Lecture 6

Merging

```
attendance <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data
standings <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/
  rename(year_of_standing = year, name = team_name)
```

Goal of today will be to do the following: make a graph of attendance on standings. In essence I want to know whether good teams have better attendance than bad teams, or if there it is simply a "branding" issue: do good brand-name teams have good attendance just because of the brand?

To do this, I'm going to need to have attendance, and the standings data merged together. I purposed renamed some columns so it wouldn't be so obvious how to do this.

We are going to be concerned with the following:

- `left_join`. Pull up documentation

This is actually a very tricky problem: one of our data sets is a lot bigger than the other. Let's take a look at how to uniquely identify them. Unique identification is key to matching data correctly. You need to make sure that your observations are in the same "levels" so to speak. Let's do a naive merge.

```
attendance %>%
  left_join(standings)
```

Way too many rows. This doesn't make sense. This is because of the documentation warning that we get (see VALUE in the documentation).

Before we can merge, we need to understand how our data is leveled. More specifically, we need to know how the data is uniquely identified in each of the data sets, so that we can safely merge.

```
attendance %>%
  count(team, year, team_name) %>%
  filter(n >2)
```

Team does not uniquely identify obviously - there are many years for each team. Let's put in year. Almost. There are many years for each team though! What about name? Still no good. Look at the data and pause for people to know.

```
attendance %>%
  count(team, year, team_name, week) %>%
  filter(n >2)
```

Yes! This is *weekly* level data. Hence, once we subset down to the week level, our data is uniquely identified (no duplicates).

Now let's take a look at our standings data. Can you uniquely identify this data? Remember, you will want to do this with columns taht it has in common with the `attendance` data. Try it out yourself using our algorithm.

```
standings %>%
  count(team, name, year_of_standing) %>%
  filter(n >2)
```

So it looks like that the standings data is in yearly, while the attendance data is in weekly. If we want to make a graph, we have to somehow aggregate the attendance data to the same level as the standings data. Let's go ahead and do that. We'll use the mean to get it up to the yearly level. Hence, we'll have average yearly attendance.

```
yearly_attendance <- attendance %>%
  group_by(team, year, team_name) %>%
  summarize(avg_attendance = mean(weekly_attendance, na.rm = T))
```

```
yearly_attendance %>%
  left_join(standings)
```

Still doesn't work right! We need to use the `by` argument:

```
football_attendence <- yearly_attendance %>%
  left_join(standings, by = c("team" = "team", "year" = "year_of_standing", "team_name" = "name"))
```

Now this works correctly. Notice how we are mergining on 3 different columns. It doesn't matter that the columnnames are different - we told R which columns align with which. Now let's go ahead and make a plot

Making Tables with Kable

```
football_attendence %>%
  group_by(team) %>%
  summarize(across(c(avg_attendance, margin_of_victory, strength_of_schedule, wins), ~mean(., na.rm = T
  arrange(desc(avg_attendance)) %>%
  kbl(col.names = c(" ","Attendance", "Maring of Victory", "Strength of Schedule", "Wins"),
      caption = "Average of Columns") %>%
  kable_paper() %>%
  column_spec(column = 1, color = "green") %>%
  row_spec(row = 1, bold = T)
```

Regressions

Now we're going to learn how to do linear regressions in R. There are several ways to do this, but I think the best way (if you're an econometrician) is to use the `fixest` package. I would type in "fixest vignette" on Google and read the document. It's good.

```
feols(hp ~ mpg, data = mtcars, se = "hetero")

## similarly
mtcars %>%
  feols(hp ~mpg, data = .)
```

```
titanic <- titanic::titanic_train %>%
  janitor::clean_names() %>%
  as_tibble()

titanic <- titanic %>%
  mutate(male = ifelse(sex == "male", 1, 0))

## fixed effects
survived_reg  <- titanic %>%
  feols(survived ~ fare + male + age | pclass, data = .)
```

I should also teach you how to make a regression using base R. To do this, you use the `lm` function which stands for "linear model"

```
titanic %>%
  lm(survived ~ fare + male + age + pclass, data = .) %>%
  summary()
```

Presentation

```
modelsummary(survived_reg, stars = T, gof_omit = "R2 Within|A|L|S|B",
             coef_rename = c("fare" = "Fare",
                             "male" = "Male",
                             "age" = "Age"),
             title = "Effect of fare, sex, and age on survival probability") %>%
  kable_paper()
```

You can also add multiple regressions:

```
survived_reg_2 <- titanic %>%
  feols(survived ~ fare + male + age|pclass + embarked, data = .)
```

```
reg_table <- modelsummary(list("(1)" = survived_reg, "(2)" = survived_reg_2),
             stars = T, gof_omit = "R2 Within|A|L|S|B",
             coef_rename = c("fare" = "Fare",
                             "male" = "Male",
                             "age" = "Age"),
             title = "Effect of fare, sex, and age on survival probability")
```