Data Science for Beginners, University of Essex

Day 5: Data Visualization

Dr. Howard Liu

14-01-2022

Learning Objectives Today

- Why ggplot?
- Faceting
- Histogram
- Box-plot
- Scatter-plot
- Smoothing
- Maps

We will learn them by performing a simple bivariate analysis (X-Y).

Why ggplot?

- ggplot2 is a plotting system for R, based on the grammar of graphics, which tries to take the good parts of base and lattice graphics and none of the bad parts.
- It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics.
- A great tool to visualize statistical relationships with human-readable syntax, and can generate aesthetically pleasing figures.

First we need to call the package from the package library

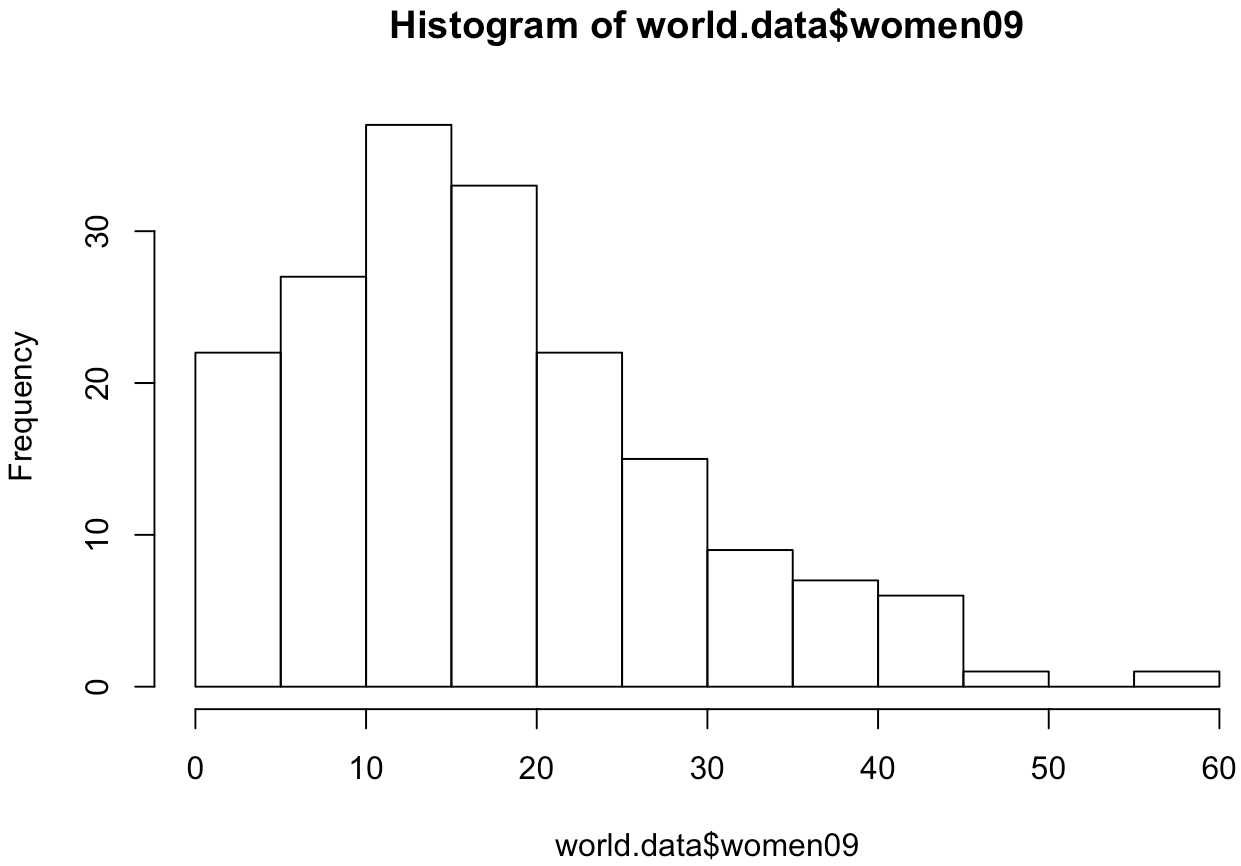
```
library(tidyverse) # the package for many useful functions, including ggplot2
library(ggplot2) # or you could just call this package
```

1. A simple task

Let's use the `world.csv` data again. Say we want to draw a simple histogram showing the `percentage of women in congress` around the world. Use base R function, we can generate something simple but not very aesthetically pleasing.

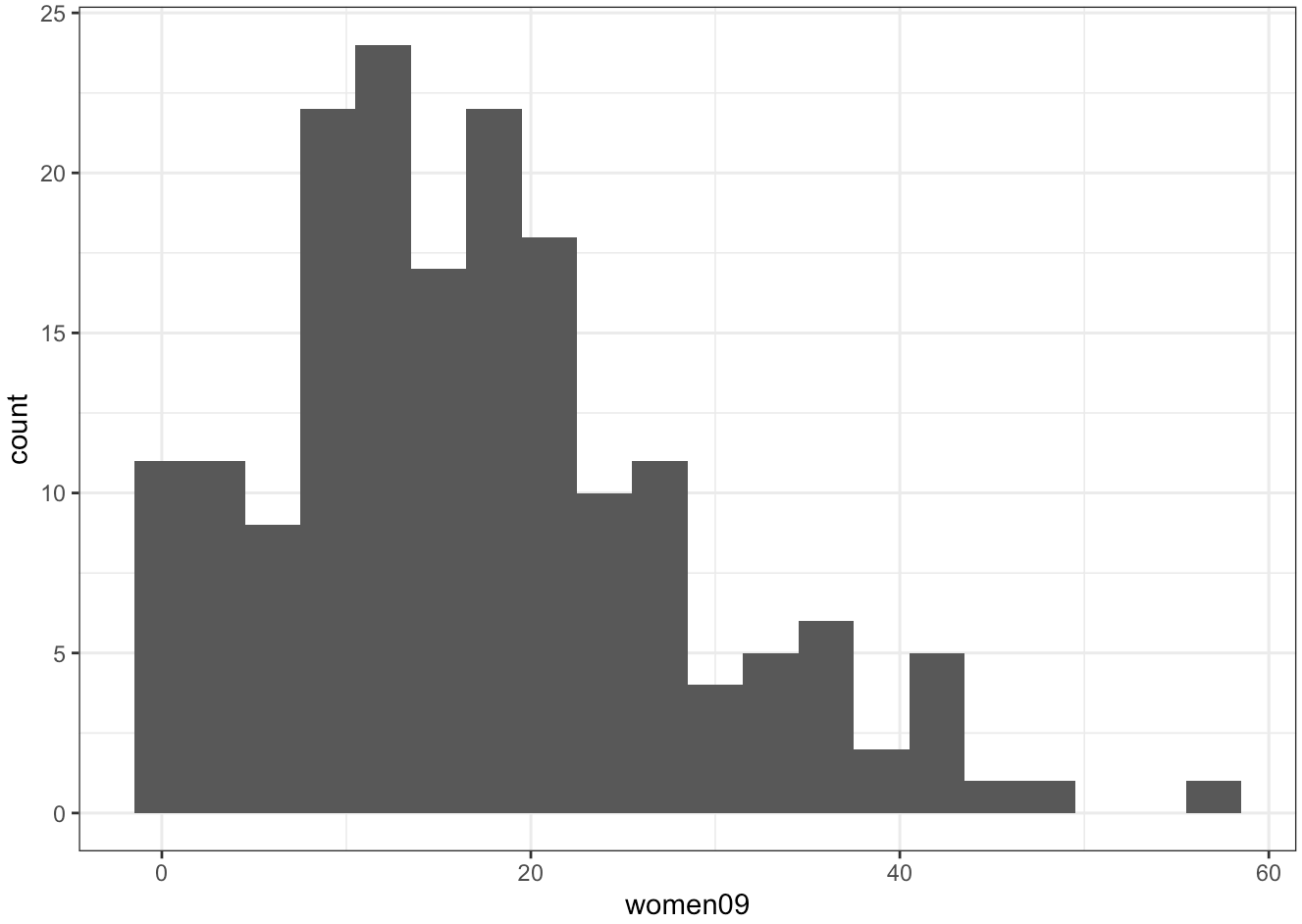
```
myPath <- "/Users/howardliu/Dropbox/Essex/data-programming-beginners/Lecture/"
setwd(myPath)
world.data = read.csv("world.csv") %>% as_tibble() # here I make it a tibble data frame just for illustration purpose
```

```
hist(world.data$women09)
```

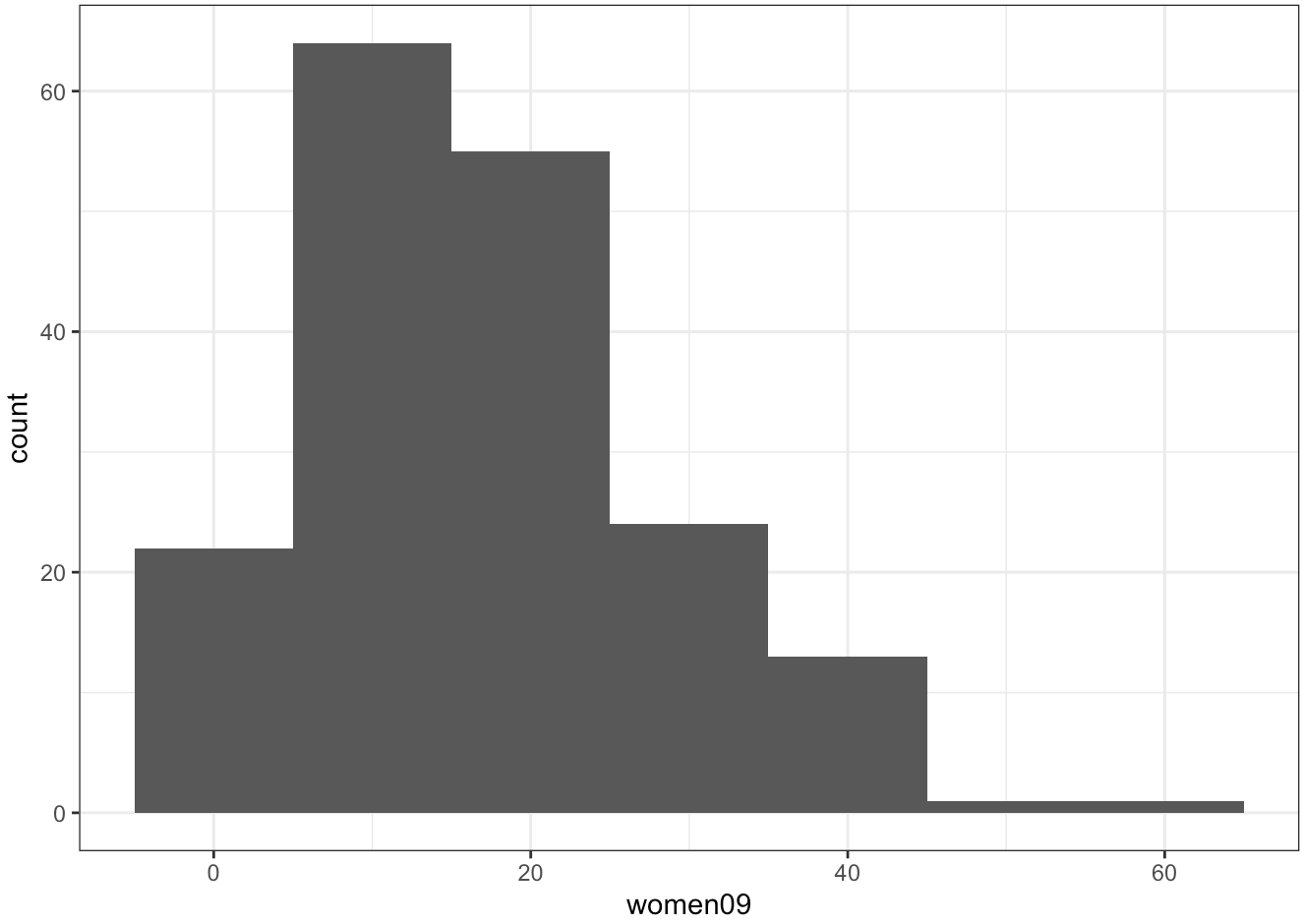


We can easily draw some prettier plots using the ggplot functions `geom_histogram` .

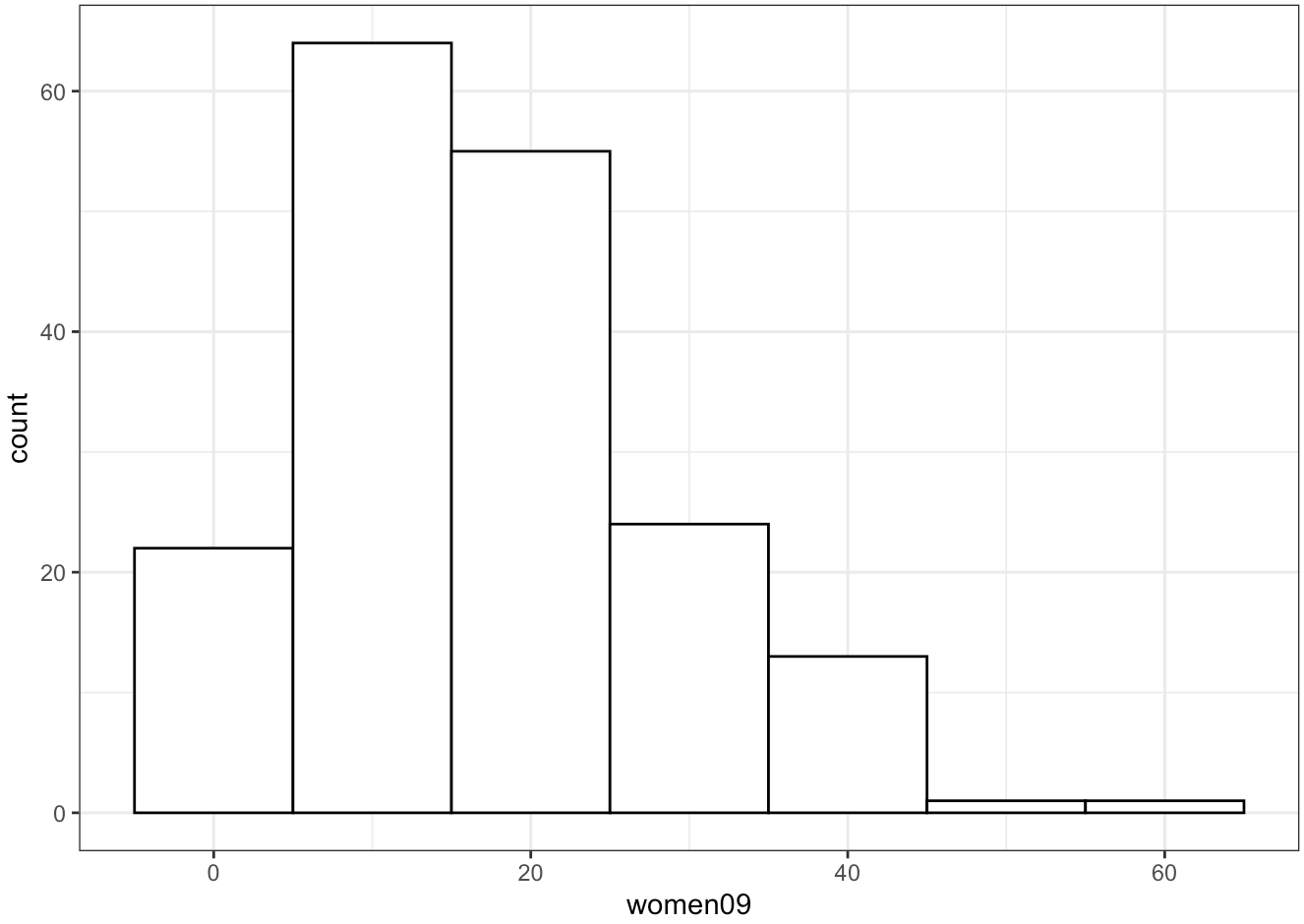
```
ggplot(world.data, aes(x = women09)) +  # aes is aesthetic specifications. Here we specify that the x is the variable of women09
  geom_histogram(binwidth = 3) + # and plot a histogram (with binwidth = 3)
  theme_bw() # with a black-white theme
```



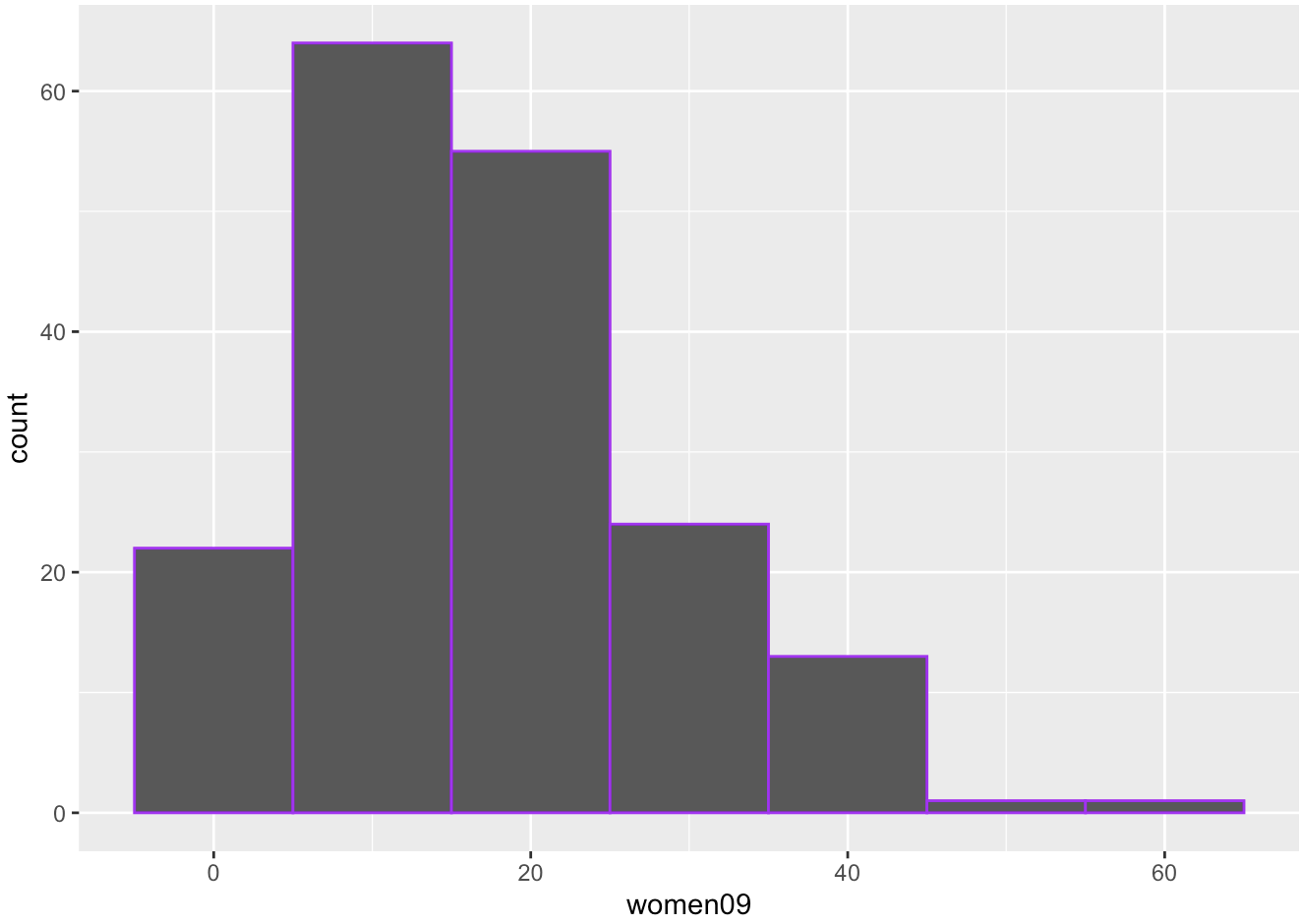

```
ggplot(world.data, aes(x = women09)) +  # aes is aesthetic specifications. Here we speci
fy that the x is the variable of women09
  geom_histogram(binwidth = 10) + # and plot a histogram (with binwidth = 10)
  theme_bw() # with a black-white theme
```



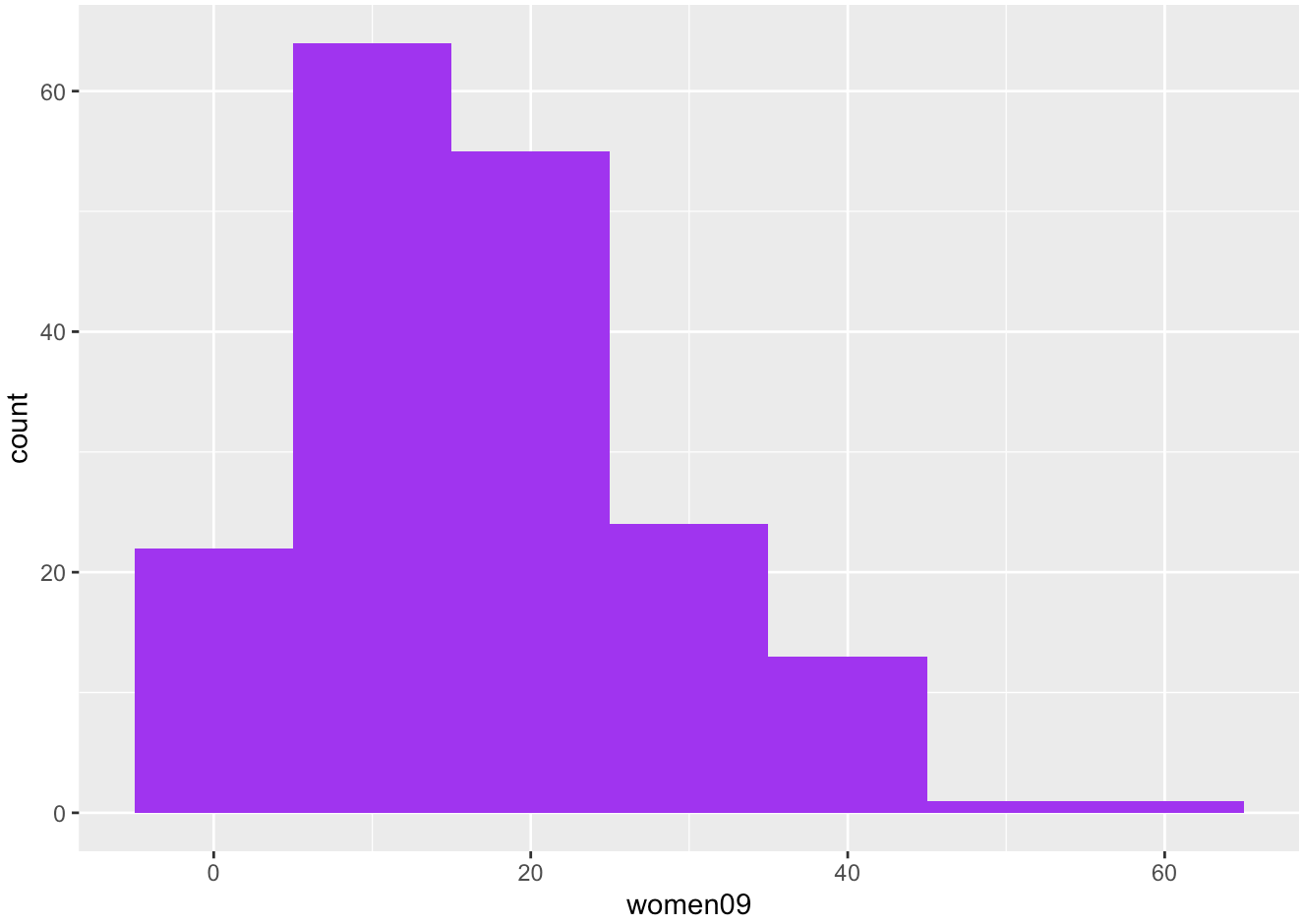

```
# make it hollow
ggplot(world.data, aes(x = women09)) +
  geom_histogram(binwidth = 10, colour="black", fill="white") + # with white color
  theme_bw()
```




```
ggplot(world.data, aes(x = women09)) +
  geom_histogram(binwidth = 10, colour="purple") # the purple boundary color
```

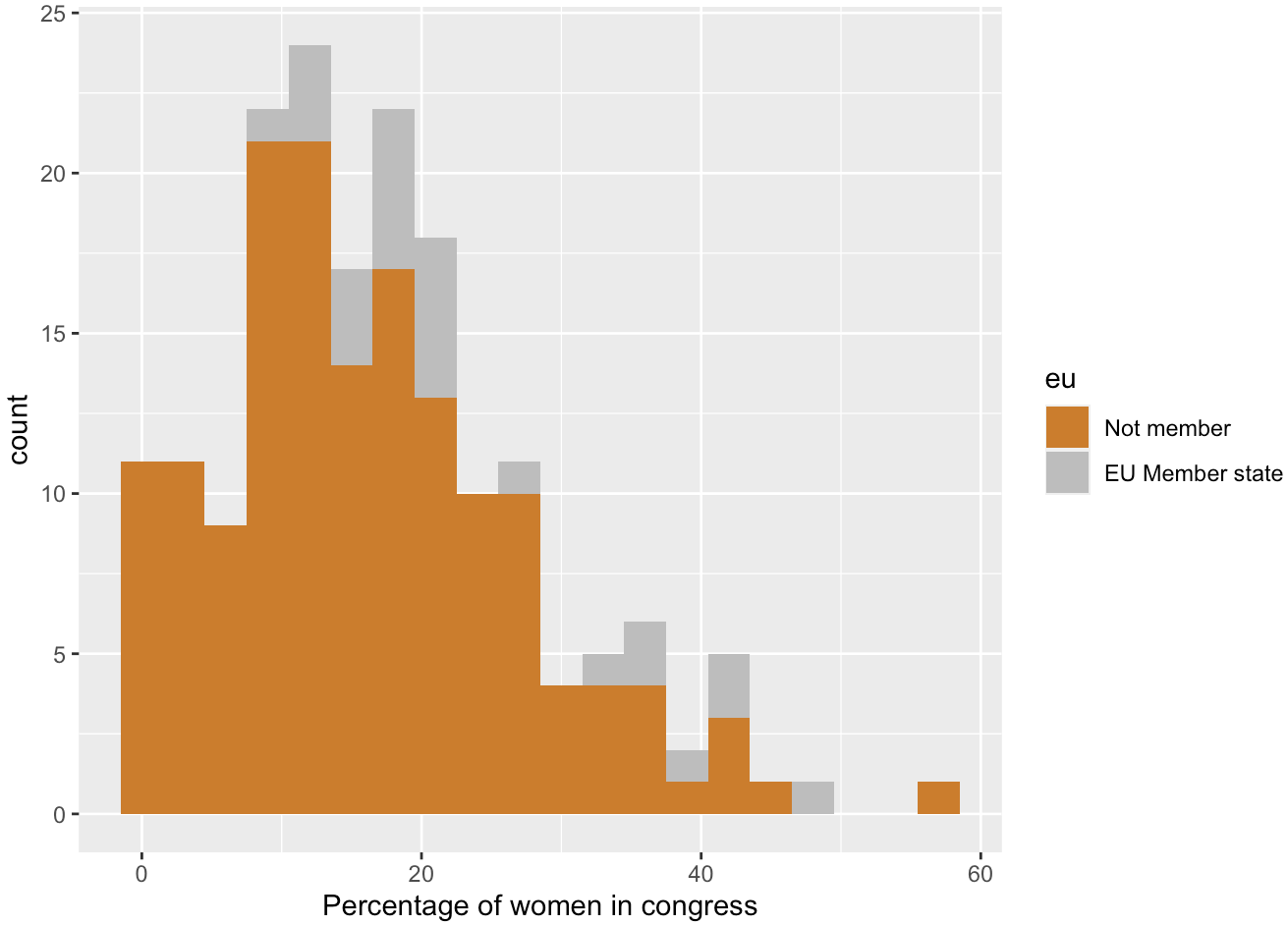



```
ggplot(world.data, aes(x = women09)) +  # aes is aesthetic specifications. Here we speci
fy that the x is the variable of women09
  geom_histogram(binwidth = 10, fill="purple")  # fill the the histogram with what color
```



We can even differentiate women presentation in EU and non EU countries and show them in different colors by using the statement in ggplot `fill = eu` .

```
ggplot(world.data, aes(women09, fill = eu)) +
  geom_histogram(binwidth = 3) +
  xlab("Percentage of women in congress") + # x label
  scale_fill_manual(values = c("Not member" = "#CD7F32", # Manually change colors.
                               "EU Member state" = "#C0C0C0")) # You can google for more
color codes
```



2. Example: a basic bivariate (X-Y) statistical analysis with graphs

Here, we will analyze the relationship between **pr_sys** (we treat this as X) and **women09** (we treat this as Y) in order to test the hypothesis that **female representation** is better in countries that implement a **proportional representation (PR) system**.

That is, our hypothesis is that the mean of women09 is higher for countries with a PR system than for countries without a PR system.

Before doing any bivariate statistical analysis, we should investigate the variables individually (i.e., do univariate analyses) first, to get to know the data.

First, let's take a look at our X (also dealing with the NAs)

```
# 1. We observe NAs
summary(world.data$women09)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    9.70   15.55   17.18   22.95   56.30      11
```

```
# 2. Let's create a new data that don't have NAs in women09
women.pr <- world.data[!is.na(world.data $ women09),]

# We can see that NA cases have been correctly removed
summary(women.pr $ women09)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.70   15.55   17.18   22.95   56.30
```

Second, let's take a look at our Y (also dealing with the NAs)

```
# No NA found
summary(women.pr$pr_sys)
```

```
##  No Yes
## 114  66
```

```
# Let's make this variable more readable by converting YES/NO --> "PR System", "Non-PR"
(called relabeling)
women.pr$pr <- factor(women.pr $ pr_sys,
                      levels = c("Yes", "No"),
                      labels = c("PR System", "Non-PR"))

# Again, we should check if this worked correctly by looking at this newly-created varia
ble
summary(women.pr$pr)
```

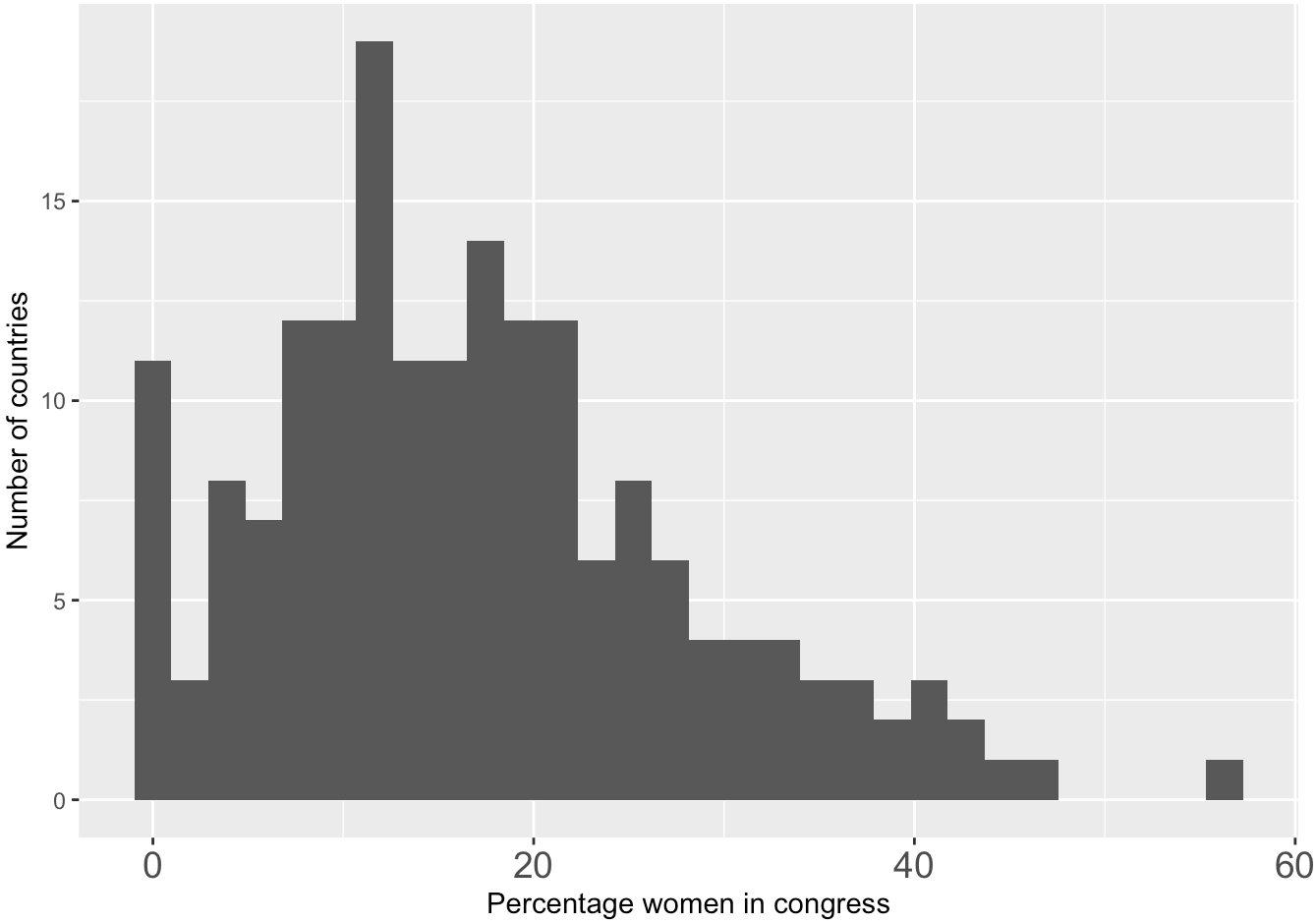
```
## PR System    Non-PR
##        66       114
```

Histogram of Y

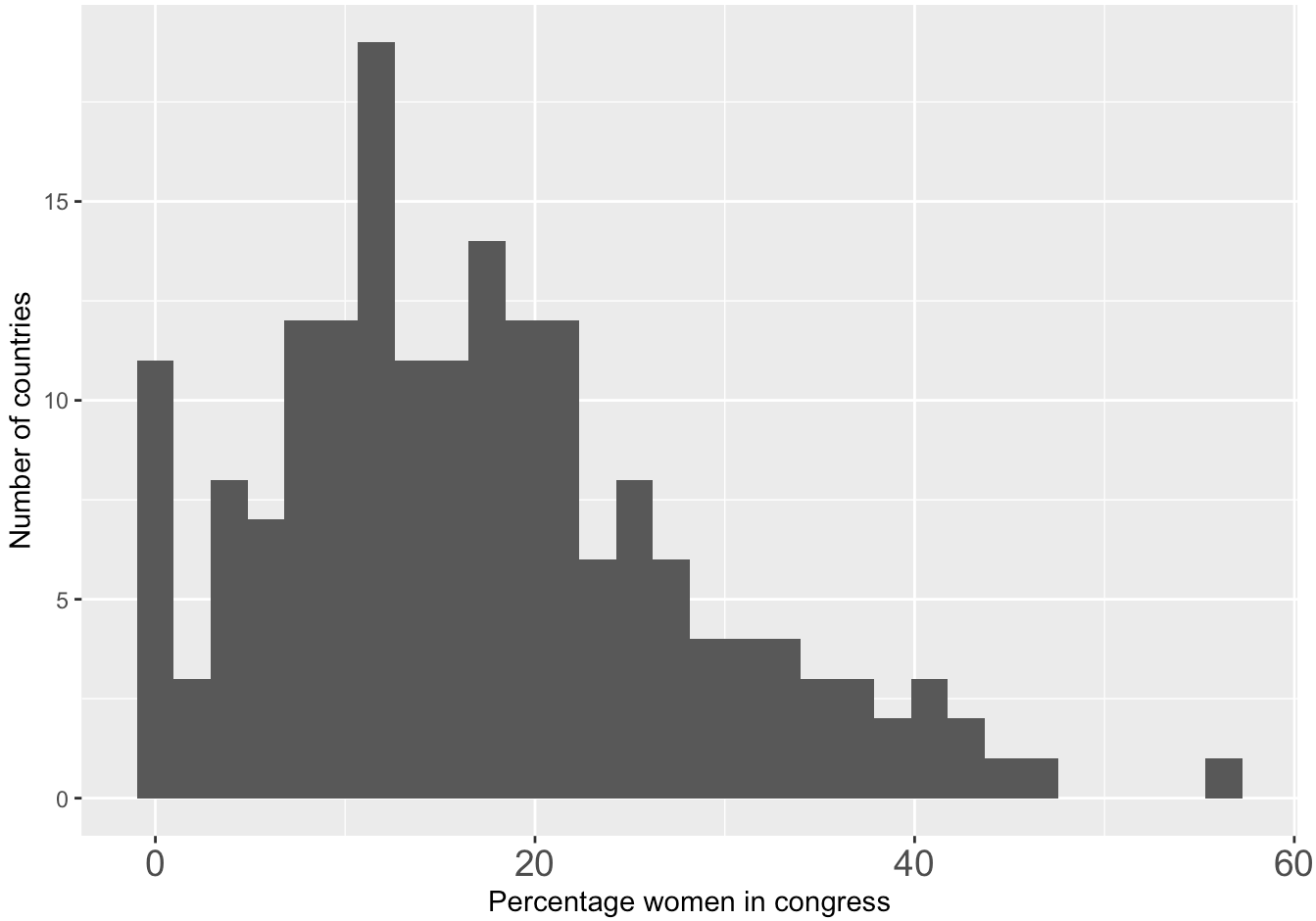
```
#=============================================
# Describe Y (numerically and graphically)
#=============================================

# Numerical summary
summary(women.pr$women09)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    9.70   15.55   17.18   22.95   56.30
```

```
# Graphical summary
g <- ggplot(women.pr, aes(x = women09)) + geom_histogram(bins = 30)
g <- g + theme(axis.text.x = element_text(size = 14)) # specify text font axis.text.x to
14
g <- g + xlab("Percentage women in congress") + ylab("Number of countries")
g
```




```
# You can do this in one chain of commands as well
g <- ggplot(women.pr, aes(x = women09)) +
  geom_histogram(bins = 30) +
  theme(axis.text.x = element_text(size = 14)) +
  xlab("Percentage women in congress") +
  ylab("Number of countries")
g
```



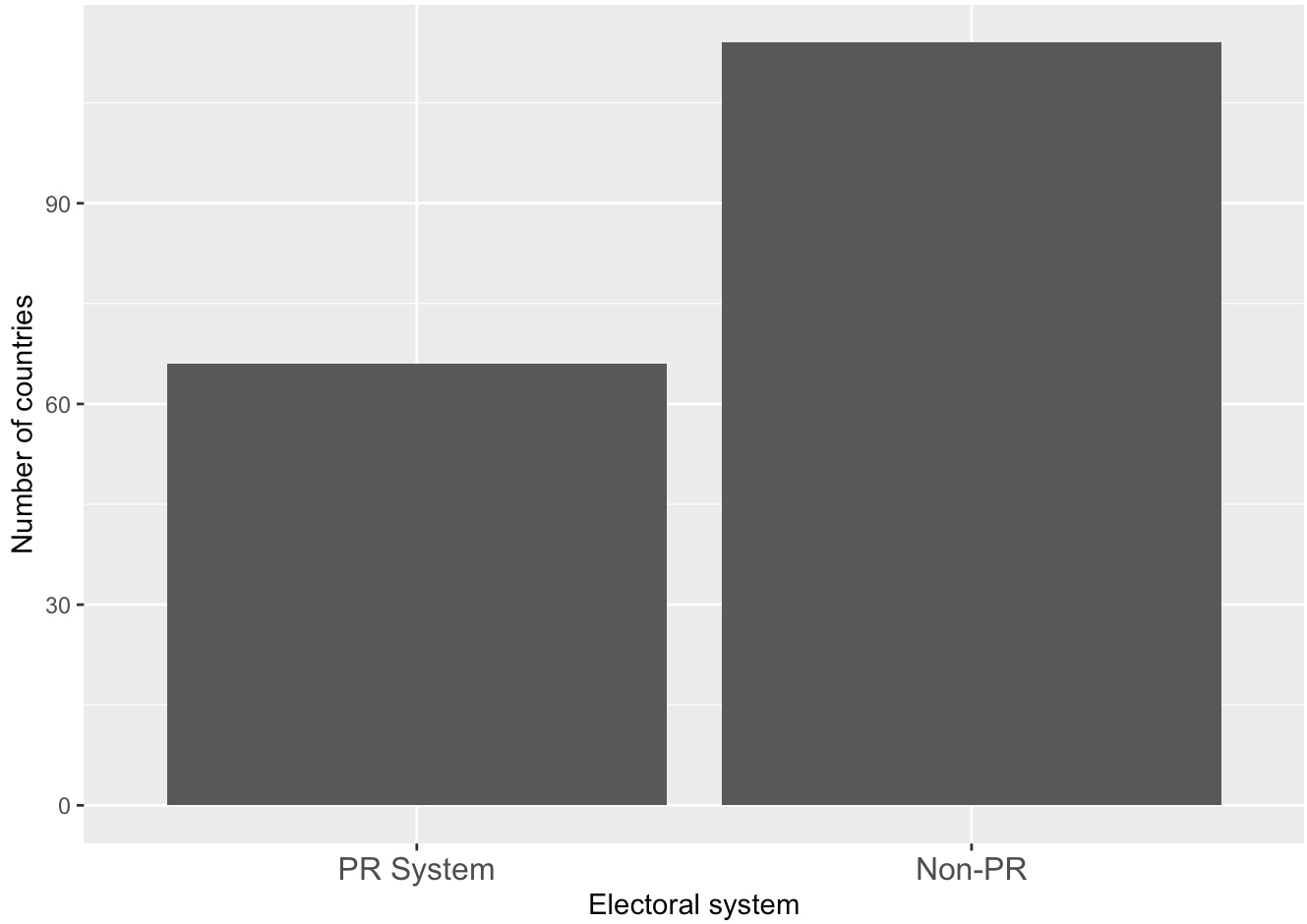
The graph shows that Y is a continuous variable.

Histogram of X

```
#=============================================
# Describe X (numerically and graphically)
#=============================================

# Numerical summary (frequency table)
table(women.pr$pr)
```

```
##
## PR System    Non-PR
##        66       114
```

```
# Graphical summary (bar chart)
g <- ggplot(women.pr, aes(x = pr)) + geom_bar()
g <- g + theme(axis.text.x = element_text(size = 12))
g <- g + xlab("Electoral system") + ylab("Number of countries")
g
```



The graph shows that X is a binary variable (only takes two values, PR/non-PR).

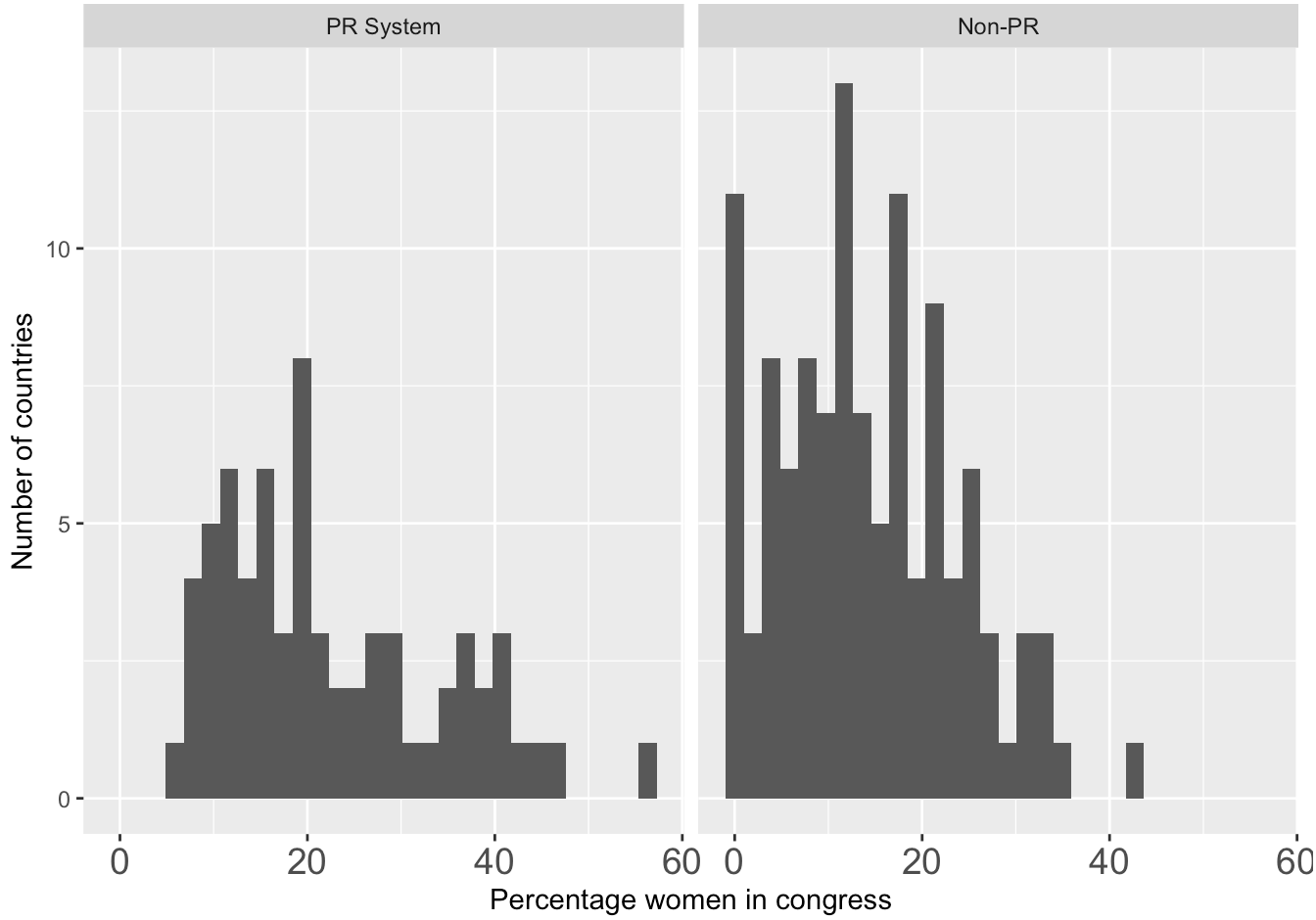
Bivariate analysis

Now that we have done univariate analyses (i.e., describe x and y individually), we will now do a bivariate analysis (i.e., describe the x-y relationship).

As I said, what we will do is to compare the values of Y (women09) for different values of X (PR or Non-PR). That is, we will
1) draw histograms of Y for PR and Non-PR, and
2) calculate statistics (mean, sd, etc.) for PR and Non-PR.
Let's first do this graphically.

1) Bivariate Histogram

```
##================================
## Describe X-Y graphically (When your x variable is binary)
##================================

# Histograms, arranged horizontally

# We will just add one line of code to the code we have already written
# above. That is,

#-------- The following are the same code from before:
g <- ggplot(women.pr, aes(x = women09)) + geom_histogram()
g <- g + theme(axis.text.x = element_text(size = 14))
g <- g + xlab("Percentage women in congress") + ylab("Number of countries")
#-------- The above are the same code from before:

# We add this:
g <- g + facet_grid(.~ pr) # facet plots into grids by this variable (pr)
g
```



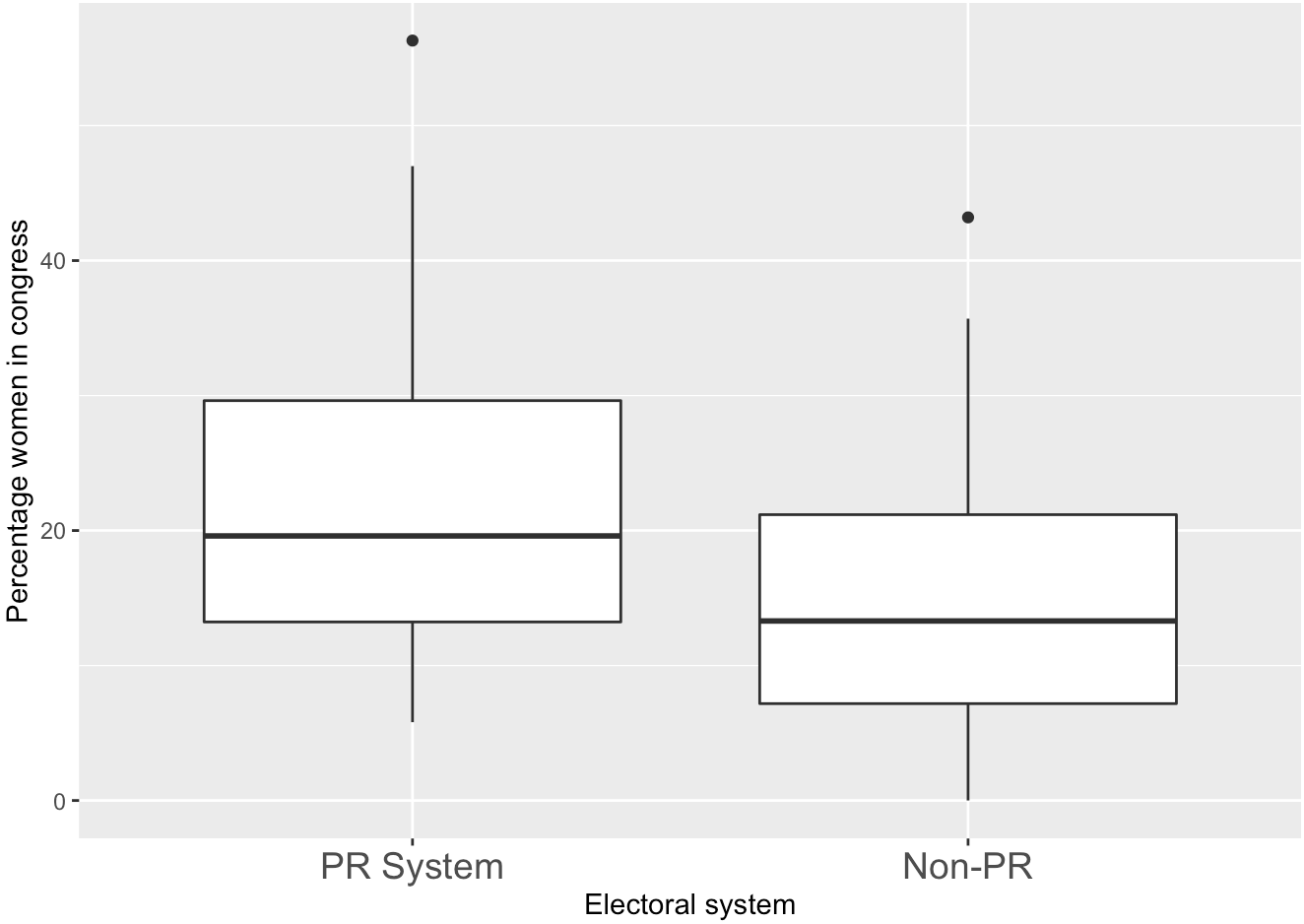
What facet is doing is that it partitions a plot into a matrix of panels. Each panel shows a different subset of the data.

2) Bivariate Boxplot

Comparing the two histograms above (women09 - pr_sys), we can see that women09 (the x axis) take higher values for PR countries than for Non-PR countries, consistent with our expectation.

Another option is to draw a graph called box-whisker plot (or box plot for short) `geom_boxplot()`.

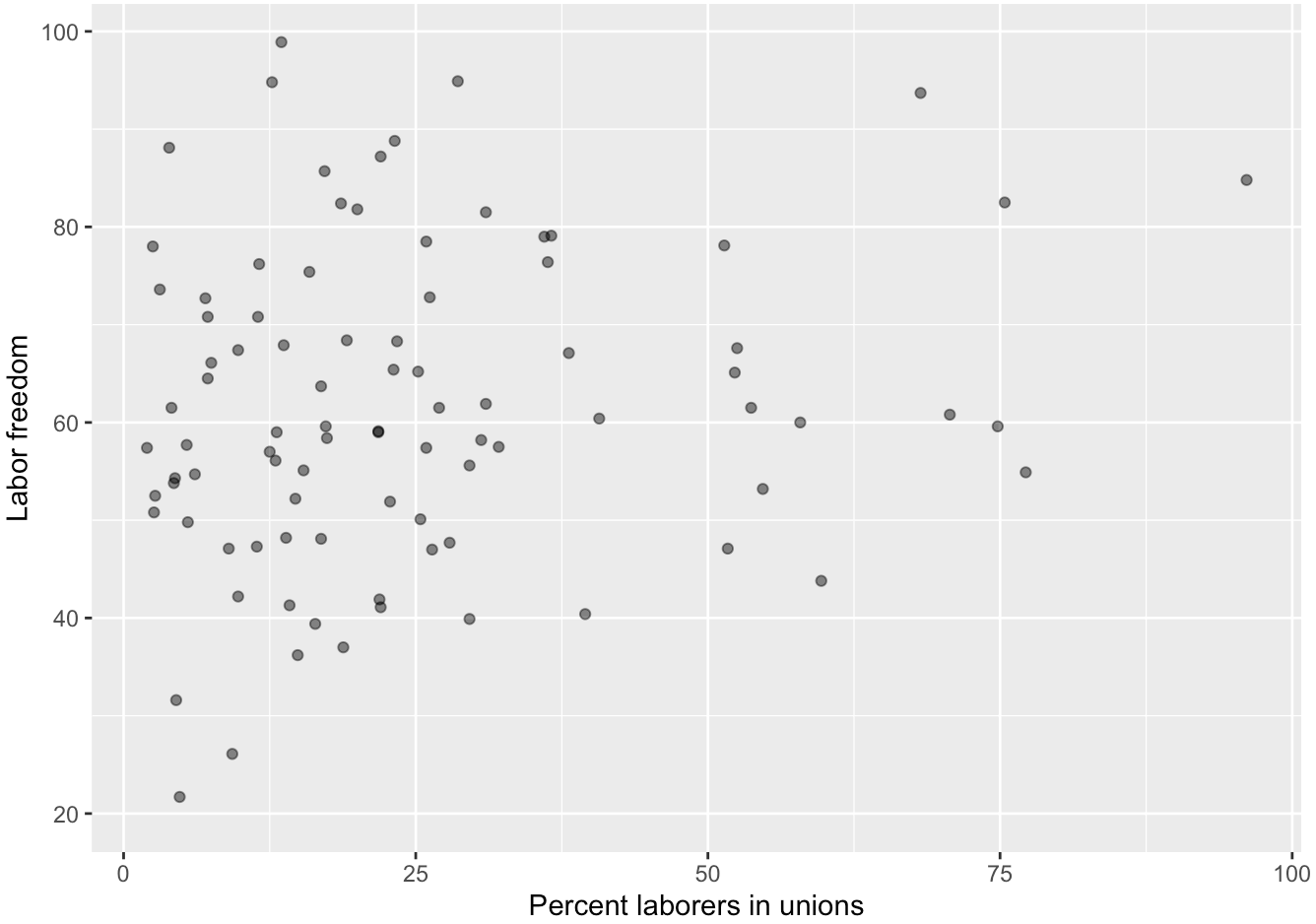
```
g <- ggplot(women.pr, aes(y = women09, x = pr)) # Remember that we need to specify both
y and x here is aes.
g <- g + geom_boxplot()
g <- g + theme(axis.text.x = element_text(size = 14))
g <- g + xlab("Electoral system") + ylab("Percentage women in congress")
g
```



3) Bivariate Scatterplot (When both X and Y are continuous variables)

To describe an X-Y relationship graphically, we can also draw what's called a scatterplot *(`geom_point()`) that shows the values of the X variable on the X-axis and the values of the Y variable on the Y-axis for all observations.

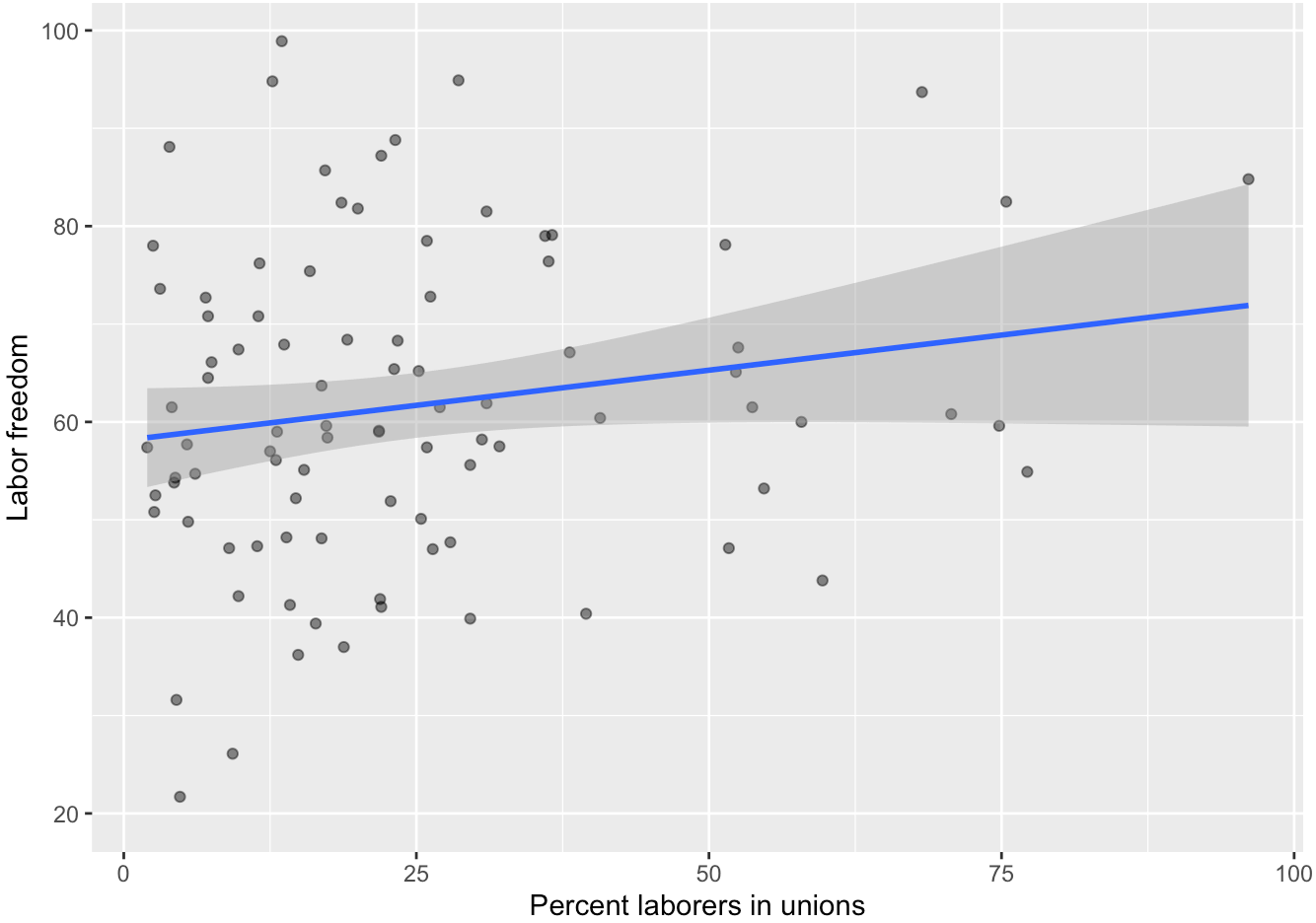
```
#--------------------------------------
# Describe X-Y (When both are continuous variables)
#--------------------------------------

# To create a scatterplot, we use the geom_point function (because
# each observation is denoted by a point.)
ggplot(data =world.data, aes(x = unions, y = free_labor)) +
  ylab("Labor freedom") + xlab("Percent laborers in unions") +
  geom_point(alpha = 0.5)
```



Smoothing - lm

If we want to fit a regression line to see if there is a significant relationship between union density (percentage of workers in a union) and the degree of labor freedom, we can easily use the smooth() function.

```
ggplot(data =world.data, aes(x = unions, y = free_labor)) +
  ylab("Labor freedom") + xlab("Percent laborers in unions") +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm") # lm stands for linear regression models
```



The relationship doesn't look substantial because the change in X doesn't lead to much change in Y. The gray-shaded area represents the confidence interval.

Visualizing cross-sectional data

Lastly, we often come across cross-sectional data, and the one we often see are countries. For example, how many civil war are we have observed in a country-year? What's the best way to visualize your cross-sectional data. The answer is mapping it out.

It turns out that ggplot also gives you a ton of flexibility to drawing map.

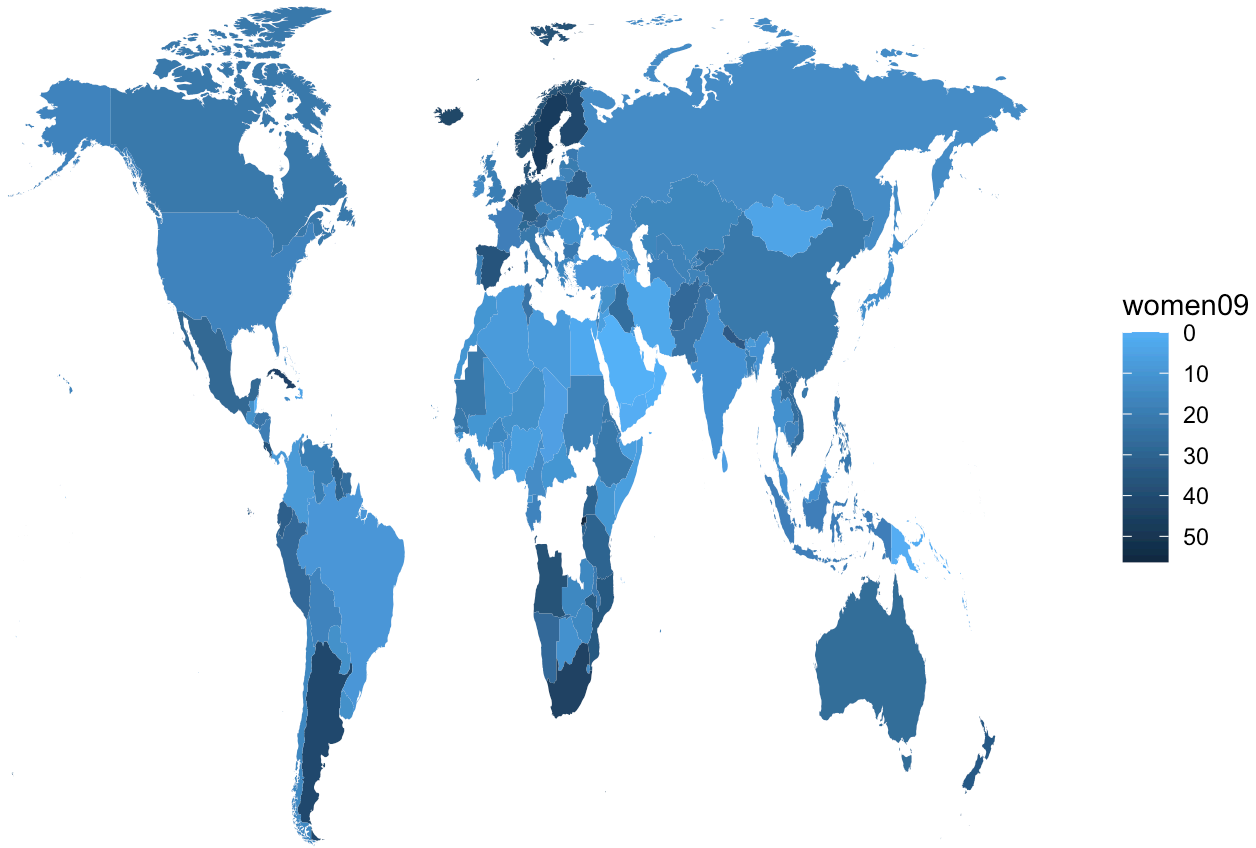
To draw a map, we need two pieces of information:
1) we need coordinates (longitudes and latitudes) to draw shapes (countries, regions, cities, electoral, districts, etc.).
2) we need variables to be graphed.

Let's first get country coordinates, which ggplot has provided already. Additionally, we need to merge the coordinates back to our `world.data` so we can map the variable we want, which is women presentation in congress.

```
# ggplot
map.world <- map_data(map="world")

# Fix unmatched spelling between two datasets
map.world$region[map.world$region == "UK"] = "United Kingdom"
map.world$region[map.world$region == "USA"] = "United States"

# We can convert the country names to characters.
world.data$country = world.data$country %>% as.character()

# merge our world.data with the map.data
map.world_merged = left_join(map.world, world.data, by = c("region" = "country") )
```

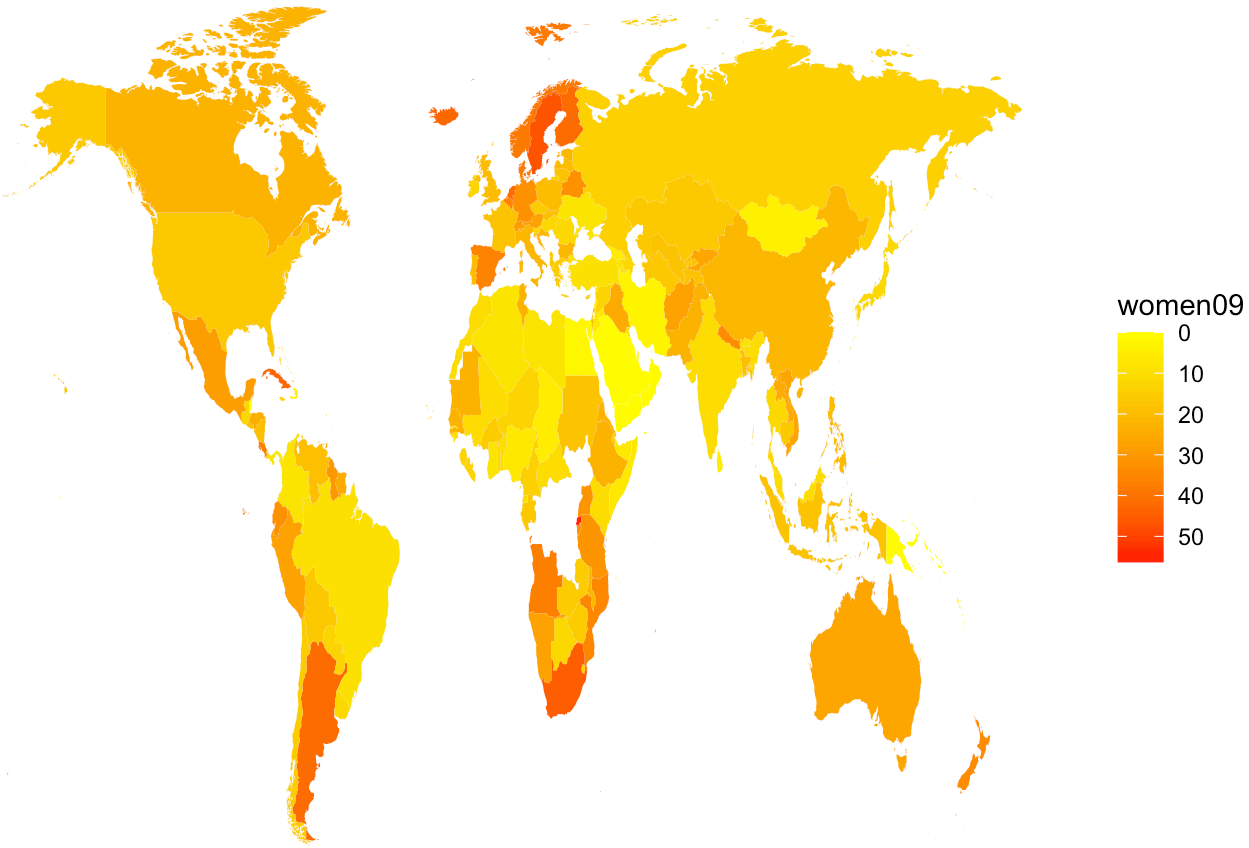
After merging is done, we have both the coordinates and the variable to plot. Let's plot it out then.

```
# We remove Antarctica
map.world_merged = map.world_merged %>% filter(. , region != "Antarctica")

# We remove country with NA values in women09
map.world_merged = map.world_merged[!is.na(map.world_merged$women09),]

ggplot(map.world_merged, aes(x= women09)) +
  theme(legend.title=element_text(size=6), # change legend title font size
        legend.text=element_text(size=8)) + # change legend title font size
  geom_map(data= map.world_merged, map= map.world_merged, aes(map_id=region, x=long, y=l
at, fill=women09)) + # fill the color by a variable
  xlab("") + ylab("") + # remove labs
  theme_bw() + # black and white theme
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + # remove everything in x-axis
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + # remove everything in y-axis
  theme(panel.border = element_blank()) + # remove the black border
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) + # remo
ve the grids
  scale_fill_continuous(trans = 'reverse') #+ # reverse the color ordering
```



If we want to specify colors, we can use this command `scale_fill_gradientn()` :

```
ggplot(map.world_merged, aes(x= women09)) +
  theme(legend.title=element_text(size=6),
        legend.text=element_text(size=8)) +
  geom_map(data= map.world_merged, map= map.world_merged, aes(map_id=region, x=long, y=l
at, fill=women09)) + # fill the color by a variable
  xlab("") + ylab("") + # remove labs
  theme_bw() + # black and white theme
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + # remove everything in x-axis
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) + # remove everything in y-axis
  theme(panel.border = element_blank()) + # remove the black border
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) + # remo
ve the grids

  scale_fill_gradientn(colours = c("red", "yellow"), trans = 'reverse') # use gradient t
o determine the color scale you want
```



Future Topics in (Big-)Data Science and Programming

Getting data down:

- Webscraping and Regular expressions (text data)
- Web APIs (Twitter API etc)
- Storage: SQL (optional)

Modeling: Machine learning and other deep learning models

- Text Data
- Network Data
- Spatial Data
- Image (optional)
- Audio (optional)

Other important topics:

- Github (this is the basics)
- Markdown and Rmarkdown for presentation

Great! We've finished the lecture and you can go to day5 exercise to do some additional practices for today's content.