Homework 4: Data Import and Enhanced Graphics

Dr. D - Solutions

2026-01-05

```
library(dplyr)
library(ggplot2)
library(readxl)
```

Part I: Importing data

Go to Dr. D's data website to find the three data files used in the questions below. Download them to your `data` folder and import them in the first three questions below.

1. Read in the `High School and Beyond` data set. Use the R function `names` to get a list of variable names in this data set.

```
hsb2 <- read.delim(here::here("data/hsb2.txt"), header = TRUE, sep = "\t")
names(hsb2)
```

```
 [1] "id"      "gender"  "race"    "ses"     "schtyp"  "prog"    "read"
 [8] "write"   "math"    "science" "socst"
```

2. Read in the `Crime Data` data set. Use R functions to create a table that shows how many states are in each division (using the `division` variable).

```
crime <- read_excel(here::here("data/Crime_Data.xlsx"), sheet = 1, col_names = TRUE)
table(crime$division)
```

```
East North Central East South Central    Middle Atlantic           Mountain
                 5                    4                  3                  8
       New England            Pacific     South Atlantic West North Central
                 6                    5                  9                  7
West South Central
                 4
```

3. Read in the `countyComplete` data set.

```
county <- read.csv(here::here("data/countyComplete.csv"), header = TRUE, sep = ",")
```

4. When reading in the Excel data set on crime, what do the arguments `sheet=1` and `col_names=TRUE` mean?

Read from the first sheet and the first row contains column names.

5. Use the `str()` function to determine how many observations and variables are contained in the `countyComplete` data set. Write the answer in a complete sentence.

```
str(county)
```

```
'data.frame':   3116 obs. of  56 variables:
 $ name                                  : chr  "Autauga County" "Baldwin County" "Barbou
 $ state                                 : chr  "Alabama" "Alabama" "Alabama" "Alabama" .
 $ FIPS                                  : int  1001 1003 1005 1007 1009 1011 1013 1015 1
 $ pop2010                               : int  54571 182265 27457 22915 57322 10914 2094
 $ pop2000                               : int  43671 140415 29038 20826 51024 11714 2139
 $ age_under_5                           : num  6.6 6.1 6.2 6 6.3 6.8 6.5 6.1 5.7 5.3 ...
 $ age_under_18                          : num  26.8 23 21.9 22.7 24.6 22.3 24.1 22.9 22.
 $ age_over_65                           : num  12 16.8 14.2 12.7 14.7 13.5 16.7 14.3 16.
 $ female                                : num  51.3 51.1 46.9 46.3 50.5 45.8 53 51.8 52.
 $ white                                 : num  78.5 85.7 48 75.8 92.6 23 54.4 74.9 58.8 9
 $ black                                 : num  17.7 9.4 46.9 22 1.3 70.2 43.4 20.6 38.7 4
 $ native                                : num  0.4 0.7 0.4 0.3 0.5 0.2 0.3 0.5 0.2 0.5 .
 $ asian                                 : num  0.9 0.7 0.4 0.1 0.2 0.2 0.8 0.7 0.5 0.2 .
 $ pac_isl                               : num  NA NA NA NA NA NA 0 0.1 0 0 ...
 $ two_plus_races                        : num  1.6 1.5 0.9 0.9 1.2 0.8 0.8 1.7 1.1 1.5 .
 $ hispanic                              : num  2.4 4.4 5.1 1.8 8.1 7.1 0.9 3.3 1.6 1.2 .
 $ white_not_hispanic                    : num  77.2 83.5 46.8 75 88.9 21.9 54.1 73.6 58.
 $ no_move_in_one_plus_year              : num  86.3 83 83 90.5 87.2 88.5 92.8 82.9 86.2 8
 $ foreign_born                          : num  2 3.6 2.8 0.7 4.7 1.1 1.1 2.5 0.9 0.5 ...
 $ foreign_spoken_at_home                : num  3.7 5.5 4.7 1.5 7.2 3.8 1.6 4.5 1.6 1.4 .
```

```
$ hs_grad                                       : num  85.3 87.6 71.9 74.5 74.7 74.7 74.8 78.5 7
$ bachelors                                     : num  21.7 26.8 13.5 10 12.5 12 11 16.1 10.8 10
$ veterans                                      : int  5817 20396 2327 1883 4072 943 1675 11757 1
$ mean_work_travel                              : num  25.1 25.8 23.8 28.3 33.2 28.1 25.1 22.1 23
$ housing_units                                 : int  22135 104061 11829 8981 23887 4493 9964 53
$ home_ownership                                : num  77.5 76.7 68 82.9 82 76.9 69 70.7 71.4 77
$ housing_multi_unit                            : num  7.2 22.6 11.1 6.6 3.7 9.9 13.7 14.3 8.7 4
$ median_val_owner_occupied                     : num  133900 177200 88200 81200 113700 ...
$ households                                    : int  19718 69476 9795 7441 20605 3732 8019 4642
$ persons_per_household                         : num  2.7 2.5 2.52 3.02 2.73 2.85 2.58 2.46 2.5
$ per_capita_income                             : int  24568 26469 15875 19918 21070 20289 16916
$ median_household_income                       : int  53255 50147 33219 41770 45549 31602 30659
$ poverty                                       : num  10.6 12.2 25 12.6 13.4 25.3 25 19.5 20.3
$ private_nonfarm_establishments                : int  877 4812 522 318 749 120 446 2444 568 350
$ private_nonfarm_employment                    : int  10628 52233 7990 2927 6968 1919 5400 38324
$ percent_change_private_nonfarm_employment: num  16.6 17.4 -27 -14 -11.4 -18.5 2.1 -5.6 -45
$ nonemployment_establishments                  : int  2971 14175 1527 1192 3501 390 1180 6329 20
$ firms                                         : int  4067 19035 1667 1385 4458 417 1769 8713 19
$ black_owned_firms                             : num  15.2 2.7 NA 14.9 NA NA NA 7.2 NA NA ...
$ native_owned_firms                            : num  NA 0.4 NA NA NA NA NA NA NA NA ...
$ asian_owned_firms                             : num  1.3 1 NA NA NA NA 3.3 1.6 NA NA ...
$ pac_isl_owned_firms                           : num  NA NA NA NA NA NA NA NA NA NA ...
$ hispanic_owned_firms                          : num  0.7 1.3 NA NA NA NA NA 0.5 NA NA ...
$ women_owned_firms                             : num  31.7 27.3 27 NA 23.2 38.8 NA 24.7 29.3 14
$ manufacturer_shipments_2007                   : int  NA 1410273 NA 0 341544 NA 399132 2679991 6
$ mercent_whole_sales_2007                      : int  NA NA NA NA NA NA 56712 NA NA 62293 ...
$ sales                                         : int  598175 2966489 188337 124707 319700 43810
$ sales_per_capita                              : int  12003 17166 6334 5804 5622 3995 11326 1367
$ accommodation_food_service                    : int  88157 436955 NA 10757 20941 3670 28427 186
$ building_permits                              : int  191 696 10 8 18 1 3 107 10 6 ...
$ fed_spending                                  : int  331142 1119082 240308 163201 294114 108846
$ fed_spend00                                   : num  7.58 7.97 8.28 7.84 5.76 ...
$ fed_spend10                                   : num  6.07 6.14 8.75 7.12 5.13 ...
$ area                                          : num  594 1590 885 623 645 ...
$ density                                       : num  91.8 114.6 31 36.8 88.9 ...
$ smoking_ban                                   : chr  "none" "none" "partial" "none" ...
```

The `countyComplete` data frame contains 3116 observations (rows) and 56 variables (columns).

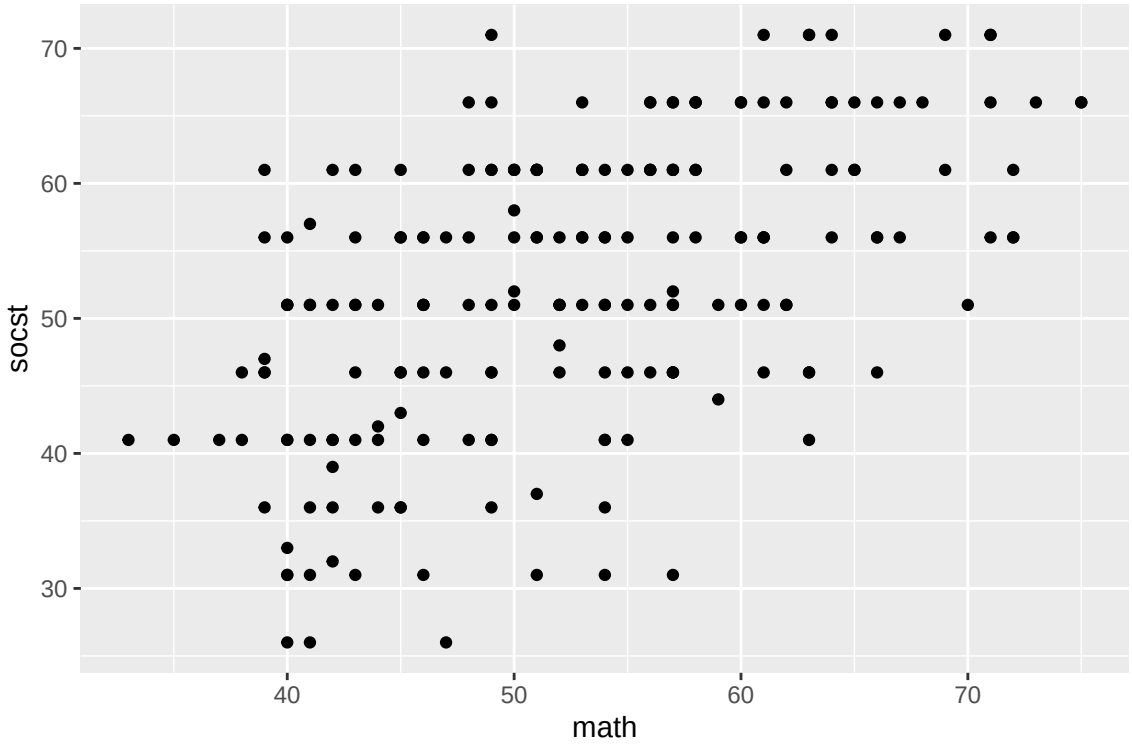
Part II: Improving your plots

This section demonstrates how you can build a nice graph in stages. Each question will have you adjust one thing on the plot. Instead of retyping the entire code each time, copy the code from the prior question and then make the requested adjustment. Small changes bit by bit helps you solidify learning what each piece of code does.

1. **Let's examine the relationship between math (`math`) and social studies (`socst`) scores using the HS & Beyond (`hsb2`) data set.**

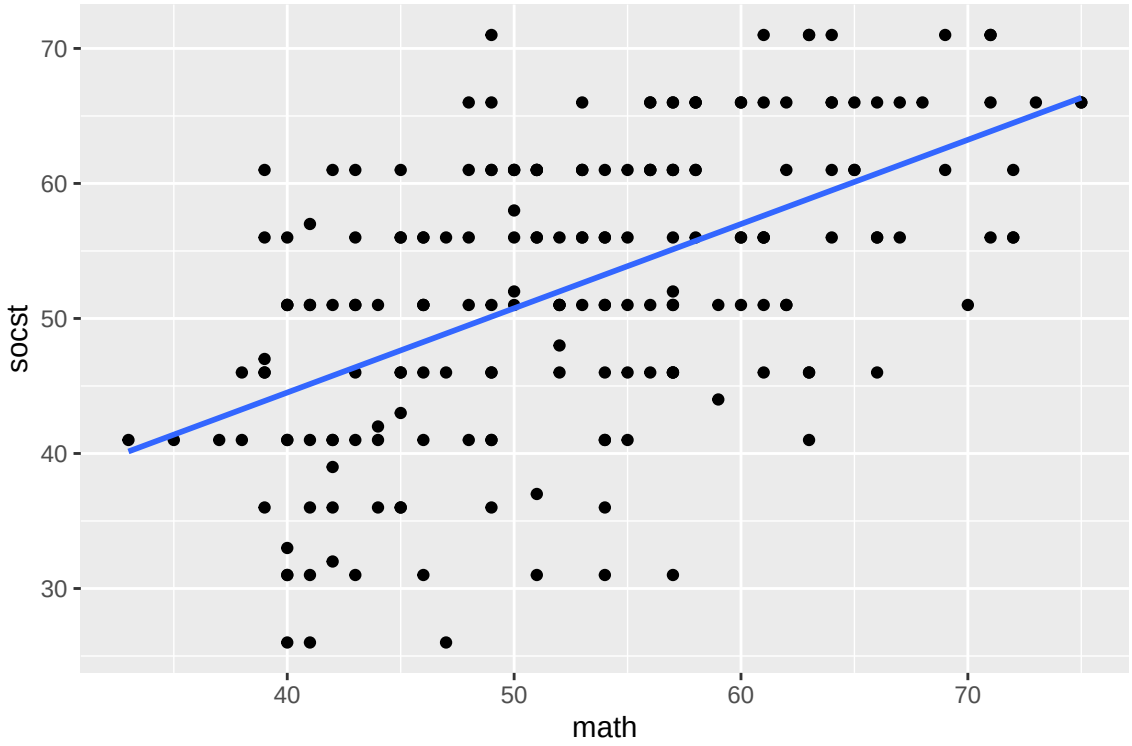
a. Start with a basic scatterplot. Create a scatterplot showing math score on the y-axis and social studies score on the x-axis.

```
ggplot(hsb2, aes(x = math, y = socst)) +
  geom_point()
```



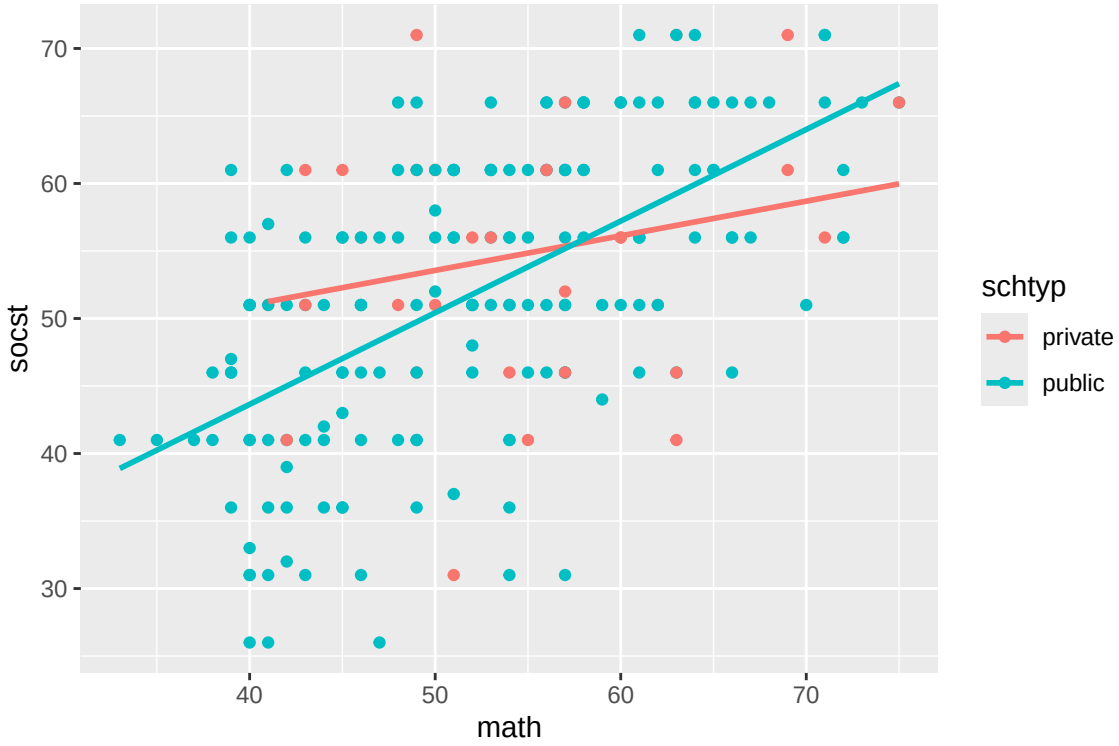
b. Add a trend line. Add a linear trend line to your plot to help clarify the overall relationship between math and social studies scores. Use a linear model line, but hide the confidence band.

```
ggplot(hsb2, aes(x = math, y = socst)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



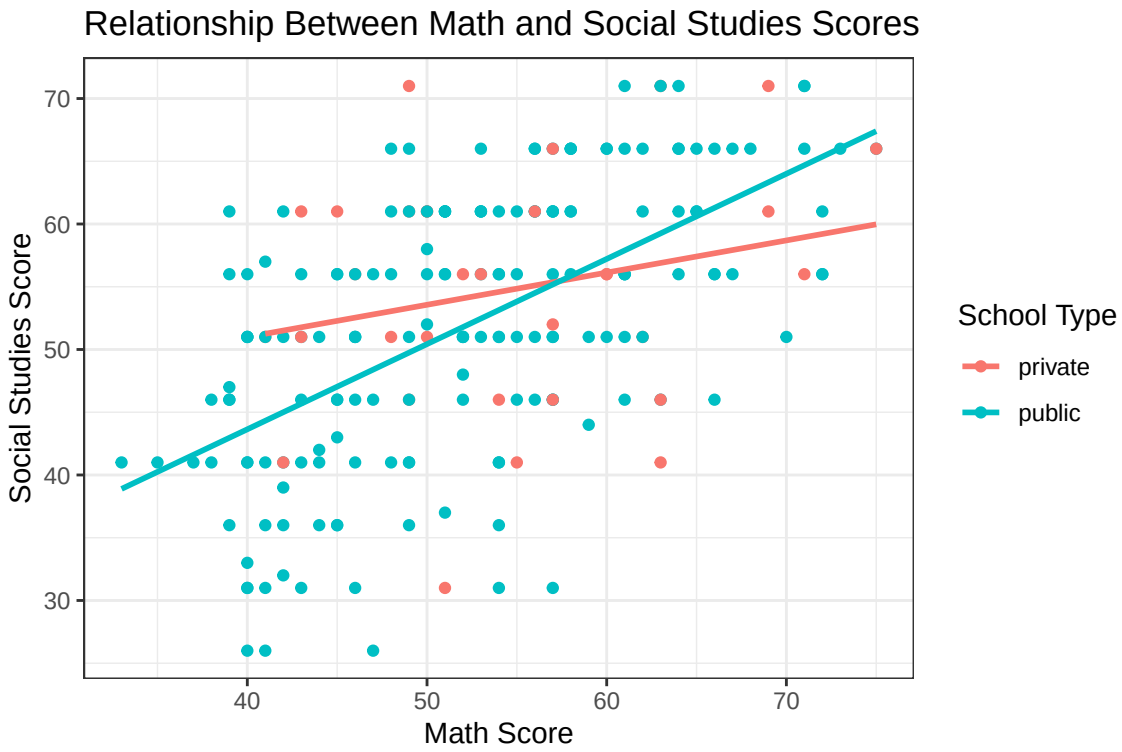
c. Group by school type. Check for a different relationship between math and social studies scores depending on school type. Color the points and lines by the `schtyp` variable. *hint: you're* ***not*** *adding code to the* `geom` *layers*

```
ggplot(hsb2, aes(x = math, y = socst, color = schtyp)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```



d. Improve communicability. Now refine the plot so that it clearly communicates results. Add a title, axis and legend labels, and apply a clean theme.

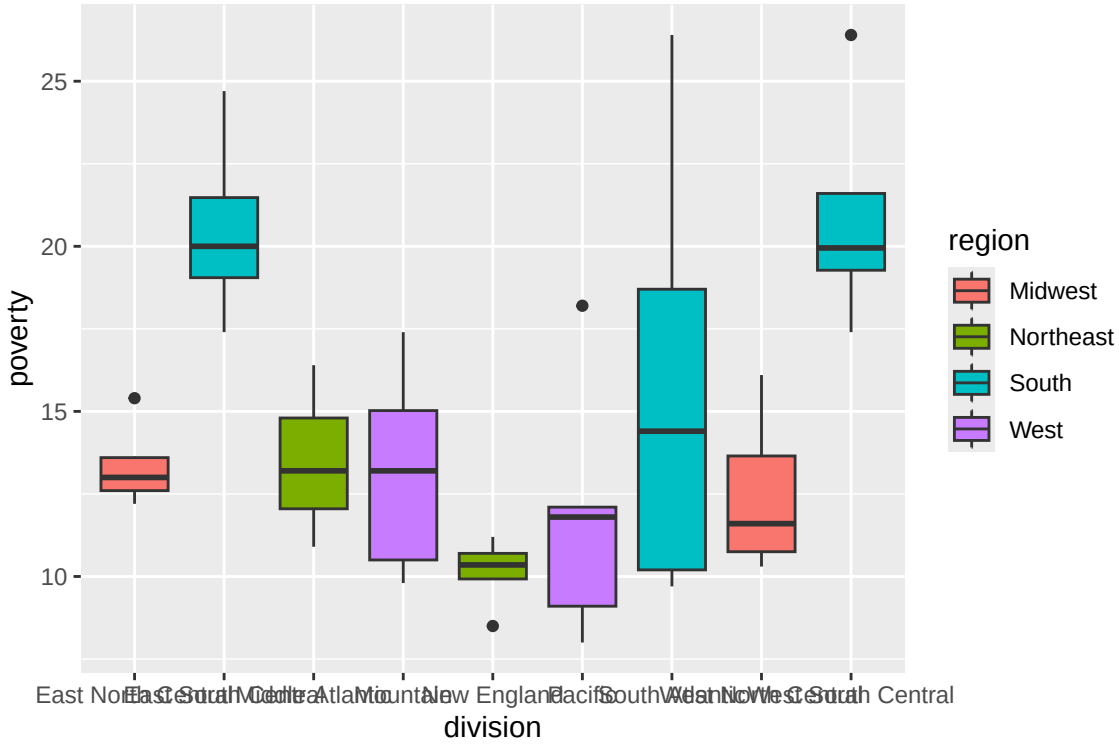
```
ggplot(hsb2, aes(x = math, y = socst, color = schtyp)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "Relationship Between Math and Social Studies Scores",
    x = "Math Score", y = "Social Studies Score") +
  scale_color_discrete(name="School Type") +
  theme_bw()
```



2. Let's explore the distribution of poverty rates in states around the USA.

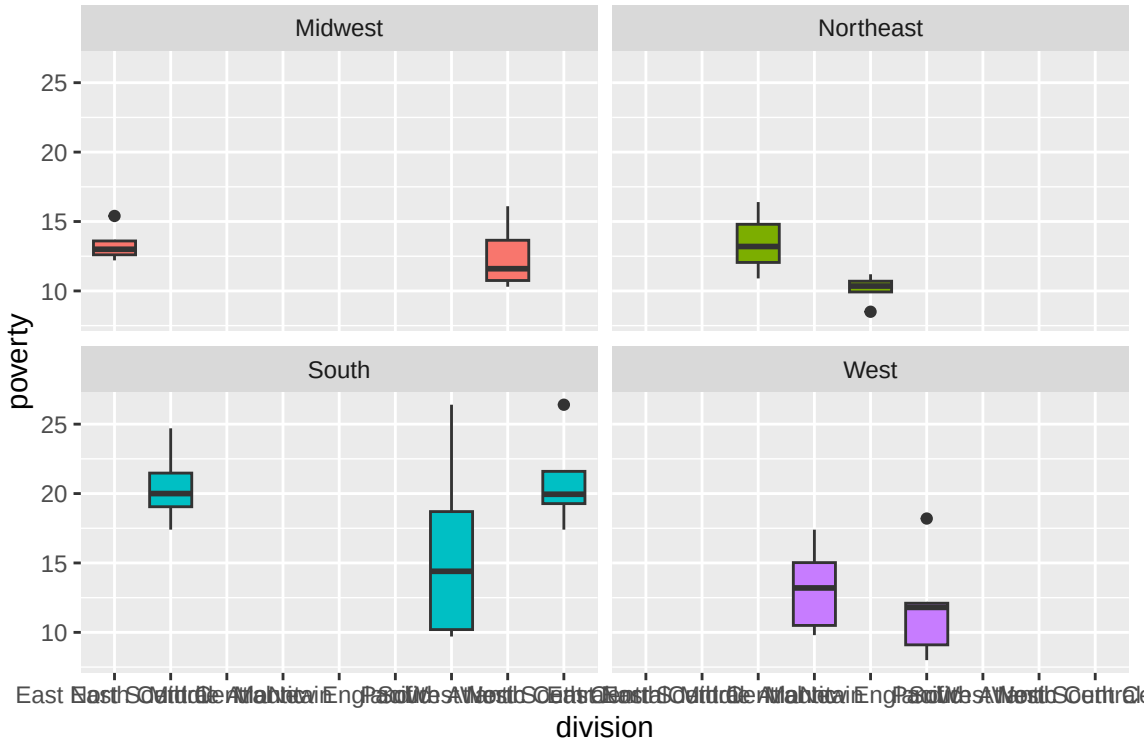
a. Using the `crime` data, create a boxplot of the distribution of `poverty` rate on the y axis, `division` on the x axis, and filling the boxes by `region`.

```
ggplot(crime, aes(y=poverty, x = division, fill = region)) +
  geom_boxplot()
```



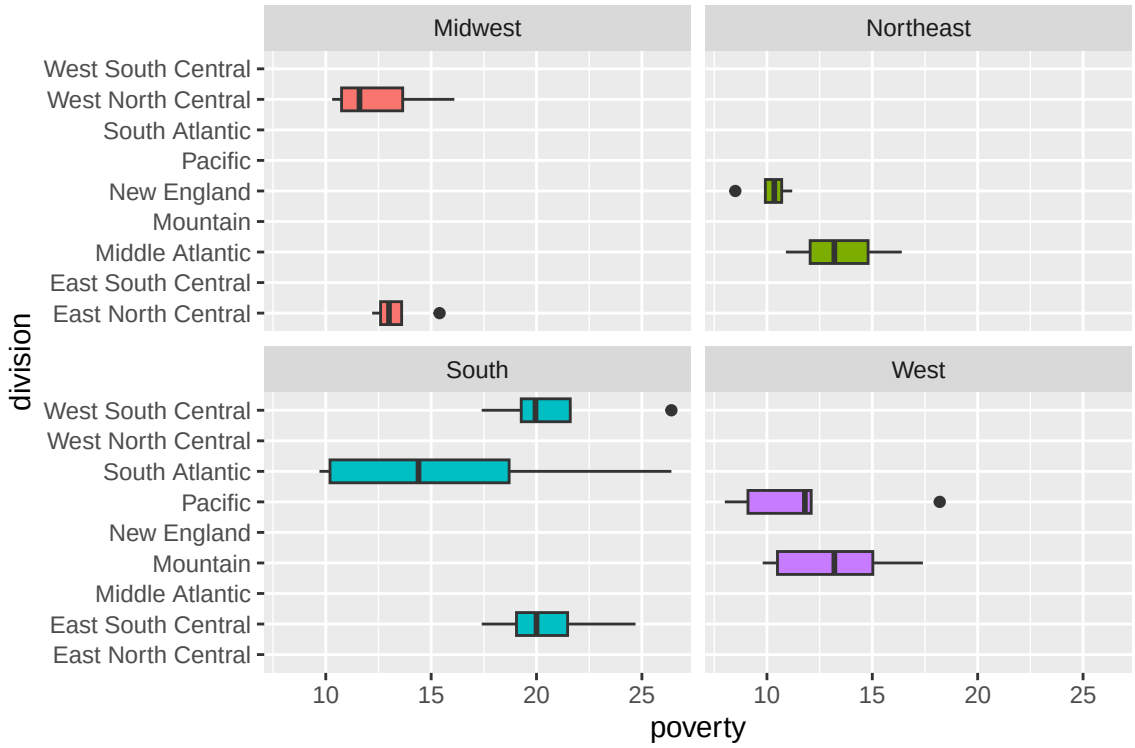
b. These are multiple divisions in a region, so let's facet on `region` so that each region is in it's own panel. Also get rid of the legend since the name of the region is in the panel already.

```
ggplot(crime, aes(y=poverty, x = division, fill = region)) +
  geom_boxplot() +
  facet_wrap(~region) +
  scale_fill_discrete(guide = "none")
```



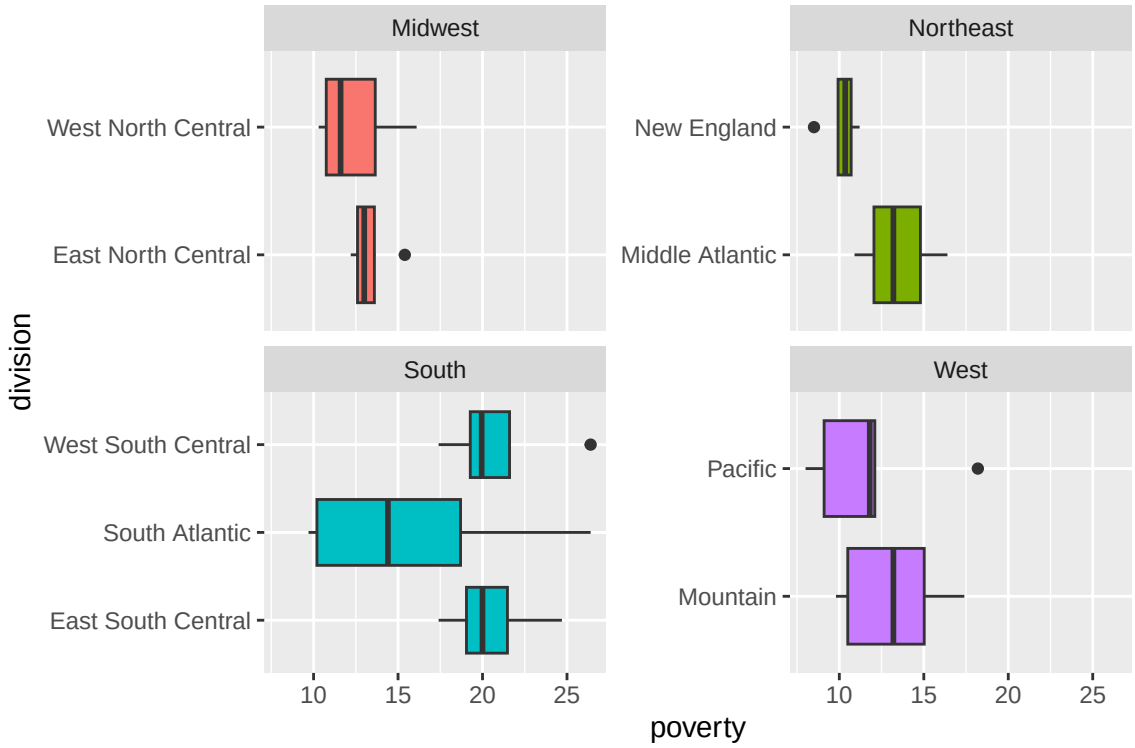
c. Improve readabilty. Let's try to fix the overlapping of the x axis labels using `coord_flip()` to rotate the plots horizontally.

```
ggplot(crime, aes(y=poverty, x = division, fill = region)) +
  geom_boxplot() +
  facet_wrap(~region) +
  scale_fill_discrete(guide = "none") +
  coord_flip()
```



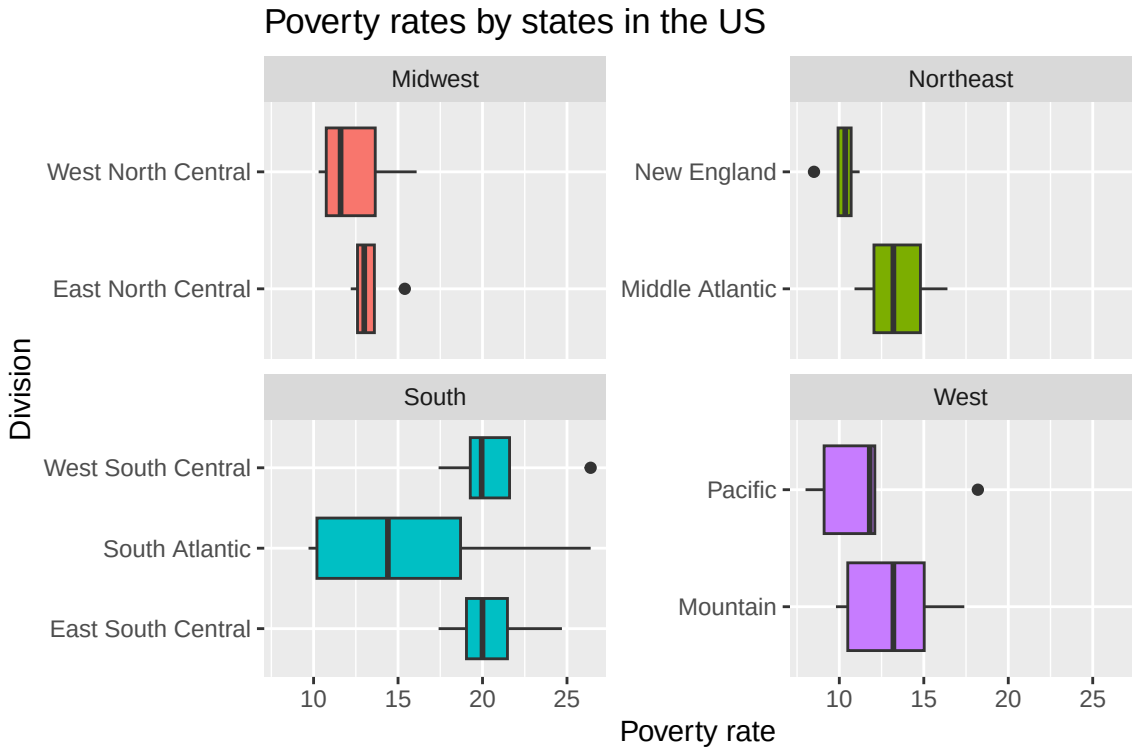
d. Simplify. Since division is nested within region we don't need ALL the levels of division to show up in each panel. Let's "free" our y axis labels by adding `scales="free_y"` to the `facet_wrap`.

```
ggplot(crime, aes(y=poverty, x = division, fill = region)) +
  geom_boxplot() +
  facet_wrap(~region, scales = "free_y")+
  scale_fill_discrete(guide = "none") +
  coord_flip()
```



e. Add titles and annotations. Finish up by adding a plot title, and axis labels. Double check your axis labels before you submit - coord_flip makes this backwards sometimes.

```
ggplot(crime, aes(y=poverty, x = division, fill = region)) +
  geom_boxplot() +
  facet_wrap(~region, scales = "free_y")+
  scale_fill_discrete(guide = "none") +
  coord_flip() +
  labs(
    title = "Poverty rates by states in the US",
    y = "Poverty rate",
    x = "Division"
  )
```



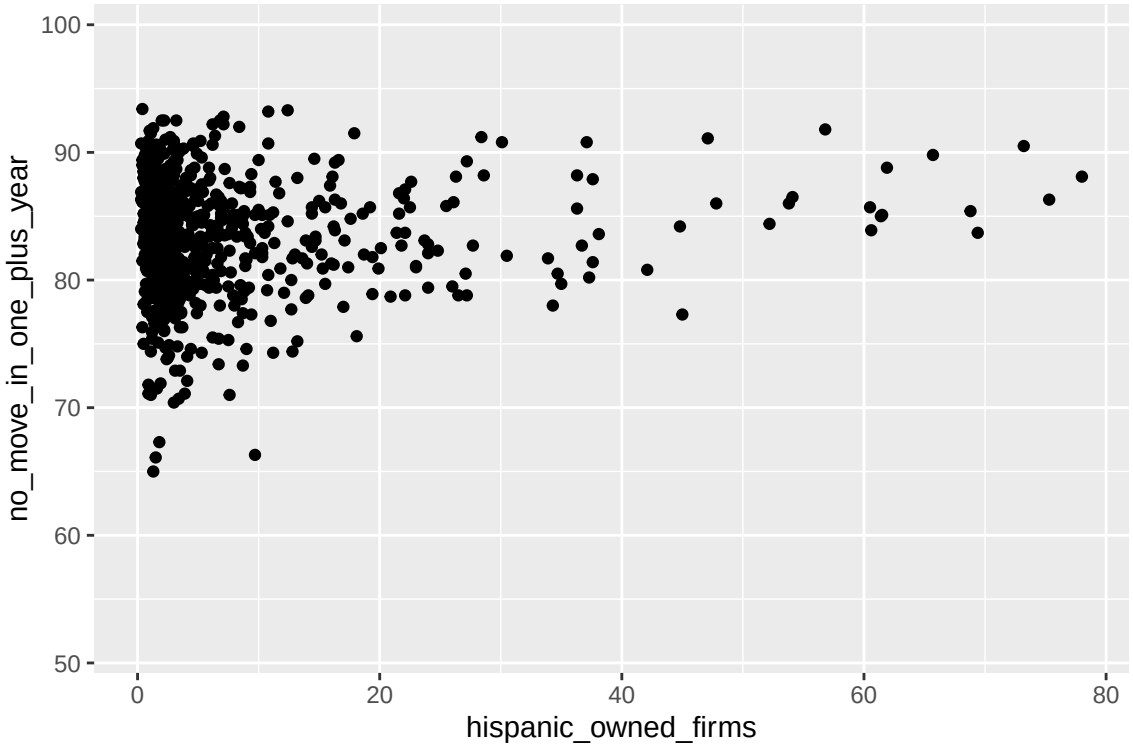
Part III: Plotting aggregated data

This section will ask you to create graphics on data that needs pre-processing using the dplyr verbs prior to plotting. Try to chain your commands together so that each question starts with a data set and ends with a plot.

3. Using the `countyComplete` data set, you will summarize business ownership characteristics and then visualize the results.

a. Use `dplyr` verbs to 1) Group the data by `state`, and then, 2) calculate the average percentage of Hispanic-owned firms (`hispanic_owned_firms`) within each state, and the standard error of this average using `se = number of counties used to calculate each state's average. (_hint: these can be done in the same` mutate `statement with the function` n()‘_)

```
ggplot(county, aes(x=hispanic_owned_firms, no_move_in_one_plus_year)) + geom_point()
```



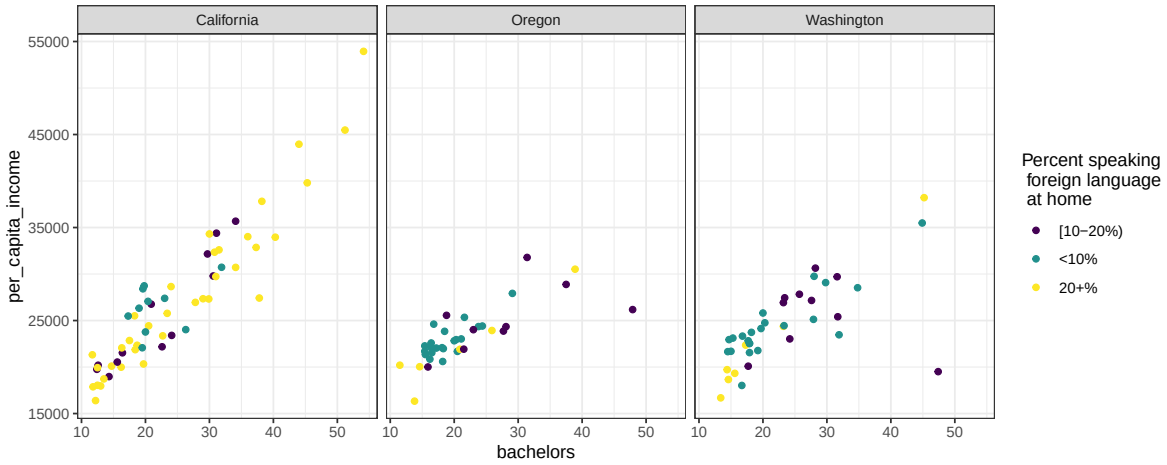
Challenge Questions

1. Using the `countyComplete` data set you loaded earlier, construct a scatterplot for `per_capita_income` against Percentage of Bachelors Degrees `bachelors`, for ONLY the data from California, Oregon, and Washington. Change the color of the dots by Color and panel on `state`, and remove the legend. As above, include a title for your plot and label your axes.

Using the `countyComplete` data set, create a scatterplot to investigate the relationship between the proportion of residents earning a bachelors degree (`bachelors`) income (`per_capita_income`) for west coast states only (CA, OR, WA).

In your visualization, counties should be colored according to grouped levels of the percentage of residents who speak a foreign language at home (<10%, 10-20%, 20%+), where the original numeric variable is recoded into a small number of meaningful percentage ranges. Present the results using separate panels for each state, apply a color scale that is appropriate for categorical groupings, and format the plot so that it is publication-ready with a clear title, labeled axes, and an informative legend.

```
library(paletteer)
county%>%
  filter(state=="California" | state=="Oregon" | state=="Washington") %>%
  mutate(fb = case_when(
    foreign_spoken_at_home < 10 ~ "<10%",
    foreign_spoken_at_home >= 10 &  foreign_spoken_at_home < 20 ~ "[10-20%)",
    foreign_spoken_at_home >=20 ~ "20+%"
  )) %>%
  ggplot(aes(x=bachelors, y=per_capita_income, col=fb)) +
  geom_point() +
  facet_wrap(~state) +
  scale_color_viridis_d(name = "Percent speaking \n foreign language \n at home")   +
  theme_bw()
```




```
  xlab("Bachelors") + ylab("Per Capita Income") +
  ggtitle("Per Capita Income by Bachelors Degree % for West Coast")
```

```
NULL
```