Homework 2

Please copy and paste your code and answers into this document and submit on Canvas.

1. Using the NLSY data, make a scatter plot of the relationship between hours of sleep on weekends and weekdays. Color it according to region (where 1 = northeast, 2 = north central, 3 = south, and 4 = west).

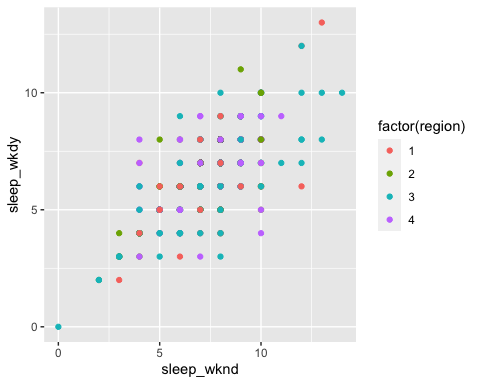
#read in data - SF: do we need to include how to read in the data in the homework instructions? or will we cover in class how to read in that specific dataset as you used the same data for your slides  
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.2 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.1   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

nlsy <- read\_csv(here::here("data", "nlsy\_cc.csv"))

Rows: 1205 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
dbl (14): glasses, eyesight, sleep\_wkdy, sleep\_wknd, id, nsibs, samp, race\_e...  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

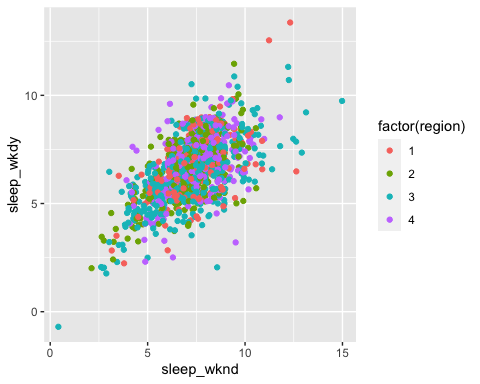
library(ggplot2)  
nlsy |> ggplot(aes(x = sleep\_wknd, y = sleep\_wkdy,   
 color = factor(region))) + geom\_point()



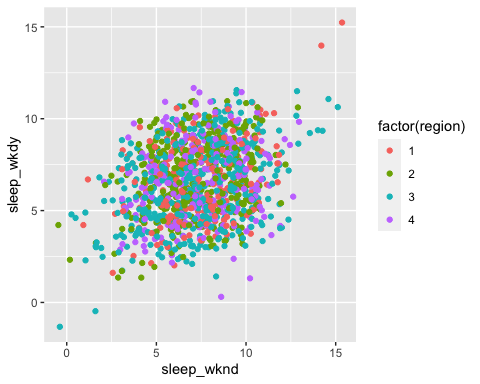
#for all of the following questions, how do you want to handle labeling for grading? would we want nicely labeled graphs or is that not the main point and therefore they can make it pretty with labels etc. if they want but we will not deduct points if they do?

1. Replace geom\_point() with geom\_jitter(). What does this do? Why might this be a good choice for this graph? Play with the width = and height = options. This site may help: https://ggplot2.tidyverse.org/reference/geom\_jitter.html

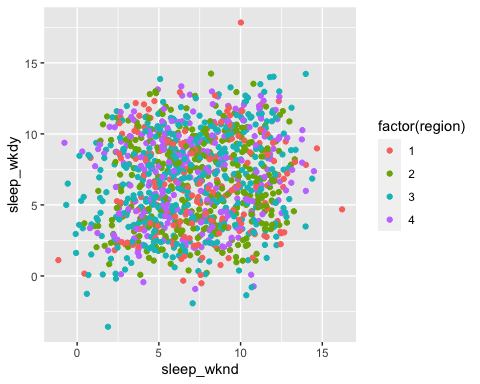
library(ggplot2)  
nlsy |> ggplot(aes(x = sleep\_wknd, y = sleep\_wkdy,   
 color = factor(region))) + geom\_jitter(width=1, height=1)



nlsy |> ggplot(aes(x = sleep\_wknd, y = sleep\_wkdy,   
 color = factor(region))) + geom\_jitter(width=3, height=3)



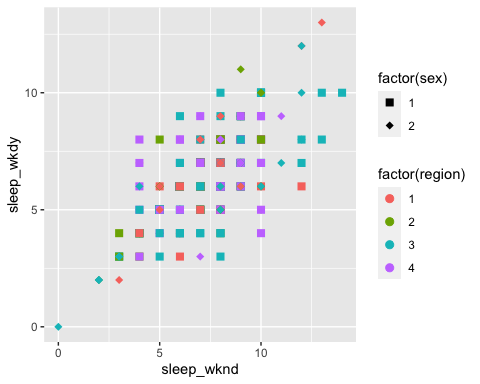
nlsy |> ggplot(aes(x = sleep\_wknd, y = sleep\_wkdy,   
 color = factor(region))) + geom\_jitter(width=5, height=5)



Response: With geom\_jitter() compared to geom\_point(), we can see more plotted dots. This is due to geom\_jitter() adding random dots around the observed values. This might help picture what the natural variability of these variables might look like in a bigger sample and indicate the true underlying distribution more accurately. The width and height option relate how far away the random dots may be located from the observed data.

1. Use the shape = argument to map the sex variable to different shapes. Change the shapes to squares and diamonds. (Hint: how did we manually change colors to certain values? This might help: https://ggplot2.tidyverse.org/articles/ggplot2-specs.html)

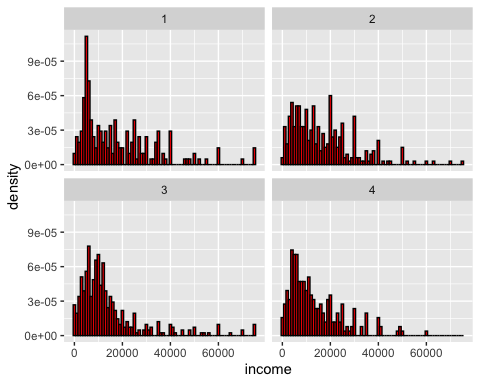
#should they use the geom\_point or geom\_jitter function here?  
#filled square = 15, filled diamond = 18  
  
library(ggplot2)  
nlsy |> ggplot(aes(x = sleep\_wknd, y = sleep\_wkdy,   
 col = factor(region) )) + geom\_point(aes(shape=factor(sex)), size=2.5) + scale\_shape\_manual(values=c(15, 18))



1. When we’re comparing distributions with very different numbers of observations, instead of scaling the y-axis like we did with the facet\_grid() function, we might want to make density histograms. Use google to figure out how to make a density histogram of income. Facet it by region.

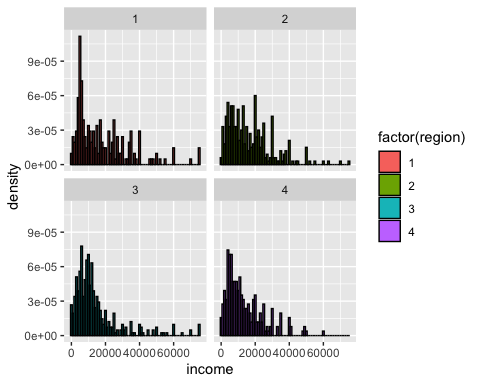
ggplot(data = nlsy, aes(x = income)) +  
 geom\_histogram(aes(y = ..density..), col = "black", fill = "red", binwidth=1000) +  
 facet\_wrap(vars(region))

Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.  
ℹ Please use `after\_stat(density)` instead.

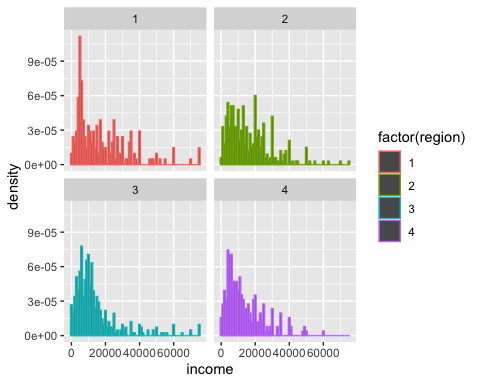


1. Make each of the regions in your histogram from part 1 a different color. (Hint: compare what col = and fill = do to histograms).

ggplot(data = nlsy, aes(x = income)) +  
 geom\_histogram(aes(y = ..density.., fill=factor(region)), col = "black", binwidth=1000) +  
 facet\_wrap(vars(region))



ggplot(data = nlsy, aes(x = income)) +  
 geom\_histogram(aes(y = ..density.., col=factor(region)), binwidth=1000) +  
 facet\_wrap(vars(region))

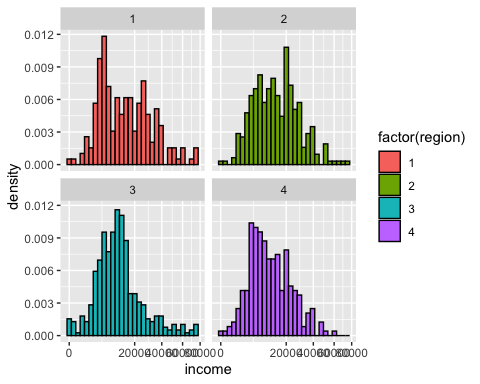


Response: The “col=” option within geom\_histogram() changes the outline color of the bars, the “fill=”-option changes the color of area/the fill of the bars.

1. Instead of a log-transformed x-axis, make a square-root transformed x-axis.

ggplot(data = nlsy, aes(x = income)) +  
 geom\_histogram(aes(y = ..density.., fill=factor(region)), col = "black") +  
 facet\_wrap(vars(region))+  
scale\_x\_sqrt()

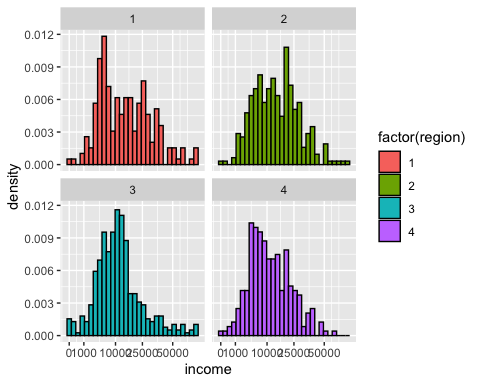
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



1. Doing part 3 squishes the labels on the x-axis. Using the breaks = argument that all the scale\_x\_() functions have, make labels at 1000, 10000, 25000, and 50000.

ggplot(data = nlsy, aes(x = income)) +  
 geom\_histogram(aes(y = ..density.., fill=factor(region)), col = "black") +  
 facet\_wrap(vars(region))+  
scale\_x\_sqrt( breaks=c(0,1000,10000,25000, 50000))

`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



1. The order of the different levels of a factor variable is important. Describe a scenario where different orders of the levels of a factor variable may lead to different results. (Hint: Think about how factor variables are treated a regression analysis.) Comment SF: This may be too advanced. But if the students already know how to run regression analysis in STATA, they might know this. In STATA, the same as in R the lowest/first level is used as reference level for categorical variables in a regression model as default.

Response: A different order of the levels in a factor variable could result in a different reference level in the results that the software returns for a regression model as the first level as specified in the order of levels is automatically set to be the reference level.

1. The “race\_eth”-variable in the NLSY dataset relates to the following level: hispanic (1), black (2), other (3). When loading the data it is a numeric variable. Turn it into a factor variable making sure it is labeled correctly.

nlsy$race\_eth <- factor(nlsy$race\_eth, levels = c(1, 2, 3),   
 labels = c("hispanic", "black", "other"))

1. Create a new variable called “race\_reor”, reordering the levels such that “other” would automatically be used as reference level in analyses. Use the count()-function to check the order of these two variables.

nlsy <- nlsy |> mutate(race\_reor=fct\_relevel(race\_eth, "other", "hispanic", "black"))  
  
count(nlsy,race\_eth )

# A tibble: 3 × 2  
 race\_eth n  
 <fct> <int>  
1 hispanic 211  
2 black 307  
3 other 687

count(nlsy, race\_reor)

# A tibble: 3 × 2  
 race\_reor n  
 <fct> <int>  
1 other 687  
2 hispanic 211  
3 black 307

1. Age of individuals contained in the NLSY dataset ranges from 13 to 52 years (variable: age\_bir). Use case\_when() to create the following age categories: teenager (13-19), young adult (20-29), adult (30-52). Make sure it is a correctly labeled factor variable (hint: this can be a two step process).

nlsy <- nlsy |> mutate(age\_cat=case\_when(age\_bir <=19 ~ 1, age\_bir>=20 & age\_bir<=29 ~ 2, age\_bir>=30 ~ 3))   
nlsy <- nlsy |> mutate(age\_cat=factor(nlsy$age\_cat, levels = c(1, 2, 3), labels = c("teenager", "young adult", "adult")))

1. Using the age group factor variable, show the mean age and total number of individuals in each age group.

nlsy|> group\_by(age\_cat) |> summarise(mean\_age=mean(age\_bir), n=n())

# A tibble: 3 × 3  
 age\_cat mean\_age n  
 <fct> <dbl> <int>  
1 teenager 17.3 358  
2 young adult 23.7 648  
3 adult 33.7 199