# Lab 6 Key

*Daniel Anderson*

1. Create a new R Markdown document and modify the YAML to

- Include your name
- Change the syntax highlighting to any scheme but the default. The options are `default`, `tango`, `pygments`, `kate`, `monochrome`, `espresso`, `zenburn`, `haddock`, and `textmate`. You can also use `NULL` if you want no highlighting.
- Include the option to make it easy to modify the rendering between PDF and HTML.

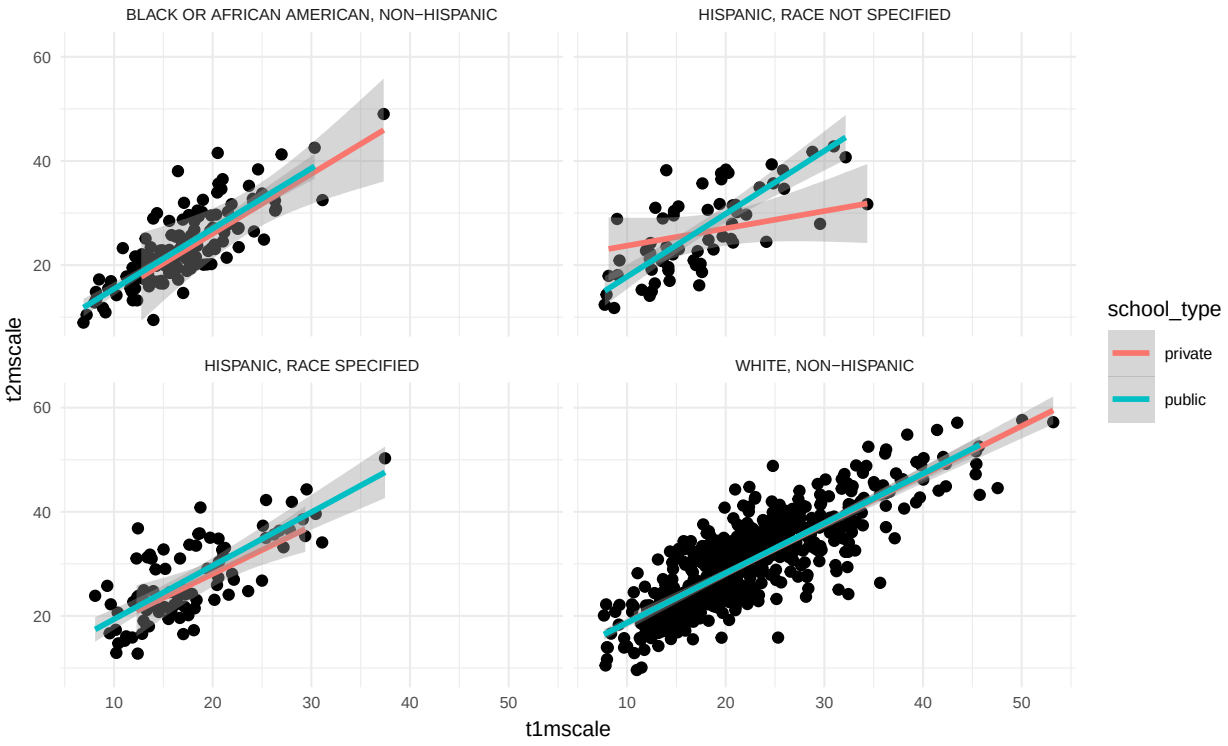
2. Create a code chunk that

- Does not display the code or any warnings, messages, etc. from the code, but evaluates every function/line of the code.
- Loads all the packages you decide to use for the lab
- Sets a global chunk option to make all figures 6.5" wide and the height to a value that makes sense to you.

3. Load the *ecls-k_samp.sav* dataset, and produce the plot below. Do not show the code you used (colors, themes, etc. don't matter here).

```
d <- import(here("data", "ecls-k_samp.sav"),
            setclass = "tbl_df") %>%
        characterize() %>%
        clean_names()

d %>%
  filter(ethnic == "WHITE, NON-HISPANIC" |
         ethnic == "BLACK OR AFRICAN AMERICAN, NON-HISPANIC" |
         ethnic == "HISPANIC, RACE SPECIFIED" |
         ethnic == "HISPANIC, RACE NOT SPECIFIED") %>%
ggplot(aes(t1mscale, t2mscale)) +
  geom_point() +
  geom_smooth(aes(color = school_type),
              method = "lm") +
  facet_wrap(~ethnic)
```


BLACK OR AFRICAN AMERICAN, NON−HISPANIC
HISPANIC, RACE NOT SPECIFIED
HISPANIC, RACE SPECIFIED
WHITE, NON−HISPANIC
t2mscale
t1mscale
60
40
20
10
20
30
40
50
school_type
private
public

4. Run The following lines of code to store the mean and standard deviation of `t1mscale`. Extend this code to calculate (in the same code chunk) the mean and standard deviation of `t2mscale`. Note this code assumes you read the ecls-k dataset in as an object called `d`. You should substitute in whatever the name is for your data object.

```
t1mean <- mean(d$t1mscale, na.rm = TRUE)
t1sd <- sd(d$t1mscale, na.rm = TRUE)

t2mean <- mean(d$t2mscale, na.rm = TRUE)
t2sd <- sd(d$t2mscale, na.rm = TRUE)
```

Using the values you calculated above, use an inline code evaluation to report the means/sds for the two time points. Also report the difference between the means (i.e., the average gain).

---

The mean for time point one was 20.51, with a standard deviation of 7.37. At time point two, the mean was 28.57, with a standard deviation of 8.78. The difference between the means was between the time points was thus 8.06 points.

---

5. Pretend you are trying to teach somebody how to load data. Describe the process below that we've discussed in class, including why it helps reproducibility, and echo chunks of code as necessary without actually evaluating any of it.

---

Create and RStudio project. This will serve as the root directory. Create empty *data* and *scripts* folders. Place any data you have in the data folder.

To import the data, create a new script, then load the following packages

```
library(rio)
library(here)
```

If you don't have these packages installed, you'll need to install them first, using `install.packages(c("rio", "here"))`.

Finally, use the {here} library to navigate file paths, and `rio::import` to import data of basically any type. For example, assuming you have created an RStudio project and you are working from that project, and you've put the data you want to load in at folder called *data*, any of the following should work

```
d <- import(here("data", "data_to_load.sav"),
            setclass = "tbl_df")

d <- import(here("data", "data_to_load.xlsx"),
            setclass = "tbl_df")

d <- import(here("data", "data_to_load.dta"),
            setclass = "tbl_df")
```

Note that the `setclass` argument is optional, and changes the import to a tibble instead of a standard data frame.