STA 141 Worksheet 3

Richard McCormick

September 21, 2023

Due Date: Thursday, September 28, 2023 before 11:00am.

Instructions

Worksheets must be turned in as a PDF file through Canvas. The worksheet is worth a total of **15 points**, which is 3 percent of your overall grade.

Exercises

Begin by running the following code block to add the packages we need to use to our library.

Exercise 1

(a) The first dataset we are going to work with comes built-in with the `tidyverse` package. Run the following to save a copy of the `midwest` dataset to a variable called `my.midwest`.

```
my.midwest <- midwest
```

The first thing we want to do is to create a bar chart for the `state` variable. Use the following code block to check if this variable is categorical or numerical. If it is not categorical convert it so that we can use it for our bar chart.

```
class( my.midwest$state )
```

```
## [1] "character"
```

```
my.midwest$state <- as.factor( my.midwest$state )
```

(b) According to this data, how many states are there in the midwest? You should be able to write something that displays this answer.

```
str( my.midwest$state )
```

```
##  Factor w/ 5 levels "IL","IN","MI",..: 1 1 1 1 1 1 1 1 1 1 ...
```

According to this data, there are 5 states in the Midwest.

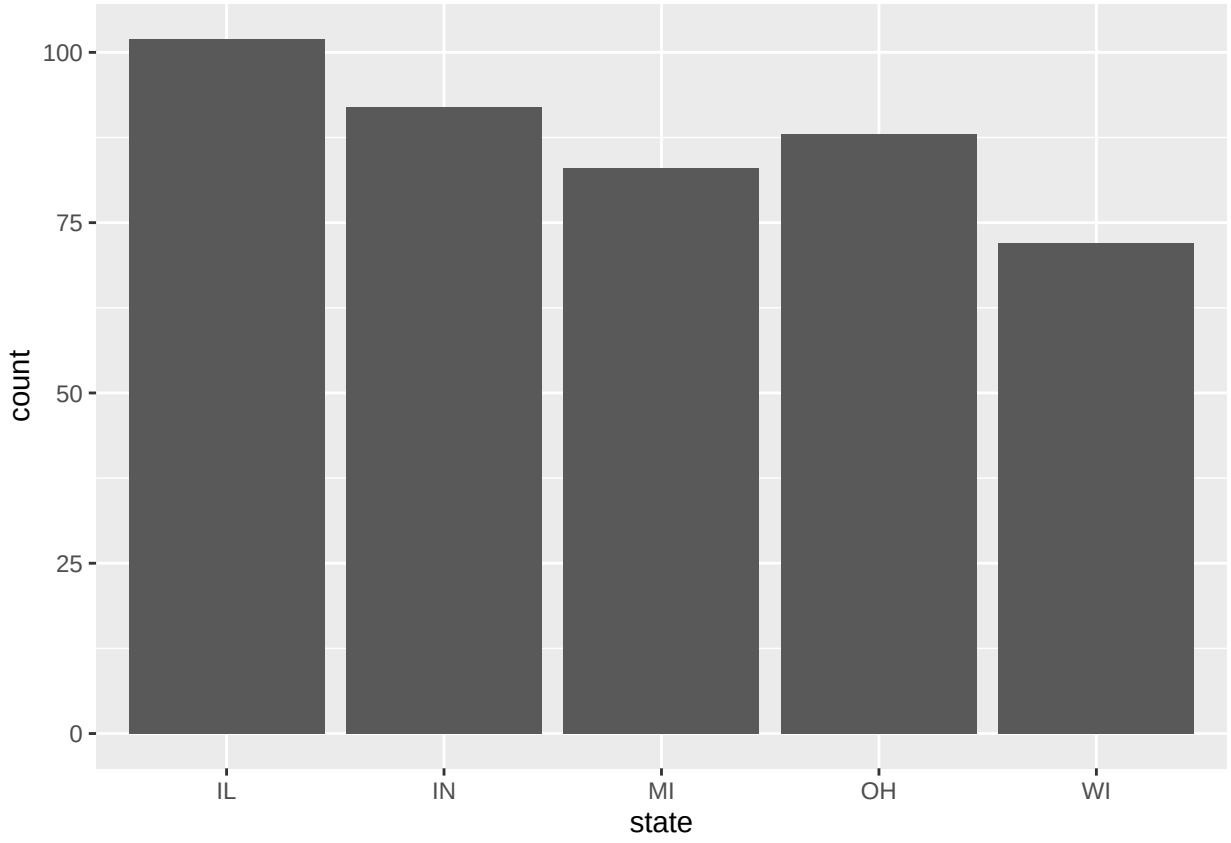
(c) We'll start by making a frequency distribution table. See if you can work out how to use the '*table*()' function. If necessary you can run `?table` in the console to see the help documents.

```
table( my.midwest$state )
```

```
##
##  IL  IN  MI  OH  WI
## 102  92  83  88  72
```

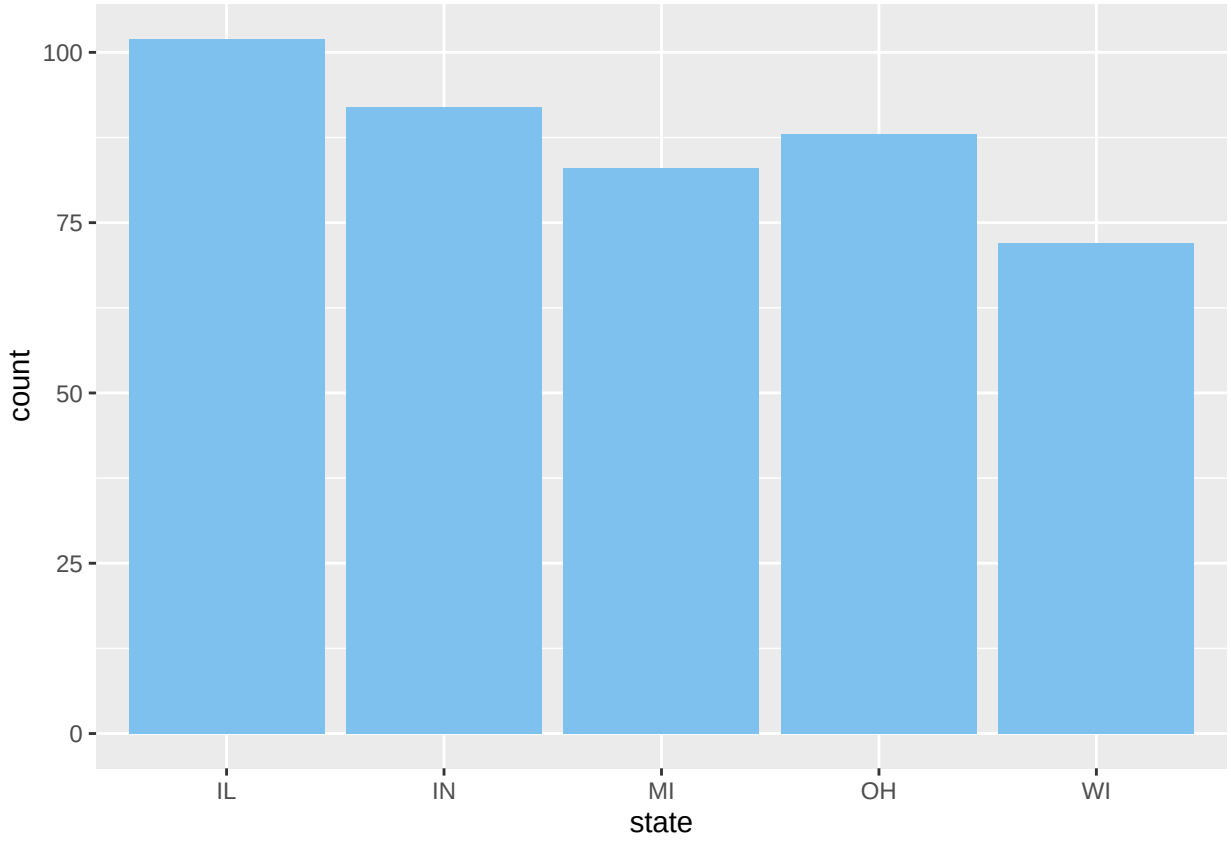
(d) Now use the next code block to see if you can create a bar graph for `state`:

```
state_bar_plot <- ggplot( data=my.midwest ) +
  geom_bar( aes( x=state ) )

state_bar_plot
```



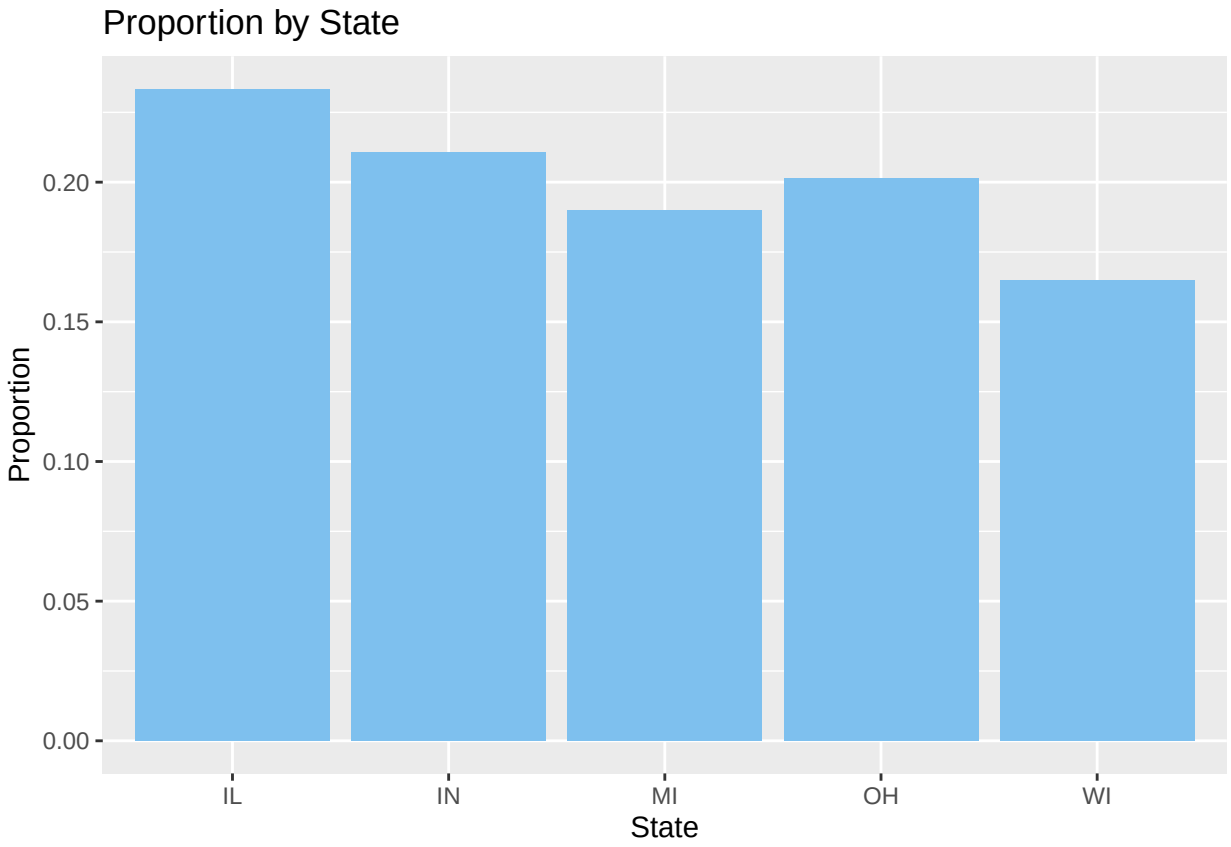
(e) Starting with the graph above, change the color of the bars using `fill=` inside the `geom_bar()` function.

```
state_bar_plot <- ggplot( data=my.midwest ) +
  geom_bar( aes( x=state ), fill='skyblue2' )

state_bar_plot
```



(f) The next thing we might want to do to our bar chart is to make the y-axis a proportion rather than a count. We can do this using the y-axis aesthetic. Here we can tell $\mathbb{R}$ that we want to use the count of the levels (`after_stat(count)`) and divide it by the total number, or the sum of the counts (`sum(after_stat(count))`). Plot a bar chart of the `state` variable with proportions on the `y-axis` below. Change the label on the `y-axis` to say proportion too.

```
state_bar_plot <- ggplot( data=my.midwest ) +
  geom_bar( aes( x=state, y=after_stat( count )/sum(after_stat( count )) ),
            fill='skyblue2' ) +
  labs( y="Proportion", x="State", title="Proportion by State" )

state_bar_plot
```

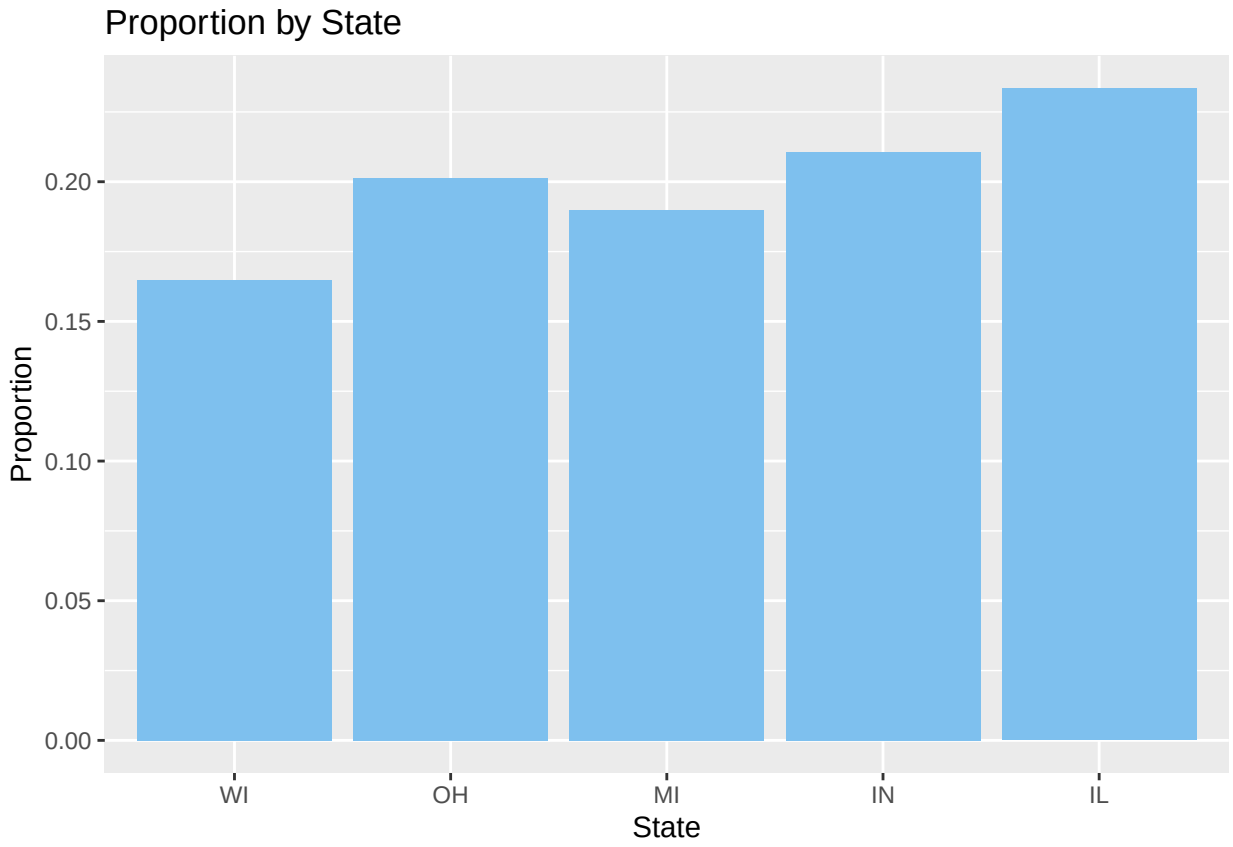


(g) Often when we are doing modeling problems in data science, it is preferable to have approximately the same number of observations from each class in the response variable. If `state` were our response variable, do we have an even distribution across levels of the variable?

The distribution of state is not equally distributed. While from a very generous perspective, we might be able to say they are roughly equal, there are significant differences between the frequency of levels.

(h) Finally, it's good to know how to reorder the levels on the x-axis. We can do this by redefining the variable as a factor and then giving the order we would like them to be displayed in (as a vector using `c()` - any time we want to use a list or vector of data in $\mathbb{R}$ we use this function). Reproduce the plot from `(f)` but reorder the levels in any way that you see fit.

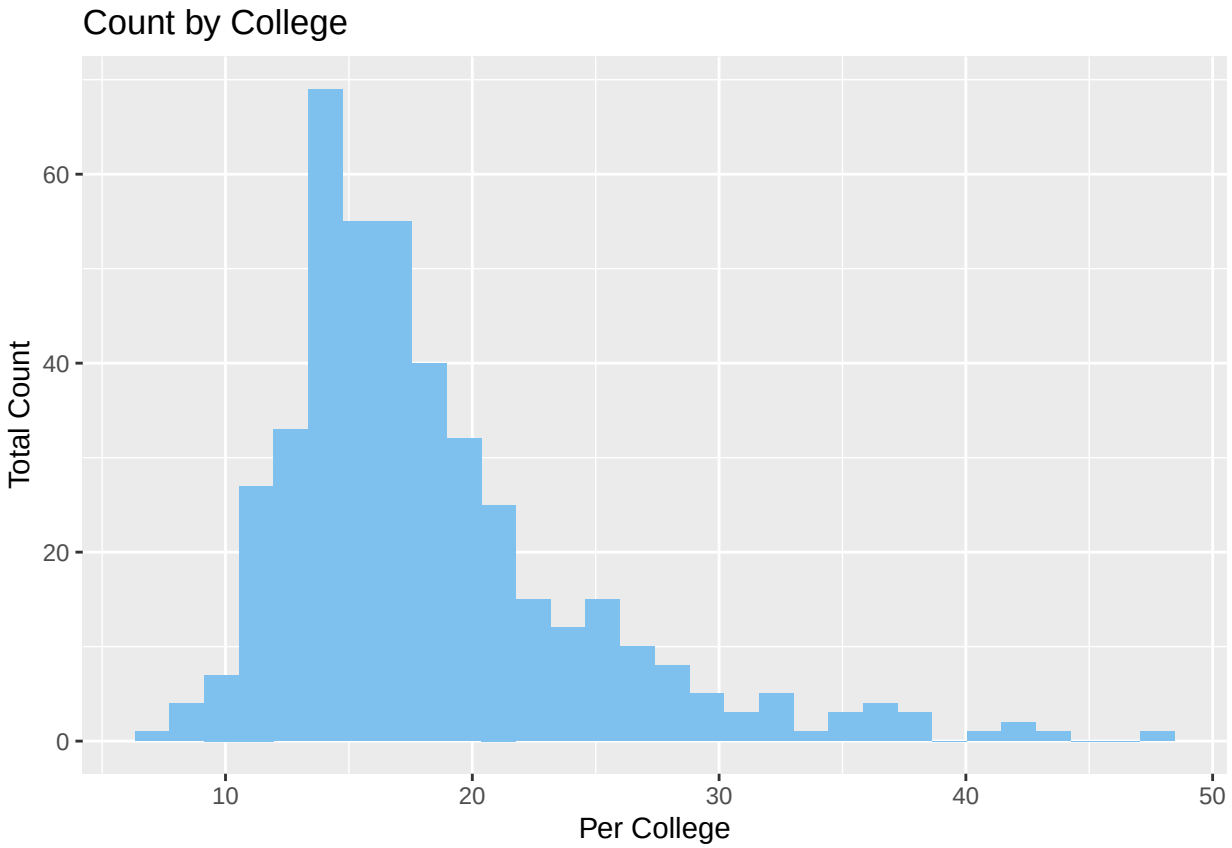
```
my.midwest$state <- as.factor( my.midwest$state )
my.midwest$state <- factor( my.midwest$state,
                            levels=c( "WI", "OH", "MI", "IN", "IL" ) )

state_bar_plot <- ggplot( data=my.midwest ) +
  geom_bar( aes( x=state, y=after_stat( count )/sum(after_stat( count )) ),
            fill='skyblue2' ) +
  labs( y="Proportion", x="State", title="Proportion by State" )

state_bar_plot
```



(i) Now let's turn our attention to histograms. Using the same dataframe, create a histogram of the percentage of adults who have a college degree (the `percollege` variable). Fill the histogram with a color of your choice.

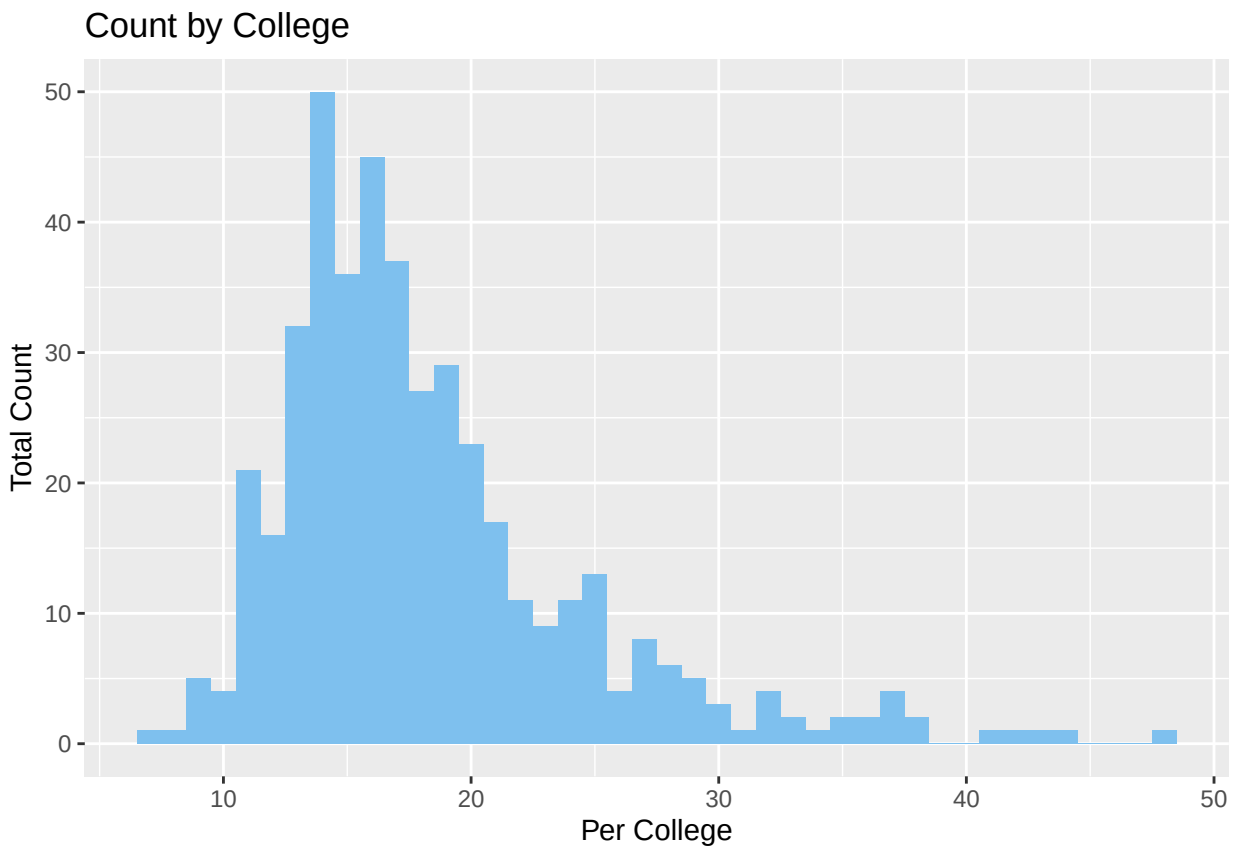
```
histogram <- ggplot( data=my.midwest ) +
  geom_histogram( aes( x=percollege ), fill='skyblue2' ) +
  labs( x="Per College", y="Total Count", title="Count by College" )

histogram
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



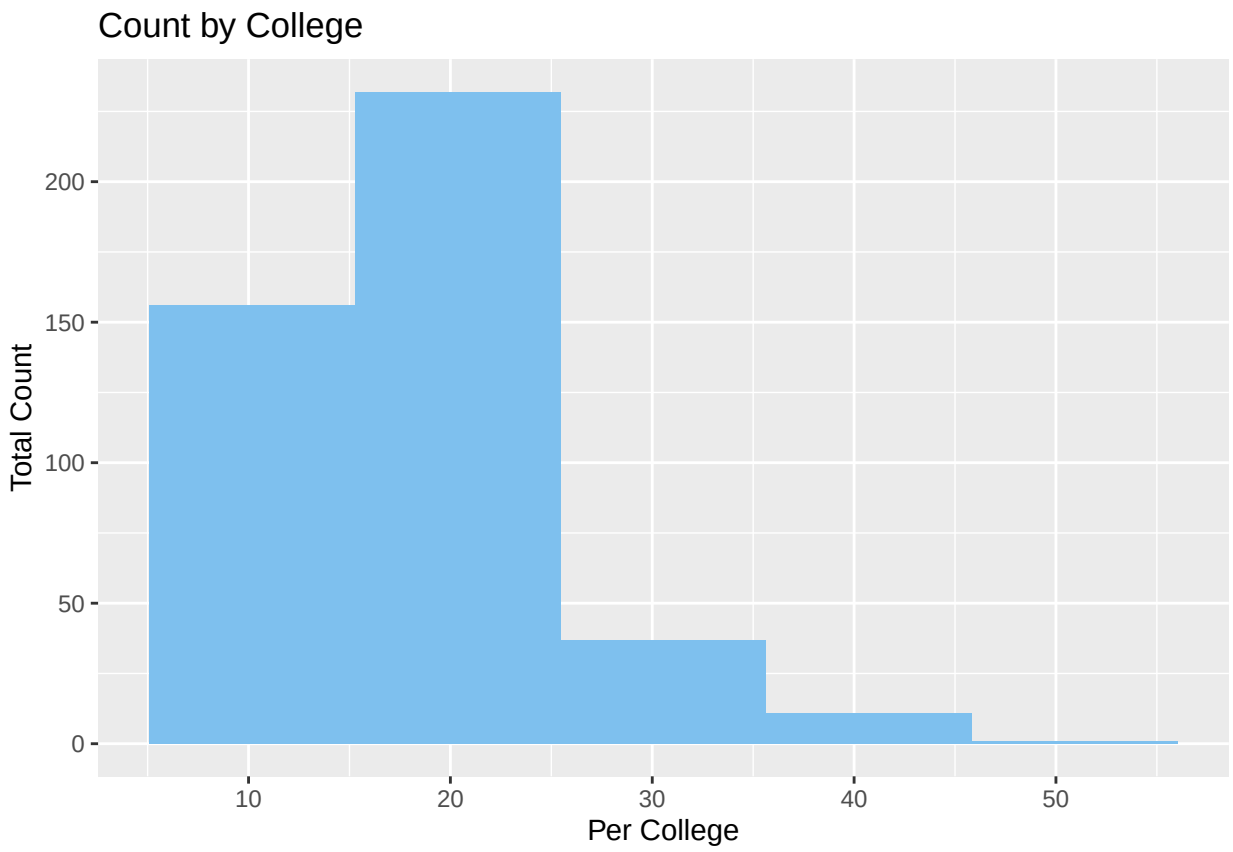
(j) Using the histogram from (i) as a starting point, change the bin width to 1 (how to do this was shown in the lecture slides).

```
histogram <- ggplot( data=my.midwest ) +
  geom_histogram( aes( x=percollege ), fill='skyblue2', binwidth=1 ) +
  labs( x="Per College", y="Total Count", title="Count by College" )

histogram
```



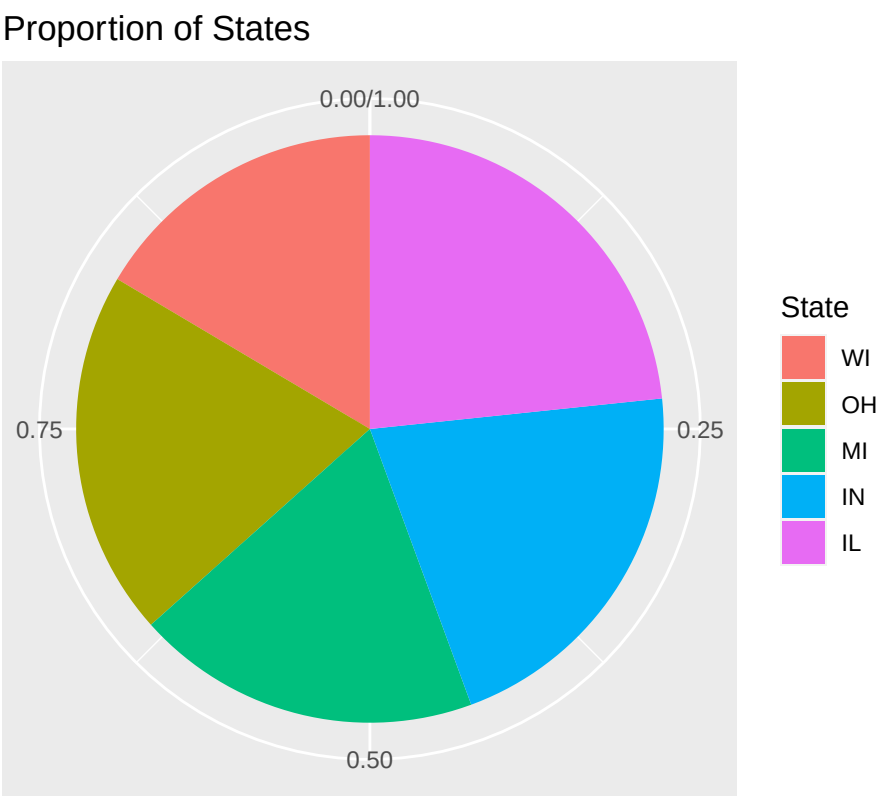
(k) We can also change the number of bins, rather than the bin width. See if you can modify your histogram from (j) to have only 5 bins.

```
histogram <- ggplot( data=my.midwest ) +
  geom_histogram( aes( x=percollege ), fill='skyblue2', bins=5 ) +
  labs( x="Per College", y="Total Count", title="Count by College" )

histogram
```



(l) Finally, try to create a pie chart of the `state` variable using the code given to you in the lecture slides.

```
histogram <- ggplot( data=my.midwest,
                     mapping=aes( x="",
                                  y=after_stat( count )/sum( after_stat( count ) ),
                                  fill=state ) ) +
  geom_bar() +
  coord_polar( theta="y" ) +
  labs(x="", y="", fill="State", title="Proportion of States" ) +
  theme( axis.line=element_blank(),
         axis.ticks=element_blank() )

histogram
```



Exercise 2

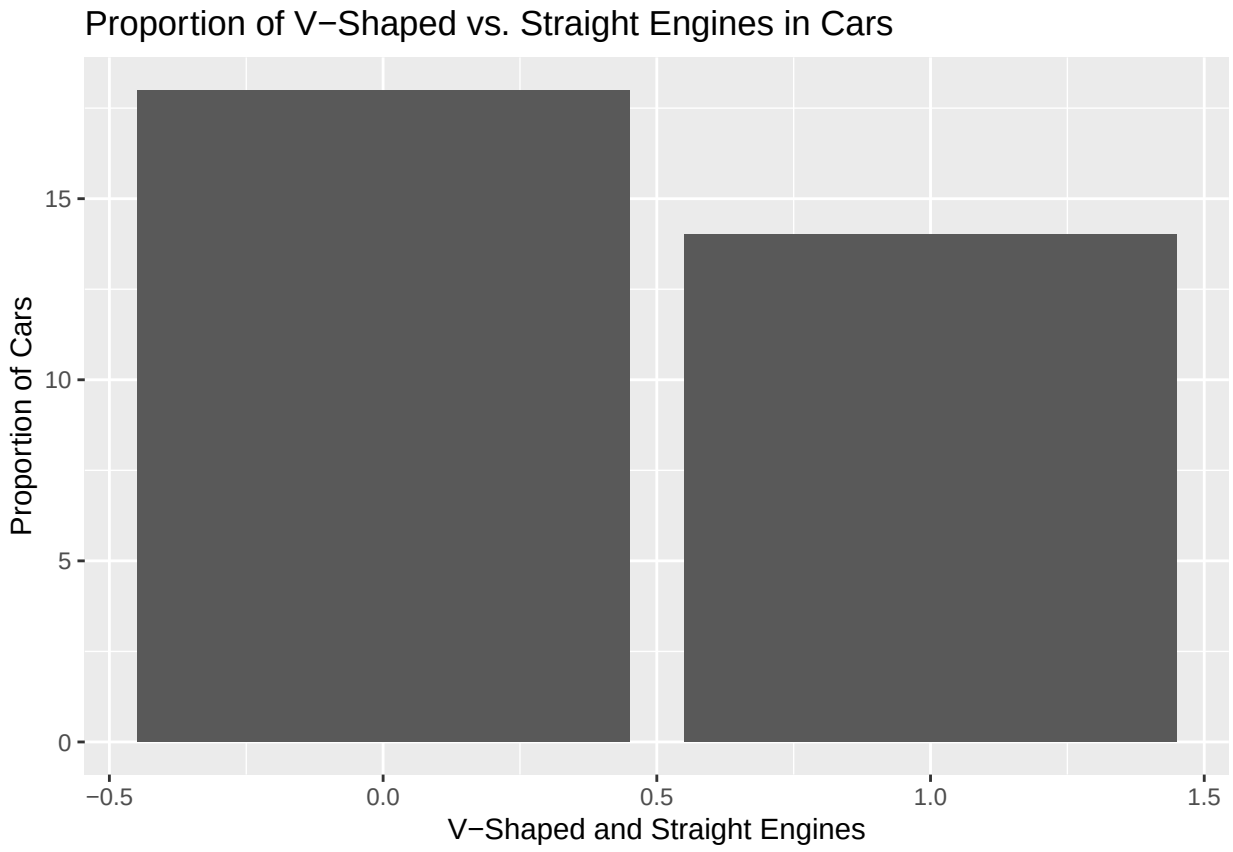
(a) The Motor Trend Car Road Tests dataset is also built-in to ℝ. Save a copy of the dataset (called `mtcars`) as `my.cars`, in the same way we did in 1(a).

```
my.cars <- mtcars
```

Don't forget that for any of the following questions you can find out more about the data by looking at the help documents in the help window on the right!

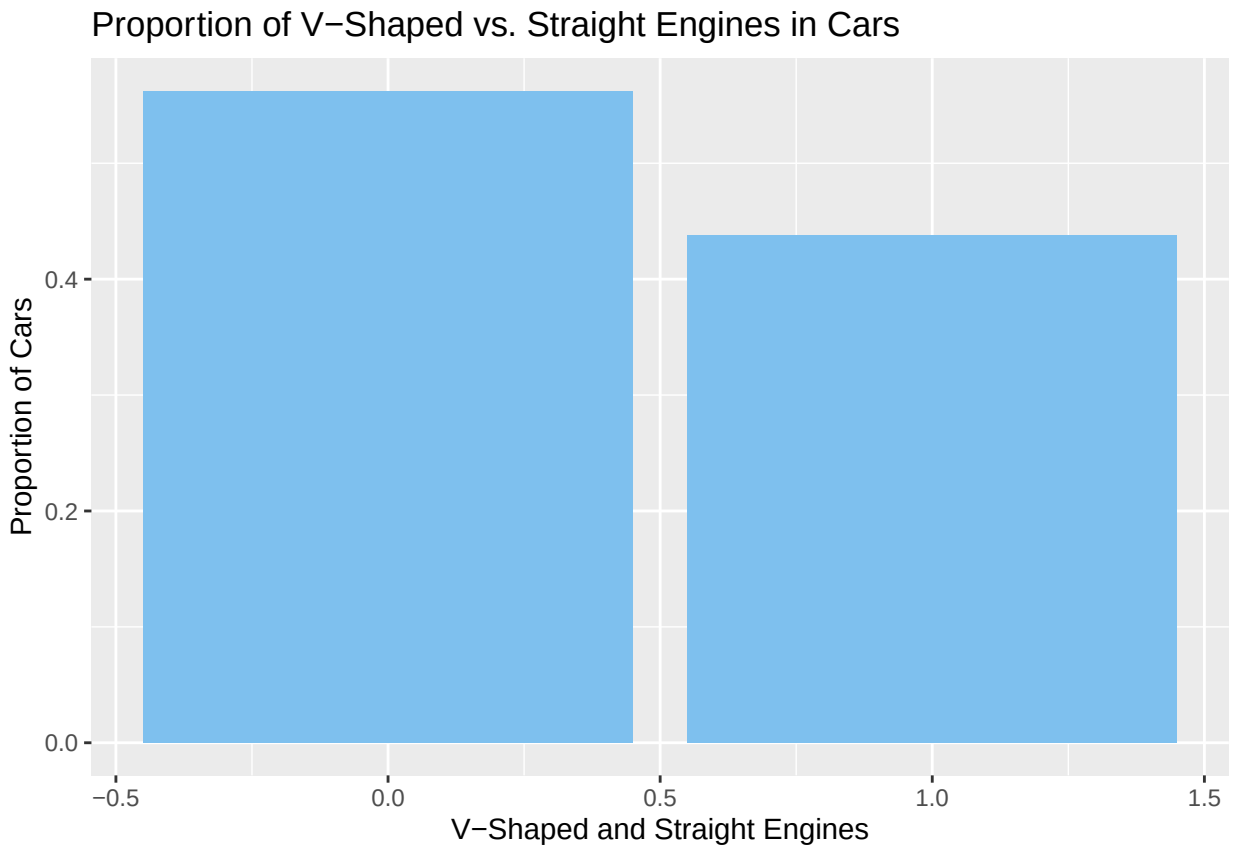
(b) Create a bar graph showing the number of v-shaped and straight engines in the data.

```
bar_graph <- ggplot( data=my.cars ) +
  geom_bar( aes( x=vs ) ) +
  labs( x="V-Shaped and Straight Engines", y="Proportion of Cars",
        title="Proportion of V-Shaped vs. Straight Engines in Cars" )

bar_graph
```



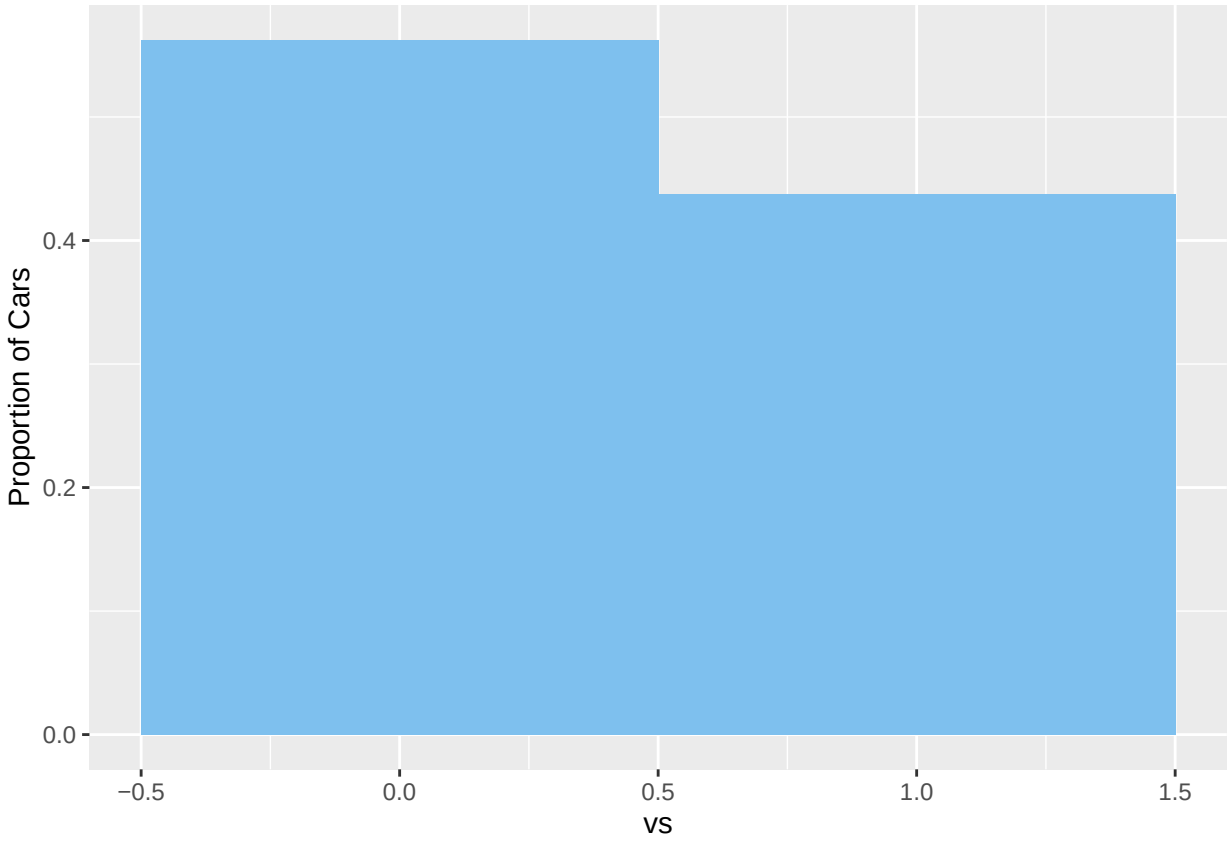
(c) Change the plot from part (b) to make the y axis a proportion, add a label to the y-axis that says `Proportion of cars`, and change the color of the bars to a color of your choice.

```
bar_graph <- ggplot( data=my.cars ) +
  geom_bar( aes( x=vs, y=after_stat( count )/sum( after_stat( count ) ) ),
            fill='skyblue2' ) +
  labs( x="V-Shaped and Straight Engines", y="Proportion of Cars",
        title="Proportion of V-Shaped vs. Straight Engines in Cars" )

bar_graph
```



(d) If vs was our response variable would we have an approximately even distribution between levels?

```
bar_graph <- ggplot( data=my.cars ) +
  geom_histogram( aes( x=vs, y=after_stat( count )/sum( after_stat( count ) ) ),
                  fill='skyblue2', bins=2 ) +
  labs( y="Proportion of Cars" )

bar_graph
```



The distribution between the two levels do not appear to be evenly distributed. There is a significant difference between the two levels, although they are relatively close, within 15%.

(e) Create a histogram of the variable that represents engine size. Use a binwidth of 100, change the color of the bars to `pink2`, and change the color of the borders of the bars to black.

```
bar_graph <- ggplot( data=my.cars ) +
  geom_histogram( aes( x=disp, y=after_stat( count )/sum( after_stat( count ) ) ),
            fill='pink2', binwidth=100, color='black' ) +
  labs( y="Proportion of Cars", x="Displacement", title="Displacement of Car Engines" )

bar_graph
```