06 - Data Visualization (Part II)

Data and Information Engineering

SYS 2202 | Fall 2019

06-dataviz2.pdf

Contents

1 Cleveland Dot Plot **2**

1.1 Baseball Team Stats 2

1.2 Analyzing Hits 2

1.3 Cleveland Dot Plot Aesthetics 4

2 Line Graphs **5**

2.1 `economics` data 5

2.2 Your Turn: Stock Price 8

Required Packages and Data

```r
library(tidyquant)  # may need to: install.packages("tidyquant")
library(Lahman)     # may need to: install.packages("Lahman")
library(tidyverse)
```

1 Cleveland Dot Plot

William Cleveland wrote a popular book on visualizing data The Elements of Graphing Data that has many useful suggestions. One element he stressed was to reduce the cognitive strain on the view. One way to do this is to use as little ink as possible. The Cleveland dot plot contains the same information as a bar graph, but instead of using all the ink needed for the bar, remove the bar altogether and place a dot at the bar height (using `geom_point()`).

1.1 Baseball Team Stats

Consider the baseball dataset `Teams` from the `Lahman` package. This gives the team performance by year.

Your Turn #1 : Get Batting Data

Get the team performance for year (`yearID`) 2018 (Boston Red Sox beat the LA Dodgers in the World Series). Specifically,

- extract only the team name (`name`), league (`lgID`), wins (`W`), runs (`R`), at-bats (`AB`), hits (`H`), doubles (`X2B`), triples (`X3B`), home runs (`HR`), walks (`BB`);
- name the new object `bat18` for (batting 2018)

```
#> Error in select(., name, lgID, W, R:BB): unused arguments (name, lgID, W, R:BB)
```

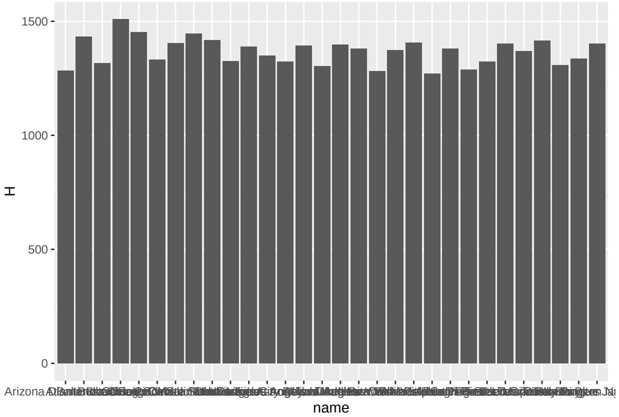
The first few rows should look like this:

name	lgID	W	R	AB	H	X2B	X3B	HR	BB	X1B
Arizona Diamondbacks	NL	82	693	5460	1283	259	50	176	560	798
Atlanta Braves	NL	90	759	5582	1433	314	29	175	511	915
Baltimore Orioles	AL	47	622	5507	1317	242	15	188	422	872
Boston Red Sox	AL	108	876	5623	1509	355	31	208	569	915
Chicago White Sox	AL	62	656	5523	1332	259	40	182	425	851
Chicago Cubs	NL	95	761	5624	1453	286	34	167	576	966

1.2 Analyzing Hits

Let's make the bar graph

```
ggplot(bat18) + geom_col(aes(x=name, y=H))
```

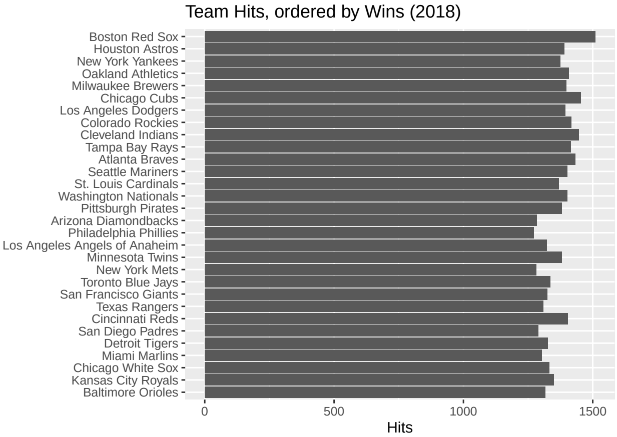


This isn't very revealing.

1. I can't see the team names
2. There should be some ordering of data.
   - ordering by Hits or Wins make more sense the the default (alphabetical)
3. Because the y-axis starts at 0, the differences between teams is not very apparent.

We can fix 1 and 2 very easily:

```r
ggplot(bat18) +
  geom_col(aes(x=reorder(name, W), y=H)) +
  labs(x='', y = 'Hits', title='Team Hits, ordered by Wins (2018)') +
  coord_flip()
```

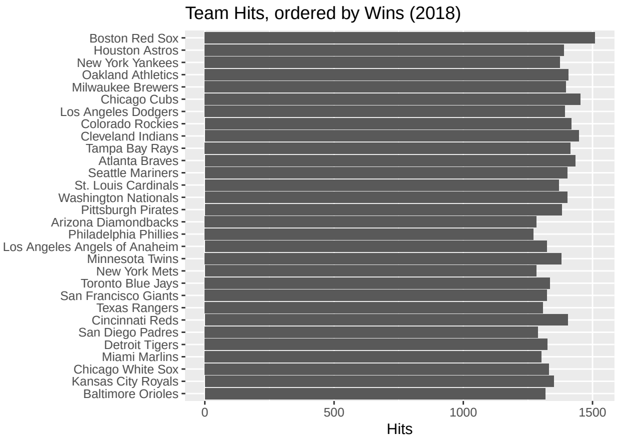


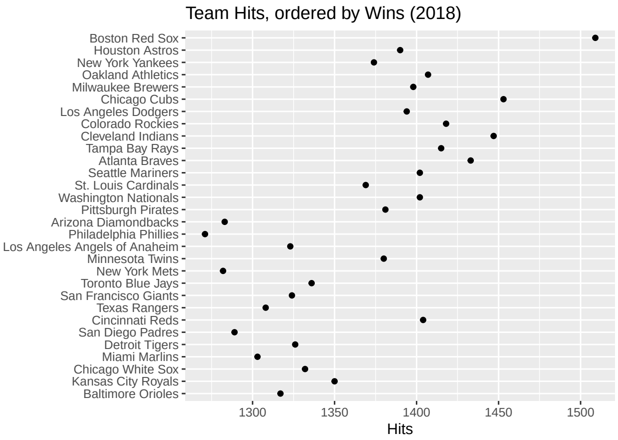
- The function `reorder()` convert a vector into a factor and orders it according to a function of a secondary variable. Above, we order the teams according to wins (`W`).
- The function `coord_flip()` swaps and x and y coordinates. Notice that the `labs()` arguments still correspond to the non-flipped axes.

Compare the bar graph with the dot plot.

```r
#- (left) bar graph
ggplot(bat18) +
  geom_col(aes(x=reorder(name, W), y=H)) +
  labs(x='', y = 'Hits', title='Team Hits, ordered by Wins (2018)') +
  coord_flip()

#- (right) corresponding dot plot
ggplot(bat18) +
  geom_point(aes(x=reorder(name, W), y=H)) +
  labs(x='', y = 'Hits', title='Team Hits, ordered by Wins (2018)') +
  coord_flip()
```





1.2.1 Your Turn

Your Turn #2 : Dot Plot vs. Bar Plot

1. What was changed in the code to make the Cleveland Dot Plot?
2. What are the differences between the two plots?
3. How would you add information about team homeruns to the bar plot? How about to the dot plot?

1.3 Cleveland Dot Plot Aesthetics

The real strength Cleveland's dotplot is in the ability to add additional aesthetics, like size, color, shape.

Your Turn #3 : Dressing up, Cleveland Style

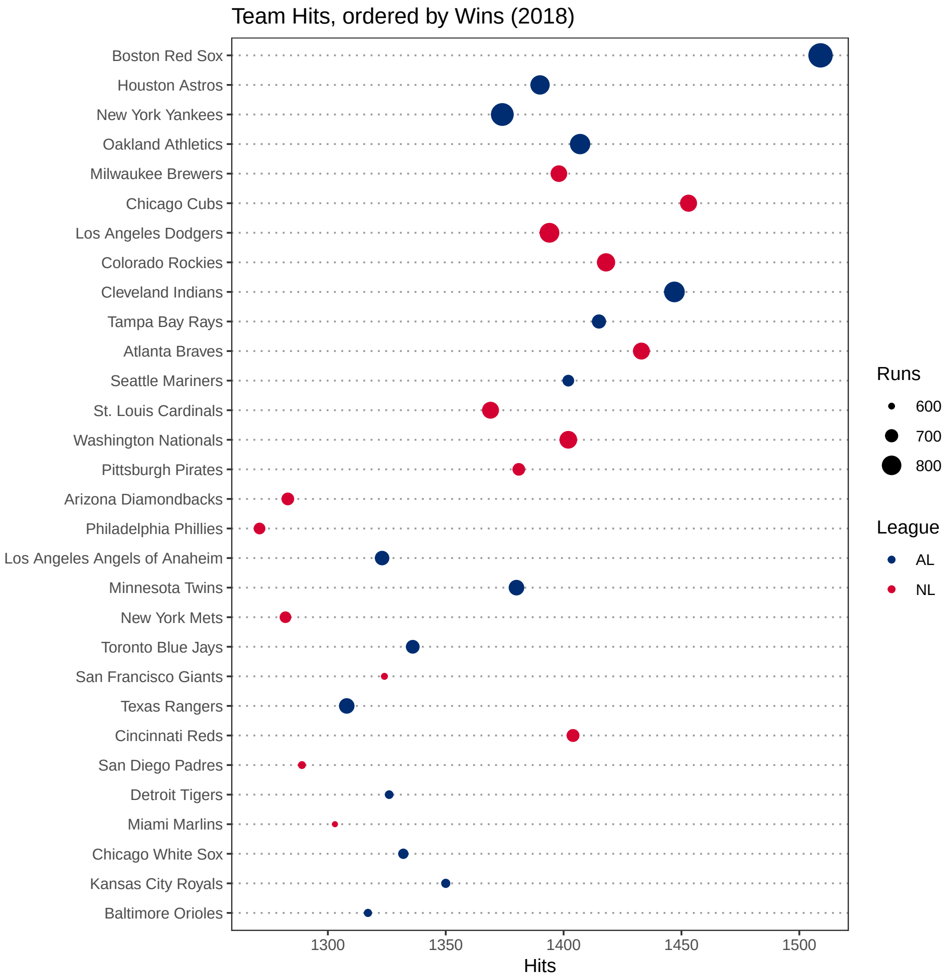
Modify the dot plot by adding the following:

1. Size the dots by runs (`R`)
2. Color the dots by league (`lgID`)

Final touches include changing the *theme*, modifying the colors and sizes

```
#- new theme
dot_theme = theme_bw() +
            theme(panel.grid.major.x=element_blank(),
                  panel.grid.minor.x=element_blank(),
                  panel.grid.major.y=element_line(color="grey60",
                                                  linetype="dotted"))

#- Cleveland dot plot
ggplot(bat18) +
  geom_point(aes(x=reorder(name, W), y=H, size=R, color=lgID)) +
  labs(x='', y = 'Hits', title='Team Hits, ordered by Wins (2018)') +
  coord_flip() +
  dot_theme +
  scale_color_manual(name="League", values=c("#002D72", "#D50032")) +
  scale_radius(name="Runs", range=c(1,6))
```



The *Cleveland* Dot Plot is an alternative to a bar plot. There is also a dot plot (geom_dotplot()) that is an alternative to a histogram.

2 Line Graphs

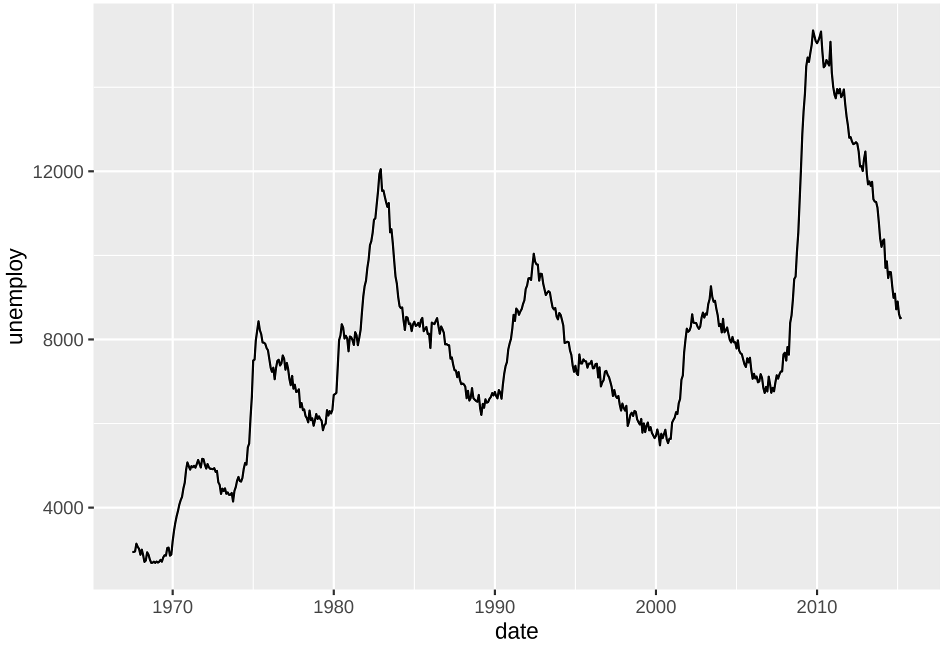
2.1 economics data

The economics data from the ggplot2 package contains some economic time series data

```
library(tidyverse)
data(economics)   # from the ggplot2 package (part of tidyverse package)
glimpse(economics)
#> Observations: 574
#> Variables: 6
#> $ date     <date> 1967-07-01, 1967-08-01, 1967-09-01, 1967-10-01, 1967...
#> $ pce      <dbl> 507, 510, 516, 512, 517, 525, 531, 534, 544, 544, 550...
#> $ pop      <dbl> 198712, 198911, 199113, 199311, 199498, 199657, 19980...
#> $ psavert  <dbl> 12.6, 12.6, 11.9, 12.9, 12.8, 11.8, 11.7, 12.3, 11.7,...
#> $ uempmed  <dbl> 4.5, 4.7, 4.6, 4.9, 4.7, 4.8, 5.1, 4.5, 4.1, 4.6, 4.4...
#> $ unemploy <dbl> 2944, 2945, 2958, 3143, 3066, 3018, 2878, 3001, 2877,...
```

We can plot the number of unemployed over time with a line plot (using `geom_line()`)

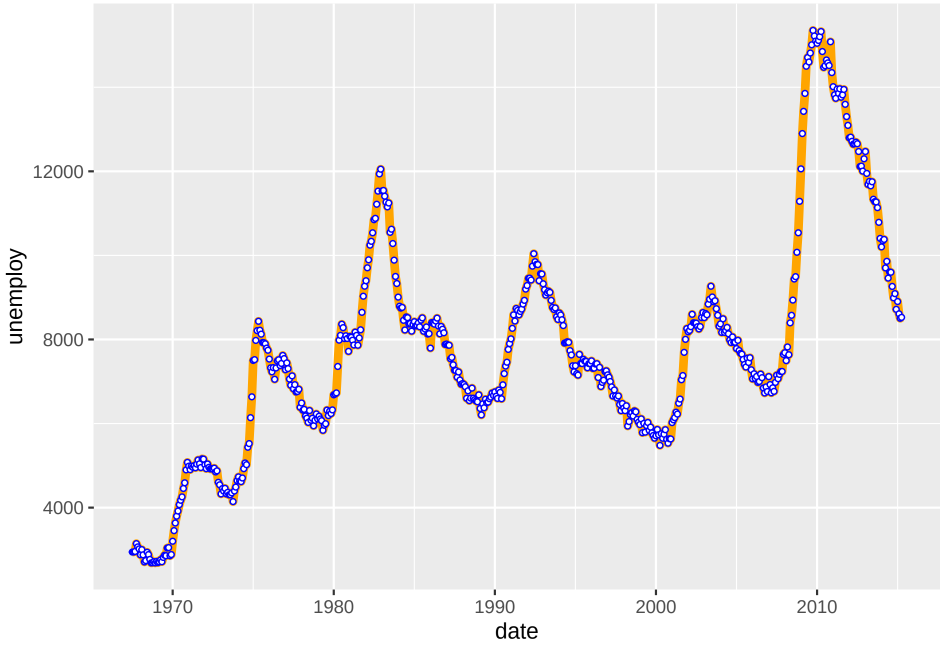
```
ggplot(economics, aes(date, unemploy)) + geom_line()
```



`ggplot()` recognizes the date class and smartly adds yearly tick marks!

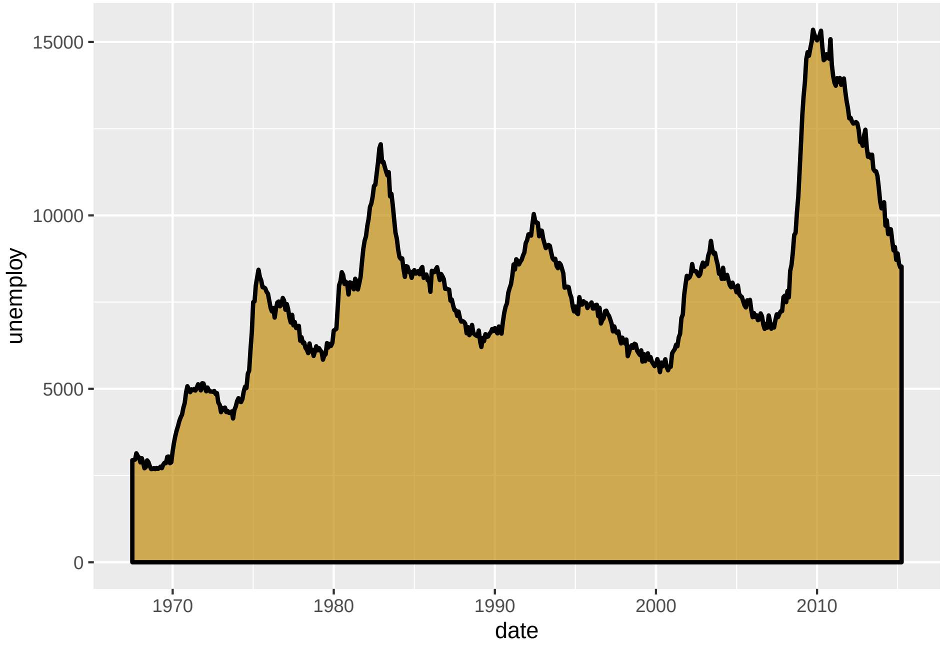
We can fancy it up, maybe add some points

```
ggplot(economics, aes(date, unemploy)) +
  geom_line(size=2, color="orange") +
  geom_point(shape=21, color='blue', fill='white', size= 1)
```



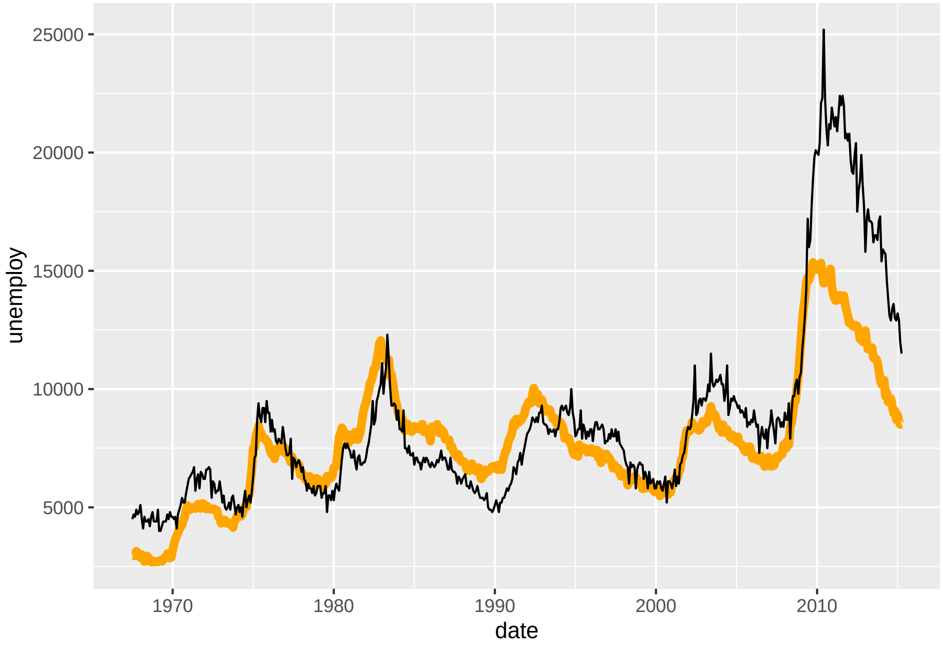
We can shade the region under the line with `geom_area()`

```
ggplot(economics, aes(date, unemploy)) +
  geom_area(color='black', fill='#C28E0E', alpha=.7, size=1)   # Go Boilers!
```



Multiple lines (using another aesthetic mapping for second line)

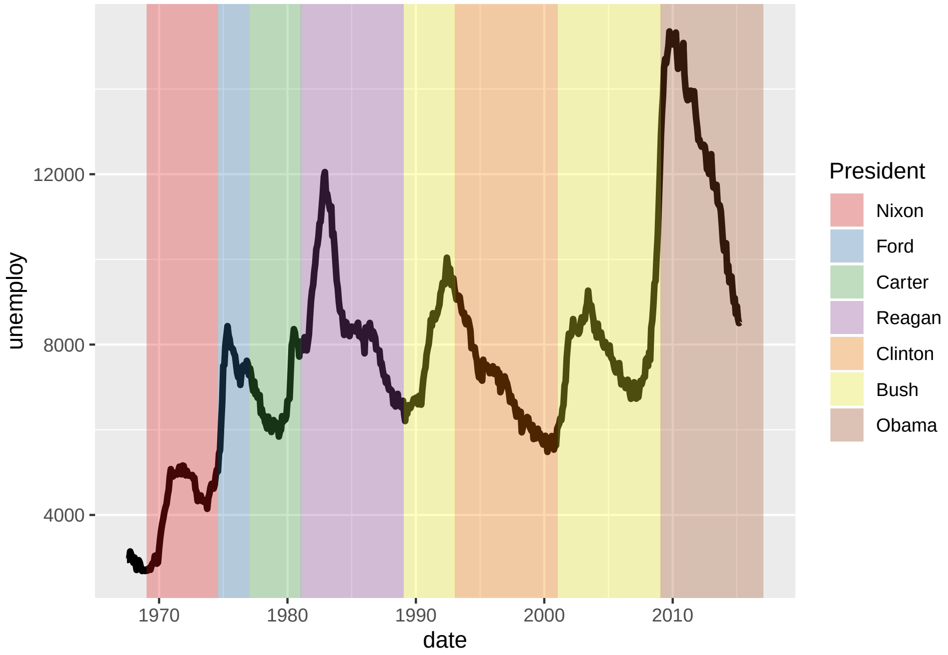
```
ggplot(economics, aes(date, unemploy)) +
  geom_line(size=2, color="orange") +     # uses y= number of unemployed
  geom_line(aes(date, uempmed*1000 ))     # uses y=1000* median duration of unemployment
```



How did the economy do for the presidents? Let's use the `presidential` data from `ggplot2` and use `geom_rect()` to shade in the time period for each president

```
data(presidential)     # load the presidential data (from ggplot2/tidyverse)

ggplot(economics) +
  geom_line(aes(date, unemploy),size=1.5, color="black") +
  geom_rect(data=filter(presidential, start>as.Date("1969-01-01")),
            aes(xmin=start, xmax=end, ymin=-Inf, ymax=Inf, fill=reorder(name, start)),
            alpha=.3) +
 scale_fill_brewer(palette="Set1", name="President")
```



2.2 Your Turn: Stock Price

Your Turn #4 : Stock Price

This exercise will walk you through a simple way to plot stock data.

1. The R package `tidyquant` provides quick access to daily stock price data. Install and load this package.
2. Get the Netflix (NFLX) stock data for 2018 - present using the `td_get()` function.

```
library(tidyquant)    # may need to install it first
NFLX = tq_get("NFLX", from = "2018-01-01", to = today()) # nifty today() function
```

3. Examine the data, then create a line plot of the `close` price by `date`. Color the line darkgreen.
4. Use `geom_area()` to fill the area below the line with lightgreen.