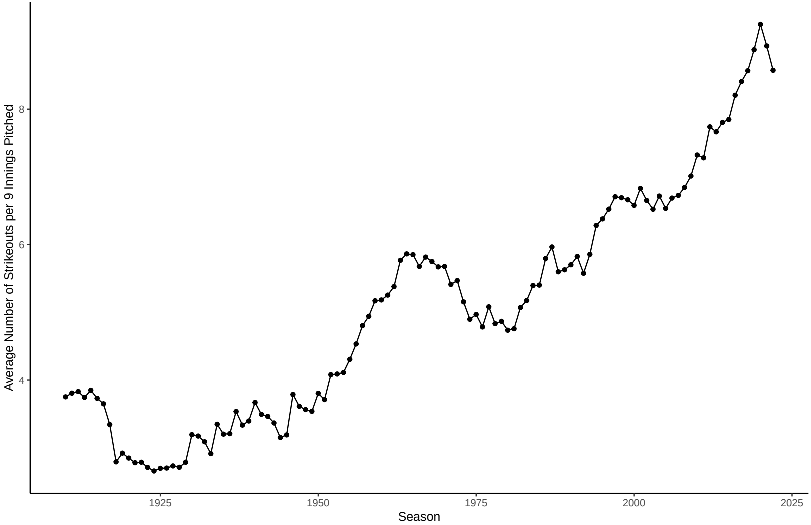
- Okay, great! Now we can make use of a new geom, called `geom_line`, to generate the time series plot:

```
so9 |>
  ggplot(aes(x=yearID,y=Mean_SO9)) +
  geom_point() +
  geom_line() +
  labs(x="Season",
       y="Average Number of Strikeouts per 9 Innings Pitched",
       title="Changes in Strikeouts per 9 Innings Pitched",
       subtitle="Between the 1910 & 2020 Regular Seasons") +
  theme_classic()
```

Communicating Variation: Time Series Plot

Changes in Strikeouts per 9 Innings Pitched

Between the 1910 & 2020 Regular Seasons



Communicating Variation: Time Series Plot

- As we can see, SO/9 has steadily increased over time with notable decreases during the first World War/Flu Pandemic and right around baseball becoming integrated.

- The results make sense: pitching has become much more specialized over time and of course, the athletes are also training much differently than they used to.

- But what is a primary limitation of this plot?

Communicating Variation: Time Series Plot

- With this plot, we can see the variation over time. But what we cannot observe is the within year variation.
    - And consequently, we can't see how the within year variation changes over time.
- This may be useful to know! As previously mentioned, this type of information is very commonly used in visualizations of financial or stock data.
- For us to include this information in this given plot, we have a few different strategies we can employ.

Communicating Variation: Time Series Plot with Standard Errors

- First, we can plot not just the mean value, but also the standard errors of the means, which is method by which variation can be quantified.

- To do this, we will reaggregate our data now using the `rstatix` package in order to obtain the standard errors of the means.

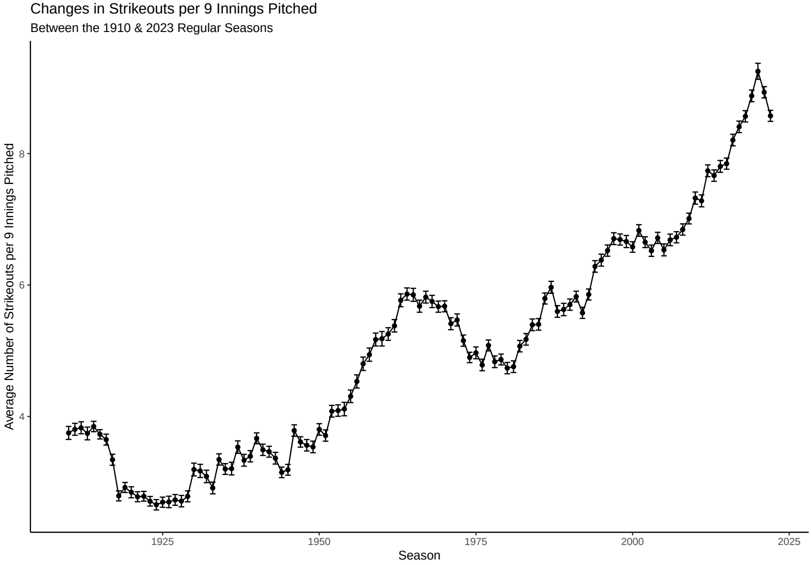
Communicating Variation: Time Series Plot with Standard Errors

```r
library(rstatix)
so91 <- Lahman::Pitching |>
  filter(between(yearID,1910,2023) & (IPouts/3) >= 15) |>
  mutate(SO9 = SO/(IPouts/3)*9) |>
  group_by(yearID) |>
  get_summary_stats(SO9,type='full')
```

Communicating Variation: Time Series Plot with Standard Errors

```r
so91 |>
  ggplot(aes(x=yearID,y=mean)) +
  geom_point() +
  geom_errorbar(aes(ymin=mean-se,ymax=mean+se)) +
  geom_line() +
  labs(x="Season",
       y="Average Number of Strikeouts per 9 Innings Pitched",
       title="Changes in Strikeouts per 9 Innings Pitched",
       subtitle="Between the 1910 & 2023 Regular Seasons") +
  theme_classic()
```

Communicating Variation: Time Series Plot with Standard Errors



Communicating Variation: Time Series Plot with Standard Errors

- What's the problem with this graph?
- Primarily, because we have so many seasons, and so many pitchers within a given season, the standard errors are small. Recall, the standard error of the sample mean is:

$$SE[\bar{x}] = \frac{s}{\sqrt{n}}$$

- So as the sample size, n, grows, the standard errors get small. And as you can see in the `so91` dataframe, every season, hundreds of pitchers pitch.

Communicating Variation: Time Series Plot with Boxplots

- To get around this issue, it may be preferable to use boxplots rather than standard errors to represent the within year variation.

- However, remember that to visualize boxplots, the data must be in its raw form. Both of our datasets are aggregated data:

```
so92 <- Lahman::Pitching |>
  filter(between(yearID,1910,2023) & (IPouts/3) >= 15) |>
  mutate(SO9 = SO/(IPouts/3)*9)
```

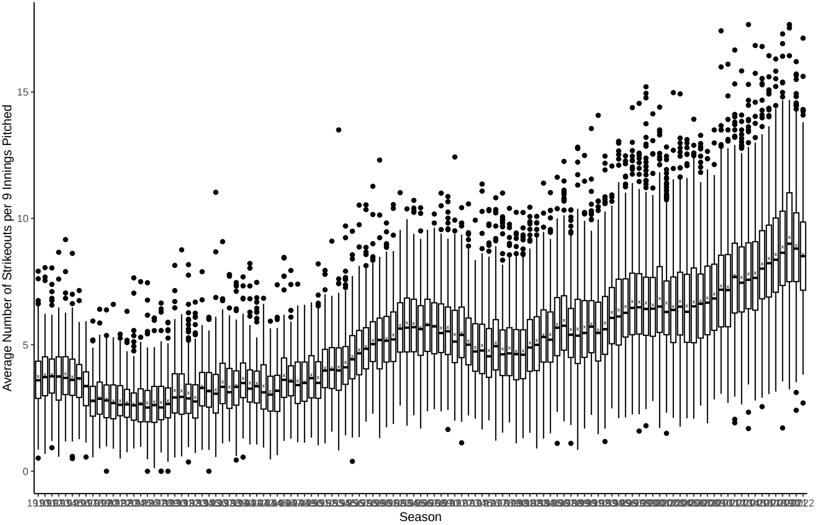
Communicating Variation: Time Series Plot with Boxplots

```r
so92 |>
  ggplot(aes(x=factor(yearID),y=SO9)) +
  geom_boxplot(fill='white',color='black') +
  geom_point(data=so91,
             aes(x=factor(yearID),
                 y=mean),shape='x') +
  labs(x="Season",
       y="Average Number of Strikeouts per 9 Innings Pitched",
       title="Changes in Strikeouts per 9 Innings Pitched",
       subtitle="Between the 1910 & 2023 Regular Seasons") +
  theme_classic()
```

Communicating Variation: Time Series Plot with Boxplots

Changes in Strikeouts per 9 Innings Pitched

Between the 1910 & 2023 Regular Seasons



Communicating Variation: Time Series Plot with Boxplots

- Okay we're getting closer! Here, we can see that not only is the mean SO/9 increasing over time, but so too is the variability from year to year.

- We can see this since the difference between the top of each box and the bottom of each box, the 75th and 25th percentiles, respectively, tends to widen as we move from left to right across the graphic.

- But notice what's going on the x-axis. The tick mark labels are all overlapping to the point where they are completely illegible.

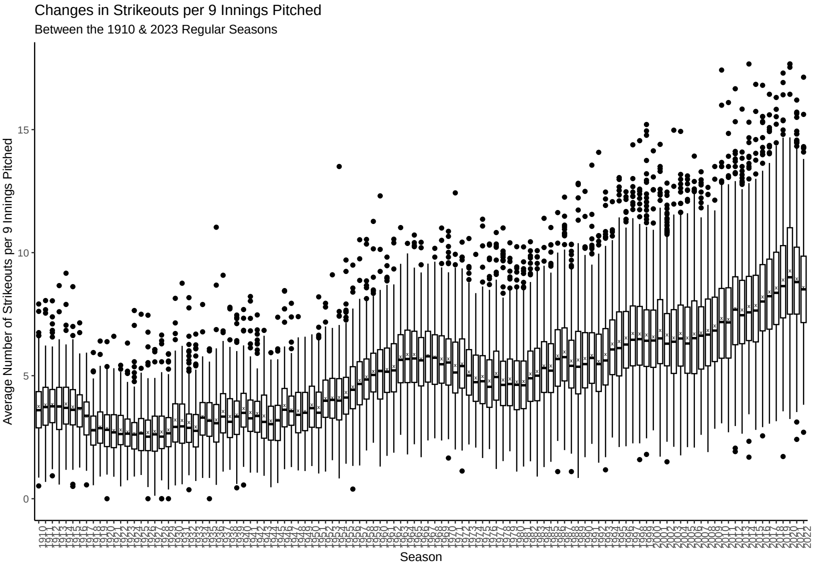
- Fortunately, we have learned a few tricks to modify the tick marks on either the x or y axis.

Communicating Variation: Time Series Plot with Boxplots

- One trick people sometimes use is changing the orientation or angle of the tick marks:

```
so92 |>
  ggplot(aes(x=factor(yearID),y=SO9)) +
  geom_boxplot(fill='white',color='black') +
  geom_point(data=so91,
             aes(x=factor(yearID),
                 y=mean),shape='x') +
  labs(x="Season",
       y="Average Number of Strikeouts per 9 Innings Pitched",
       title="Changes in Strikeouts per 9 Innings Pitched",
       subtitle="Between the 1910 & 2023 Regular Seasons") +
  theme_classic() +
  theme(axis.text.x=element_text(angle=90))
```

Communicating Variation: Time Series Plot with Boxplots



Changes in Strikeouts per 9 Innings Pitched

Between the 1910 & 2023 Regular Seasons

Communicating Variation: Time Series Plot with Boxplots

- In this case, that was helpful to eliminate the complete overlap of the labels, but now the problem is that there are too many labels.
    - It is still difficult to make out the individual digits for each year.

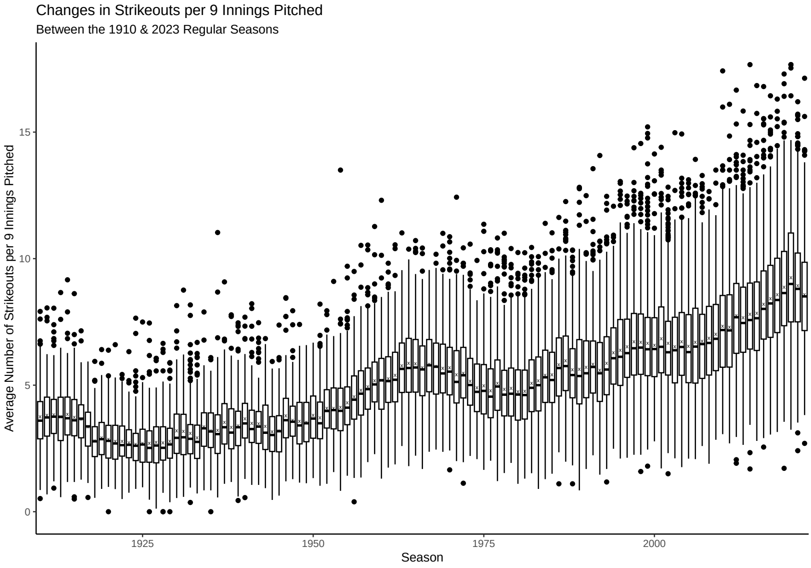
- What if we instead removed some of the labels to have the same number as we had with the regular time series plot (i.e., 1925, 1950, etc)

- Let's see how that can be achieved!

Communicating Variation: Time Series Plot with Boxplots

```r
so92 |>
  ggplot(aes(x=factor(yearID),y=SO9)) +
  geom_boxplot(fill='white',color='black') +
  geom_point(data=so91,
             aes(x=factor(yearID),
                 y=mean),shape='x') +
  labs(x="Season",
       y="Average Number of Strikeouts per 9 Innings Pitched",
       title="Changes in Strikeouts per 9 Innings Pitched",
       subtitle="Between the 1910 & 2023 Regular Seasons") +
  theme_classic() +
  scale_x_discrete(breaks=seq(1925,2023,by=25))
```

Communicating Variation: Time Series Plot with Boxplots


Changes in Strikeouts per 9 Innings Pitched
Between the 1910 & 2023 Regular Seasons
Average Number of Strikeouts per 9 Innings Pitched
Season