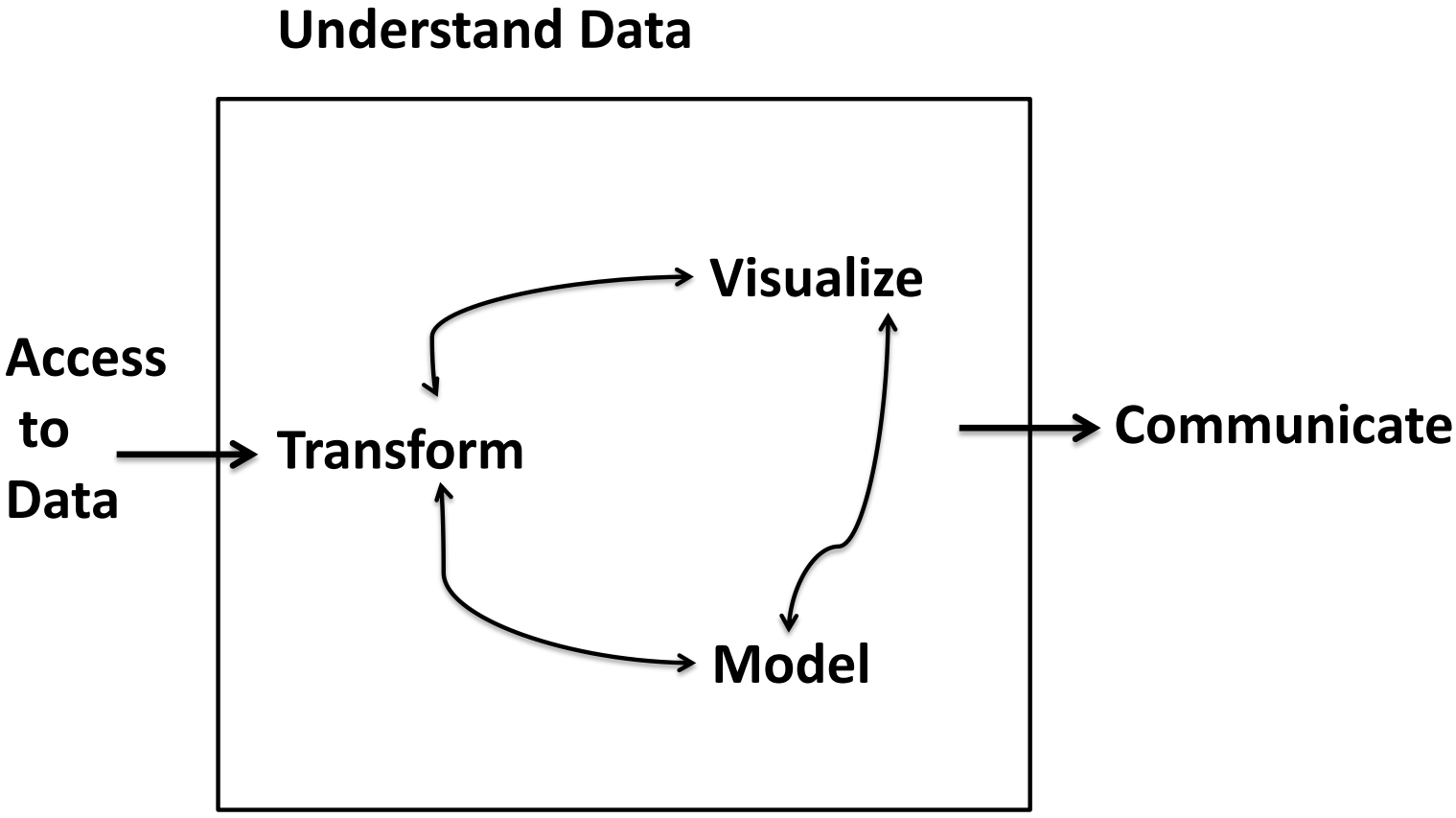
EM622 Data Analysis and Visualization Techniques for Decision-Making

Basic Graphs

Agenda

1. Data Exploration Process
2. Scatterplots
3. Enhacend Scatterplots
4. Scatterplot Matrix(SPLOM)
5. Faceting

Data Exploration [1]



[1] Hadley Wickham. Engineering Data Analysis. Google tech-talk

Data Exploration

"Visualization is great for revealing the unexpected, but it does not scale very well. If you have 100 thousand variables, you can't look at every single scatterplot . that is when models come in. Models give a nice scalable approach to data analysis. They provide a way of passing the hard computations from our head to a computer. But the problem with models is that they won't tell you something you really did not expect" - Hadley Wickham

Ben Schneiderman's Mantra[2]

1)Overview First, 2) Zoom & Filter, 3) Details on Demand

[2]Shmueli & Hadoorn. INFORMS Data Exploration course

ggplot2

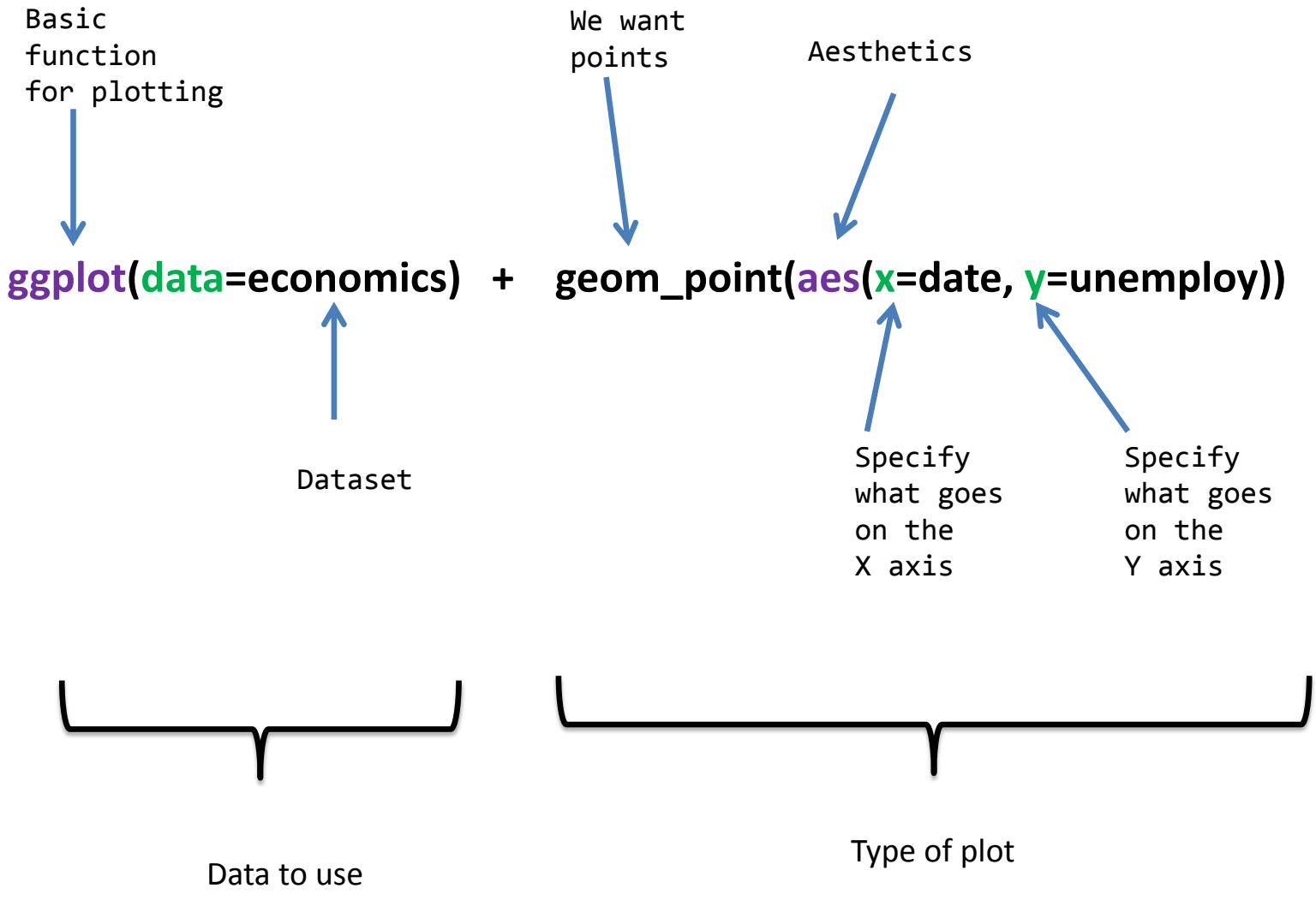
- ggplot2 is an R package designed for creating high quality plots.
- ggplot is based on the layered gramar of graphics, which means that plots can be constructed layer by layer [3].

```
#you need to install the package just once
install.packages('ggplot2')
```

[3] http://vita.had.co.nz/papers/layered-grammar.pdf

Composition of plots in ggplot2

Plots have two main components: 1) data to use and 2) type of plot.



Case Study: *economics dataset*

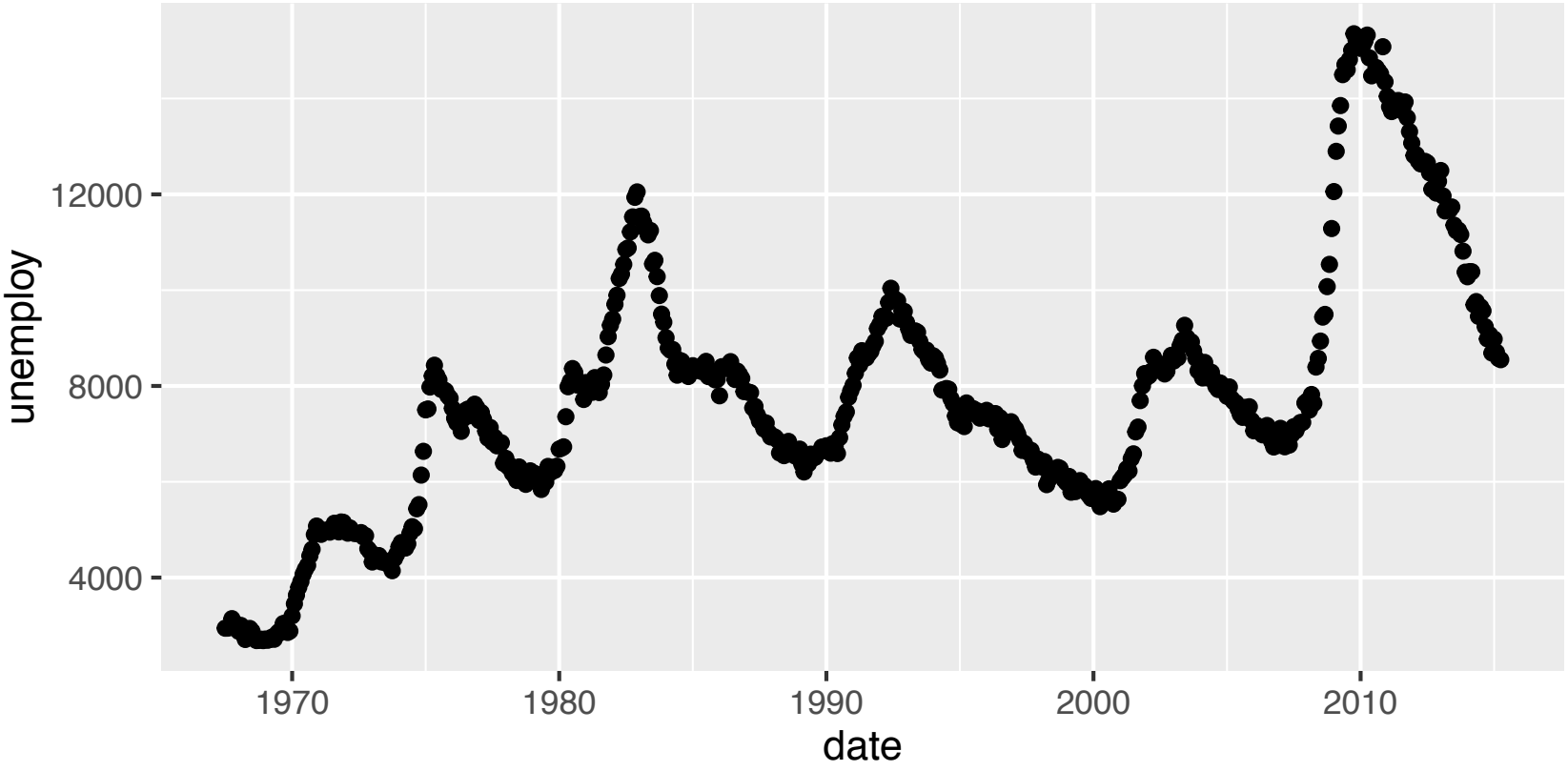
The View command receives as input the name of a dataset and displays it in a similar way as the data you usually store in excel (rows and columns).

```
View(economics) #view
```

	date	pce	pop	psavert	uempmed	unemploy
1	1967-06-30	507.8	198712	9.8	4.5	2944
2	1967-07-31	510.9	198911	9.8	4.7	2945
3	1967-08-31	516.7	199113	9.0	4.6	2958
4	1967-09-30	513.3	199311	9.8	4.9	3143
5	1967-10-31	518.5	199498	9.7	4.7	3066
6	1967-11-30	526.2	199657	9.4	4.8	3018
7	1967-12-31	532.0	199808	9.0	5.1	2878
8	1968-01-31	534.7	199920	9.5	4.5	3001
9	1968-02-29	545.4	200056	8.9	4.1	2877
10	1968-03-31	545.1	200208	9.6	4.6	2709
11	1968-04-30	550.9	200361	9.3	4.4	2740
12	1968-05-31	557.4	200536	8.9	4.4	2938
13	1968-06-30	564.4	200706	7.8	4.5	2883
14	1968-07-31	568.2	200898	7.6	4.2	2768
15	1968-08-31	569.5	201095	7.6	4.6	2686
16	1968-09-30	572.9	201290	7.8	4.8	2689

Scatter plot

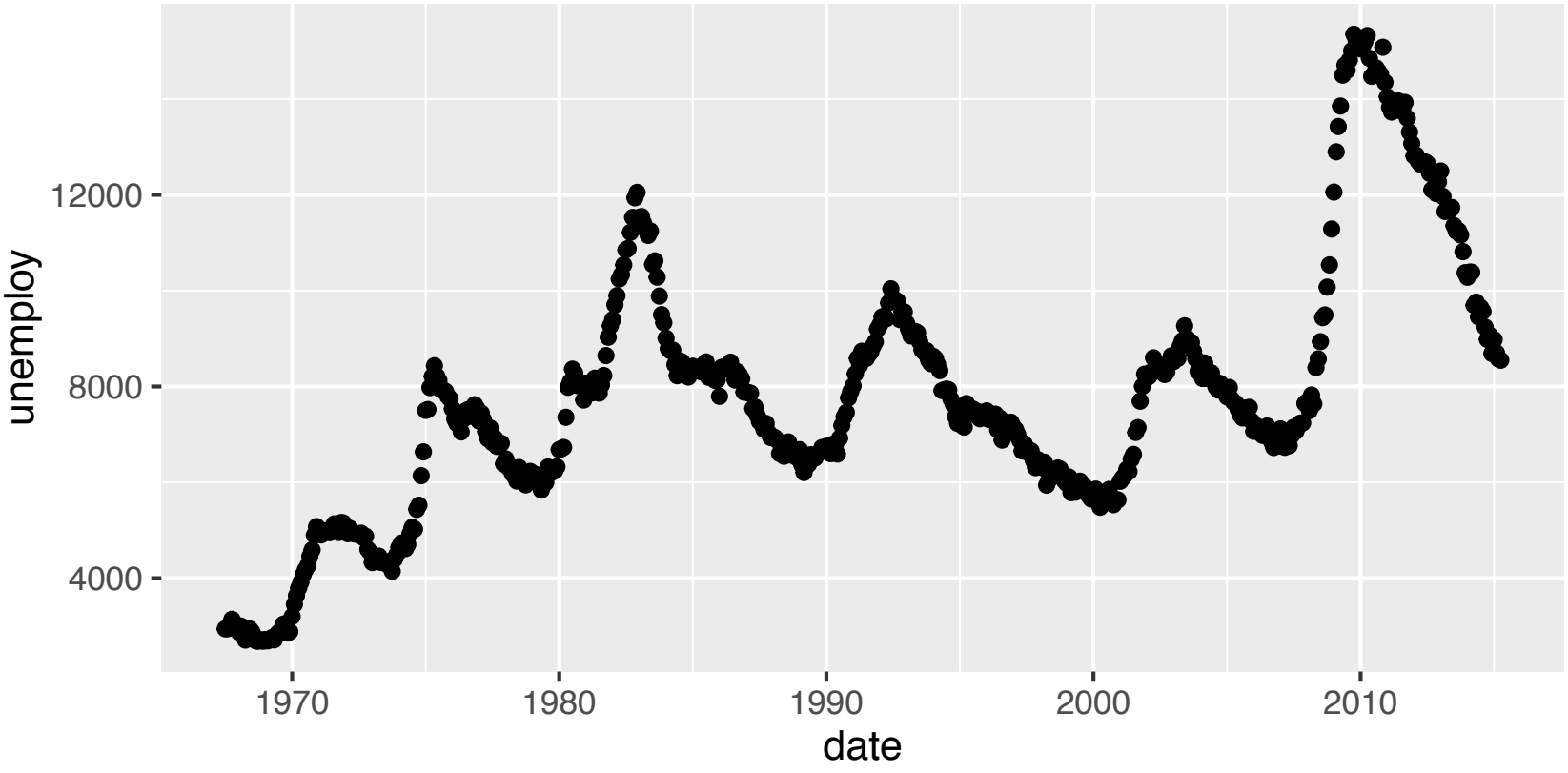
```
#you need to load the library everytime you start an RStudio session
library(ggplot2)
ggplot(data=economics) + geom_point(aes(x=date,y=unemploy))
```



Composition of plots in ggplot2

We can also generate the previous plot using multiple steps

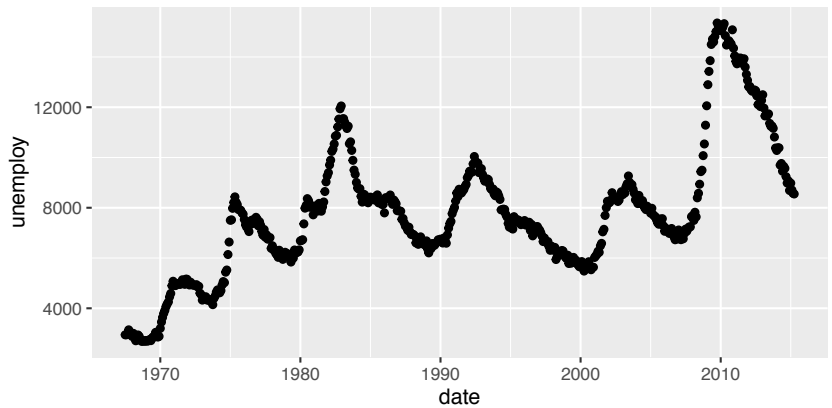
```
myplot <- ggplot(data=economics)
myplot <- myplot + geom_point(aes(x=date,y=unemploy))
myplot
```



Displaying time series - using ggplot2

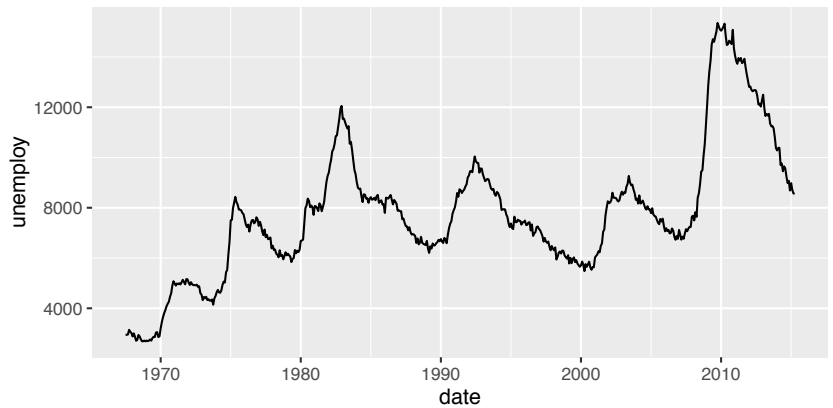
Plotting points

```
myplot <- ggplot(data=economics)
myplot +
geom_point(aes(x=date,y=unemploy))
```



Plotting lines

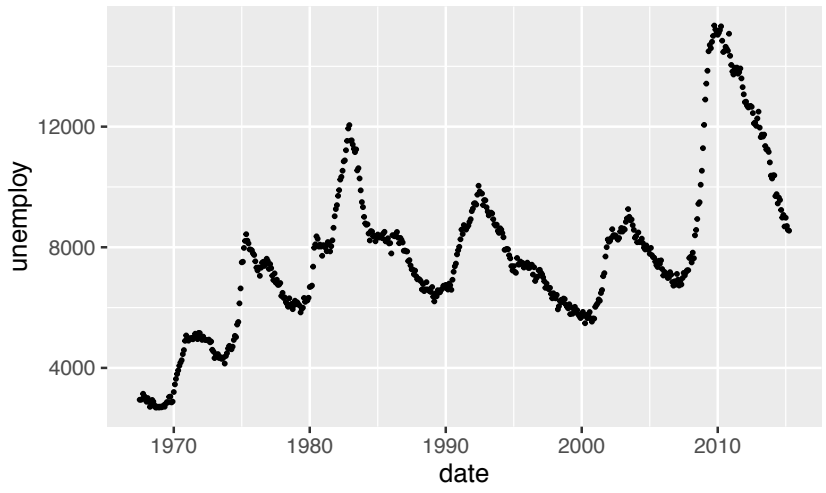
```
myplot <- ggplot(data=economics)
myplot +
geom_line(aes(x=date,y=unemploy))
```



Changing the size of the points

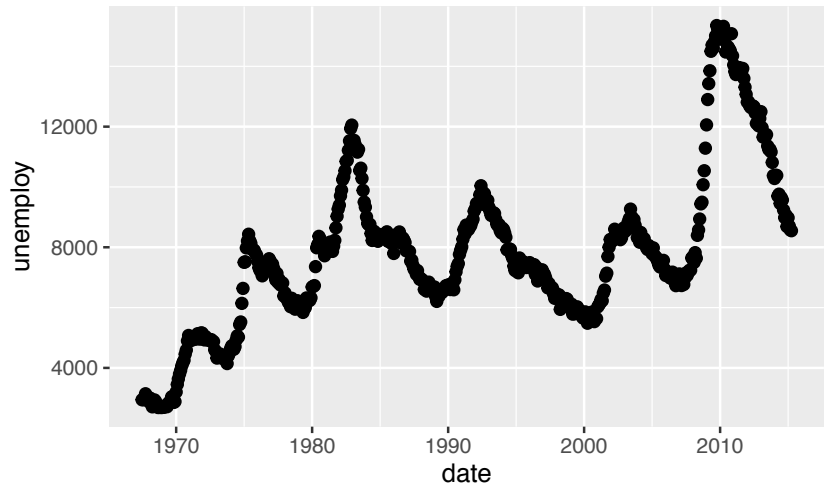
Plotting small points

```
myplot <- ggplot(data=economics)
myplot + geom_point(
  aes(x=date,y=unemploy),size=0.5)
```



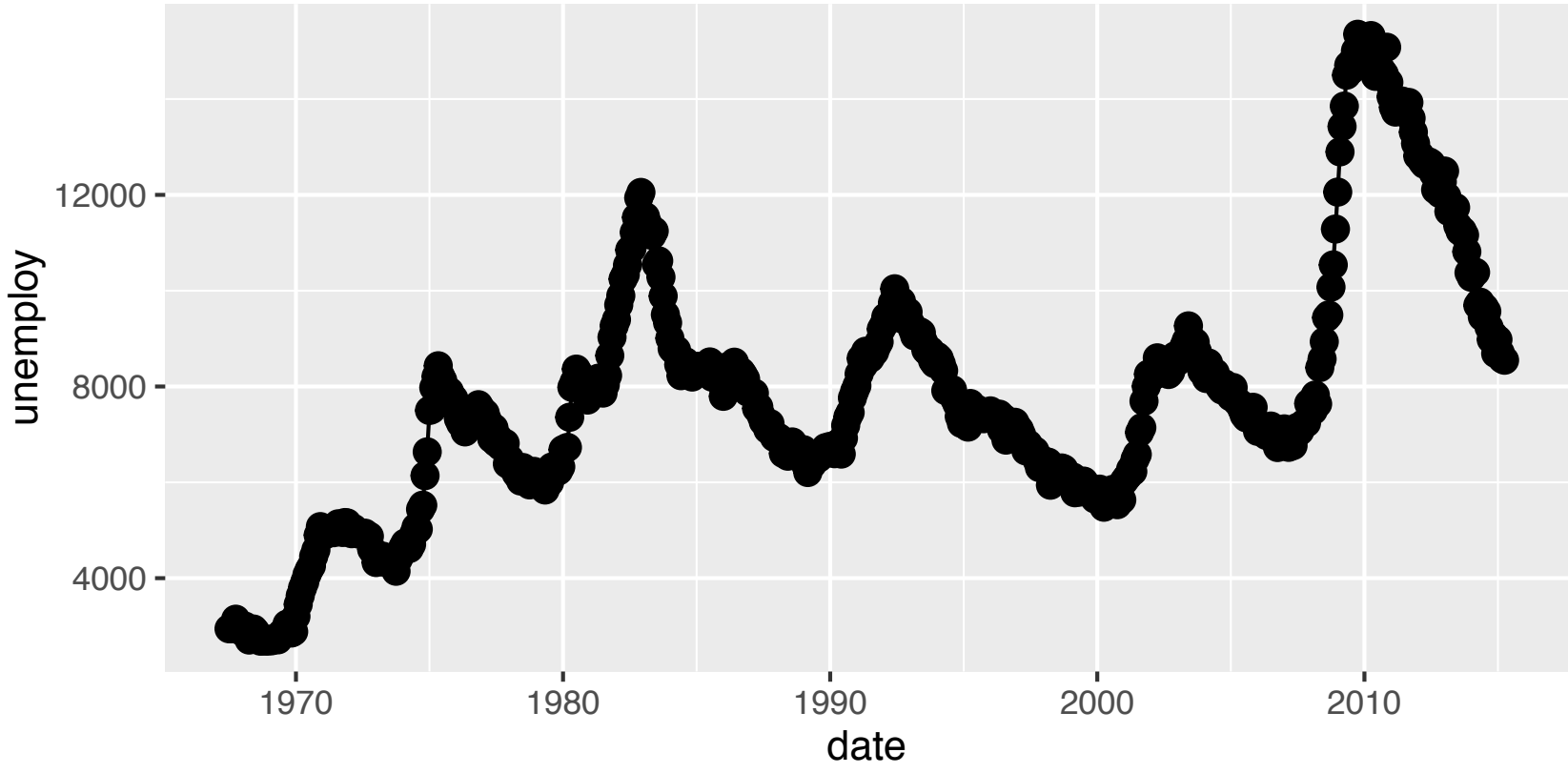
Plotting big points

```
myplot <- ggplot(data=economics)
myplot + geom_point(
  aes(x=date,y=unemploy),size=2.0)
```

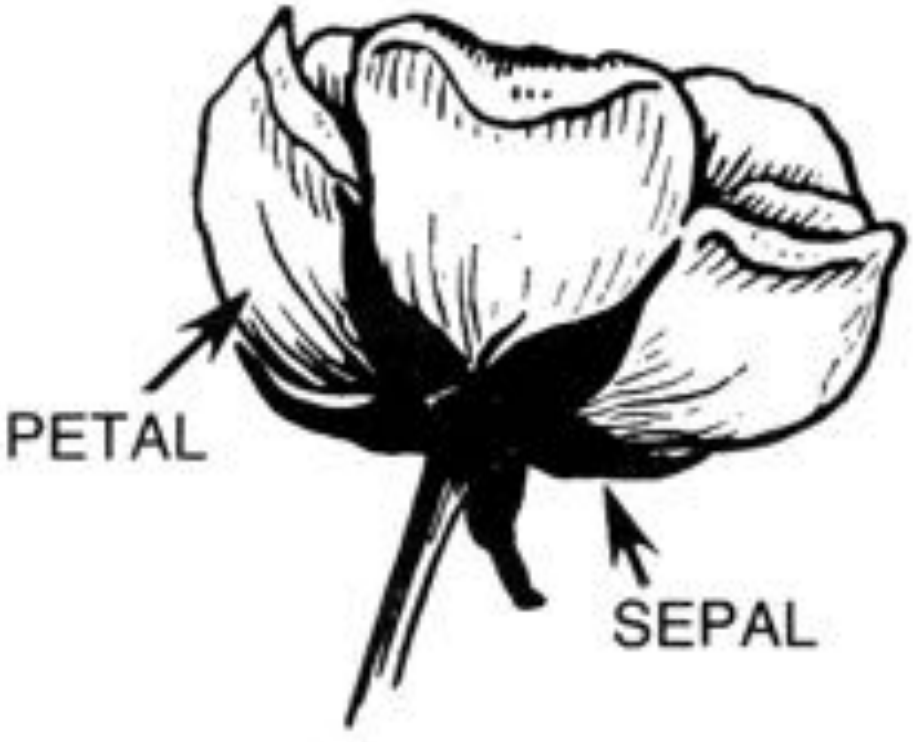


Displaying points and lines

```
myplot <- ggplot(data=economics)
myplot +
geom_point(aes(x=date, y=unemploy),size=3) +
geom_line(aes(x=date, y=unemploy))
```



Case Study: *iris dataset*

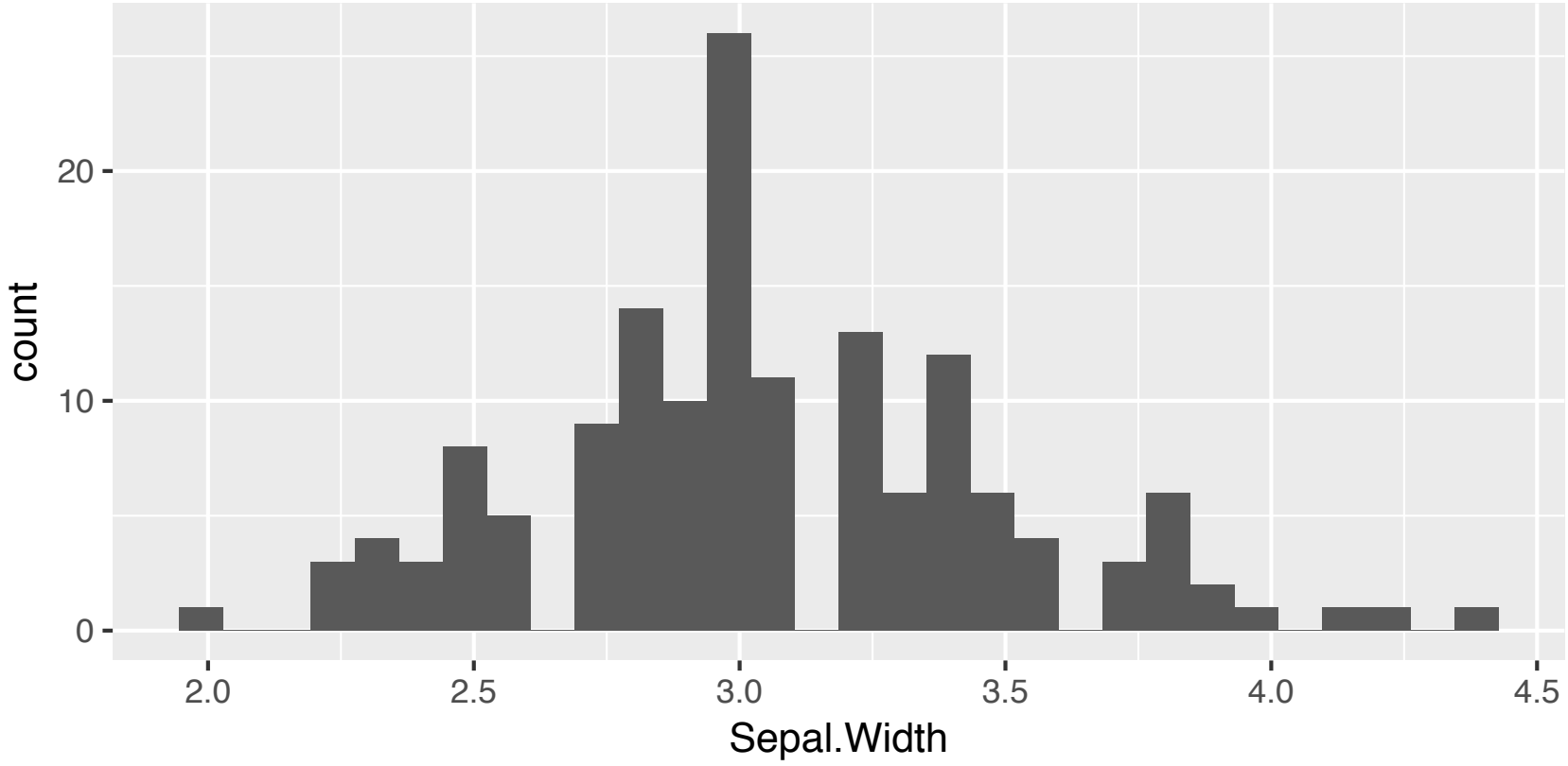


More tools for inspecting the Iris dataset

```
#determine type and quantity of data
str(iris)

## 'data.frame': 150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1

#take a look at the first couple of rows
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

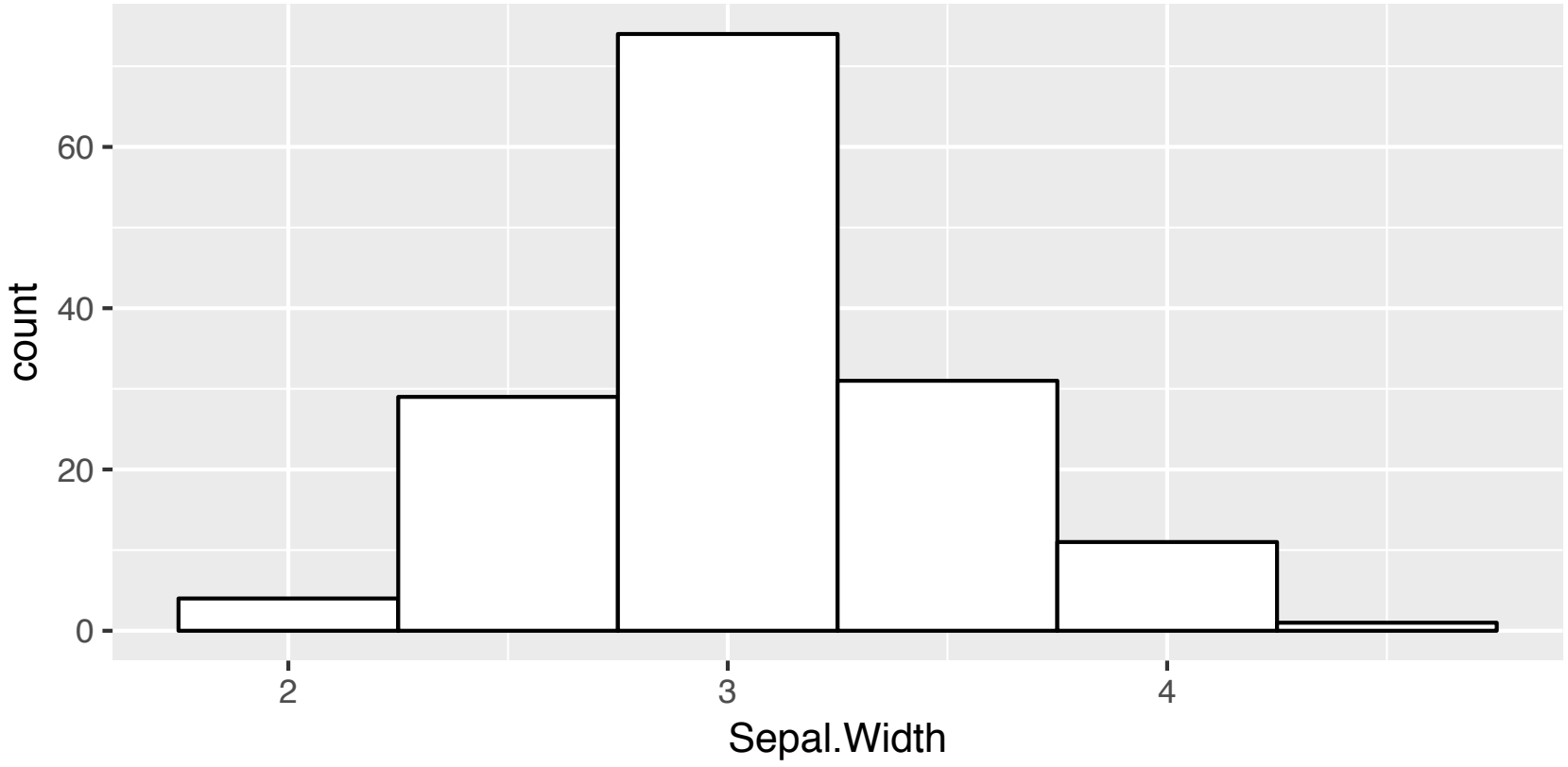
Basic Histogram

```
ggplot(data=iris)+ geom_histogram(aes(x=Sepal.Width))

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



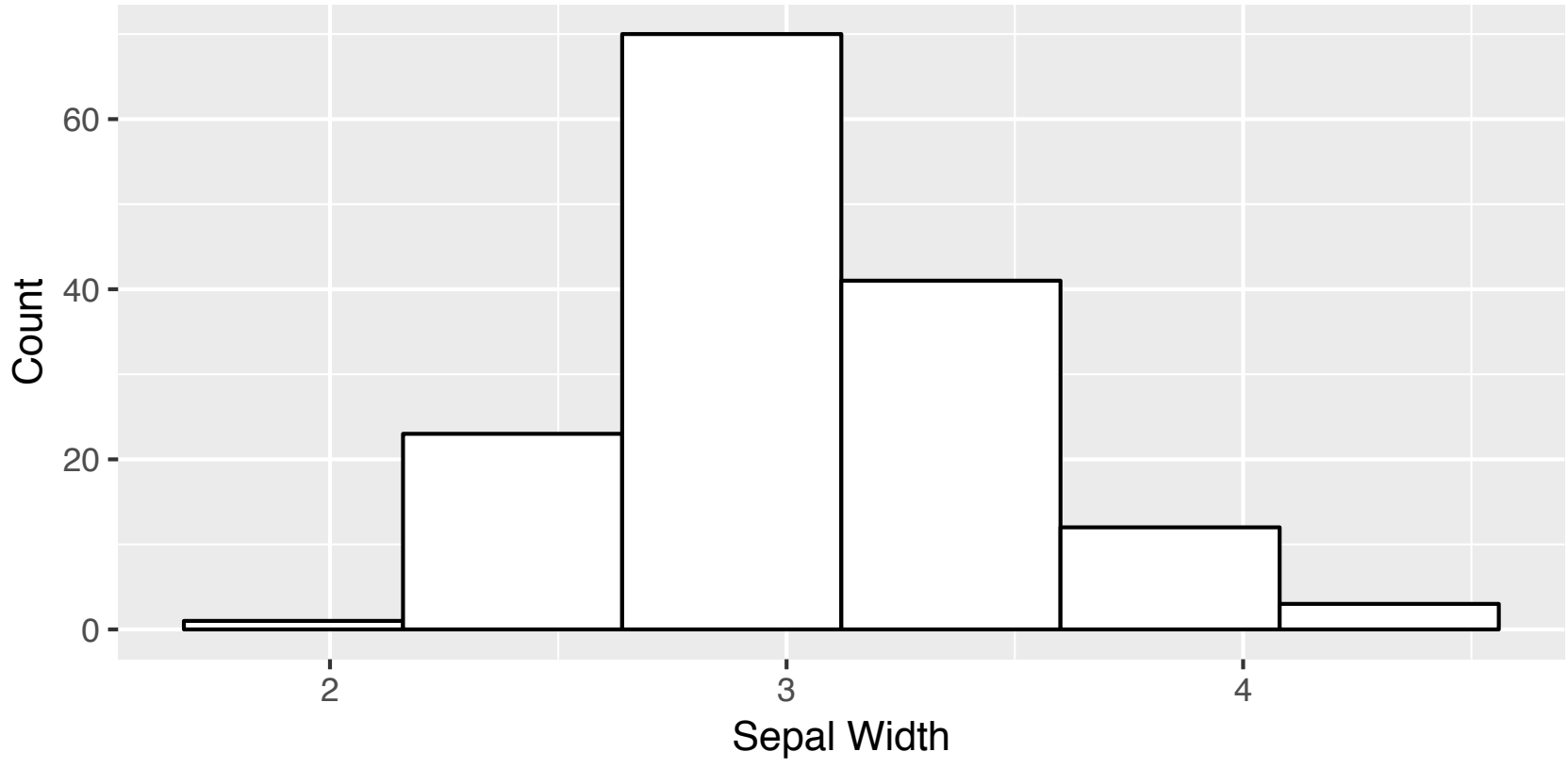
Basic Histogram (Binsize)

```
#Set the width of each bin to 0.5
ggplot(data=iris)+ geom_histogram(aes(x=Sepal.Width),
            binwidth=0.5,fill="white", colour="black")
```



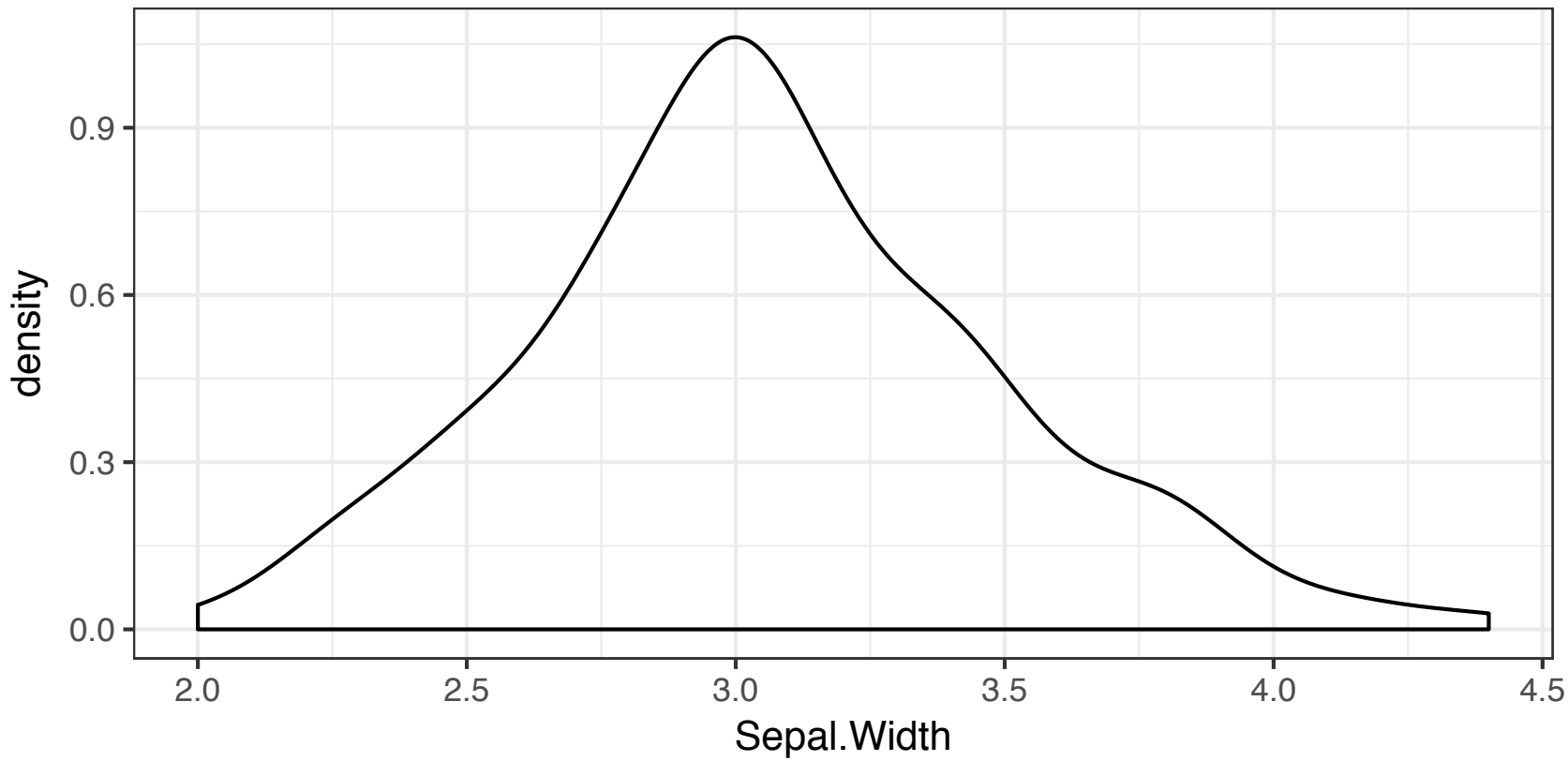
Basic Histogram (Binsize)

```
# Divide the x range into 5 bins
binsize <- diff(range(iris$Sepal.Width))/5
# binsize =0.48
ggplot(data=iris)+ geom_histogram(aes(x=Sepal.Width),
      binwidth=binsize,fill="white", colour="black")+
      xlab("Sepal Width")+ylab("Count")
```



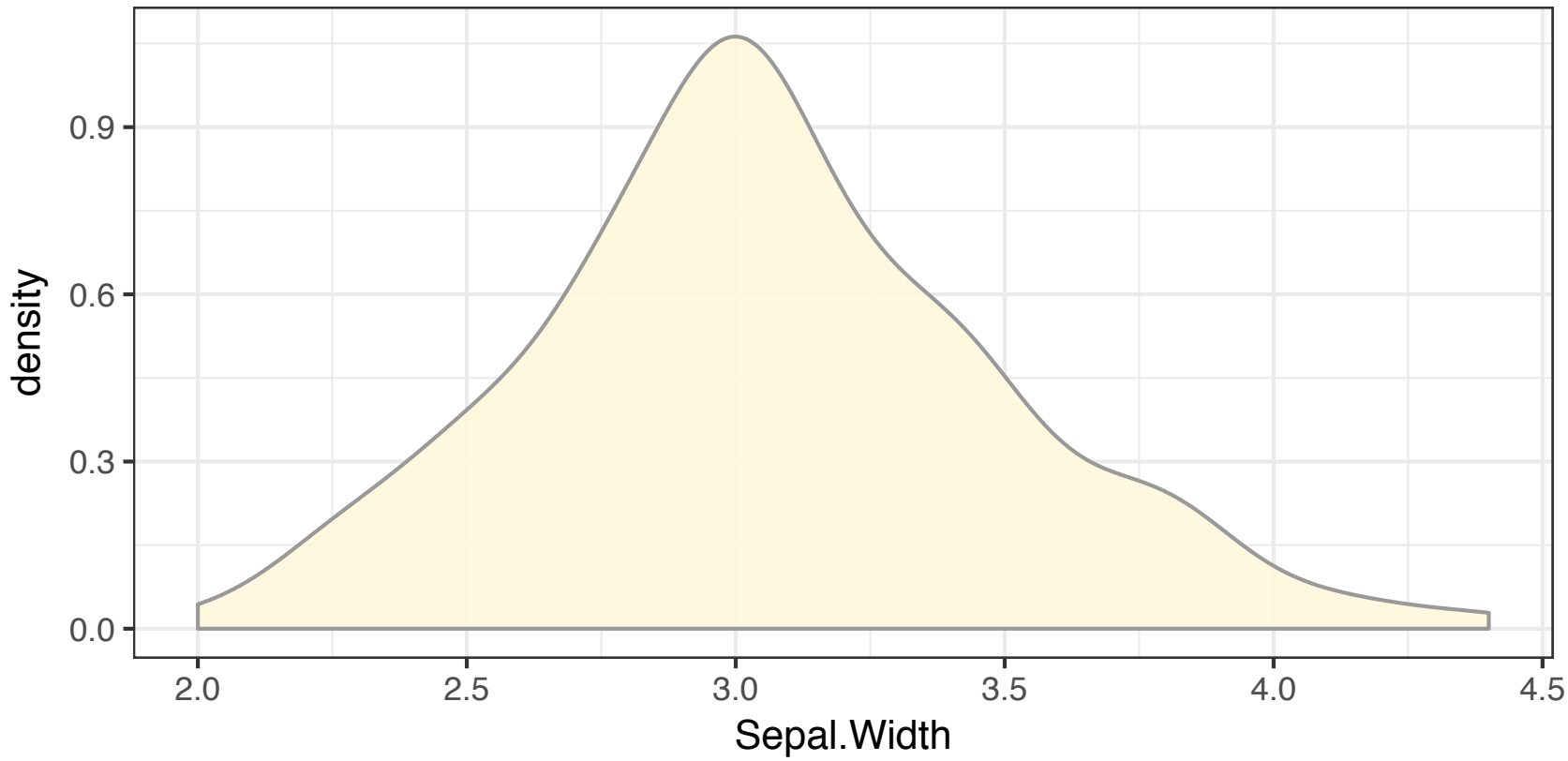
Density Plot

```
ggplot(data=iris,aes(x=Sepal.Width))+geom_density()+theme_bw()
```



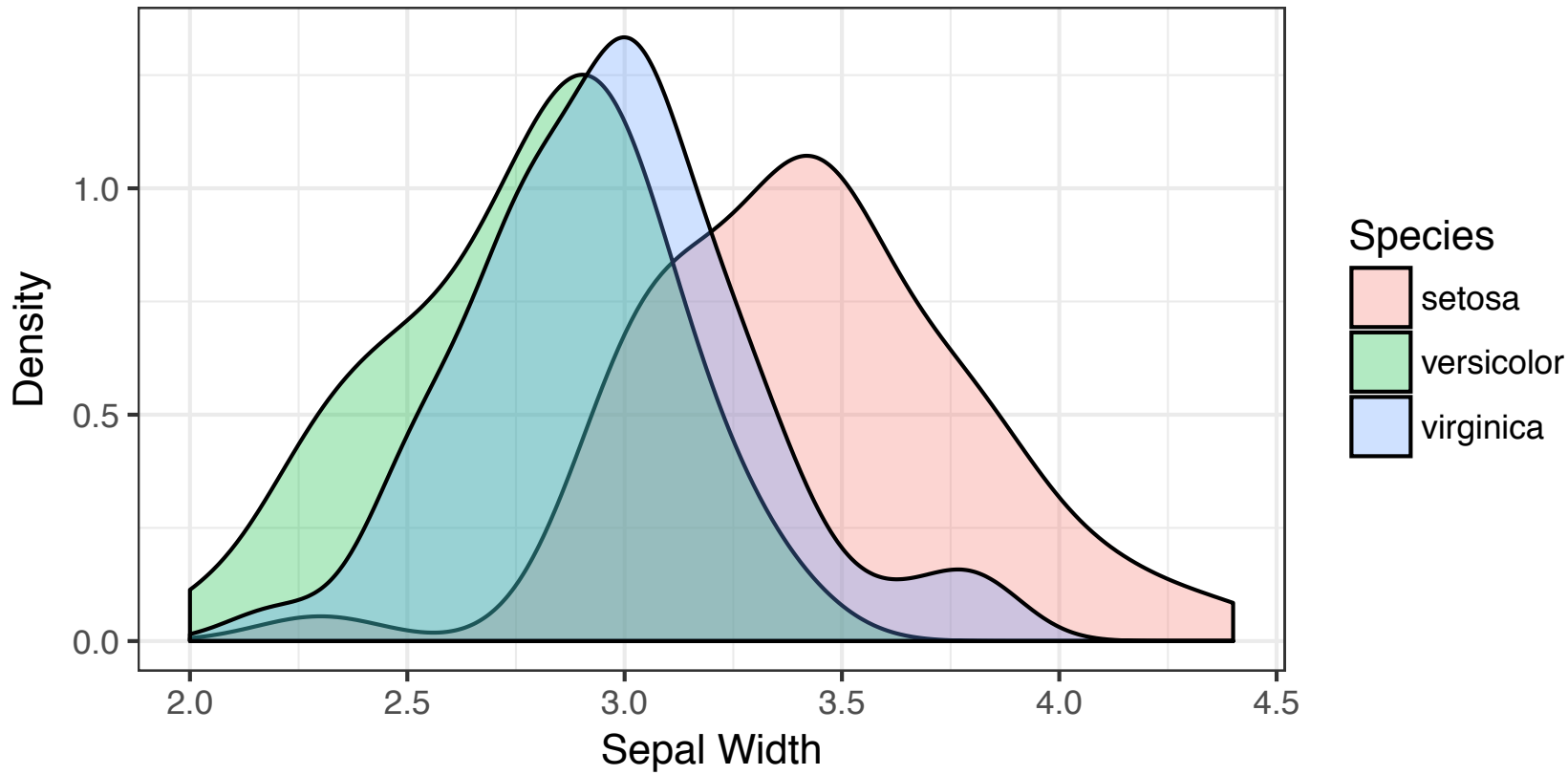
Density Plot (Fill)

```
ggplot(data=iris,aes(x=Sepal.Width))+theme_bw()+
  geom_density(alpha=0.9,fill="cornsilk",colour="grey60")
```



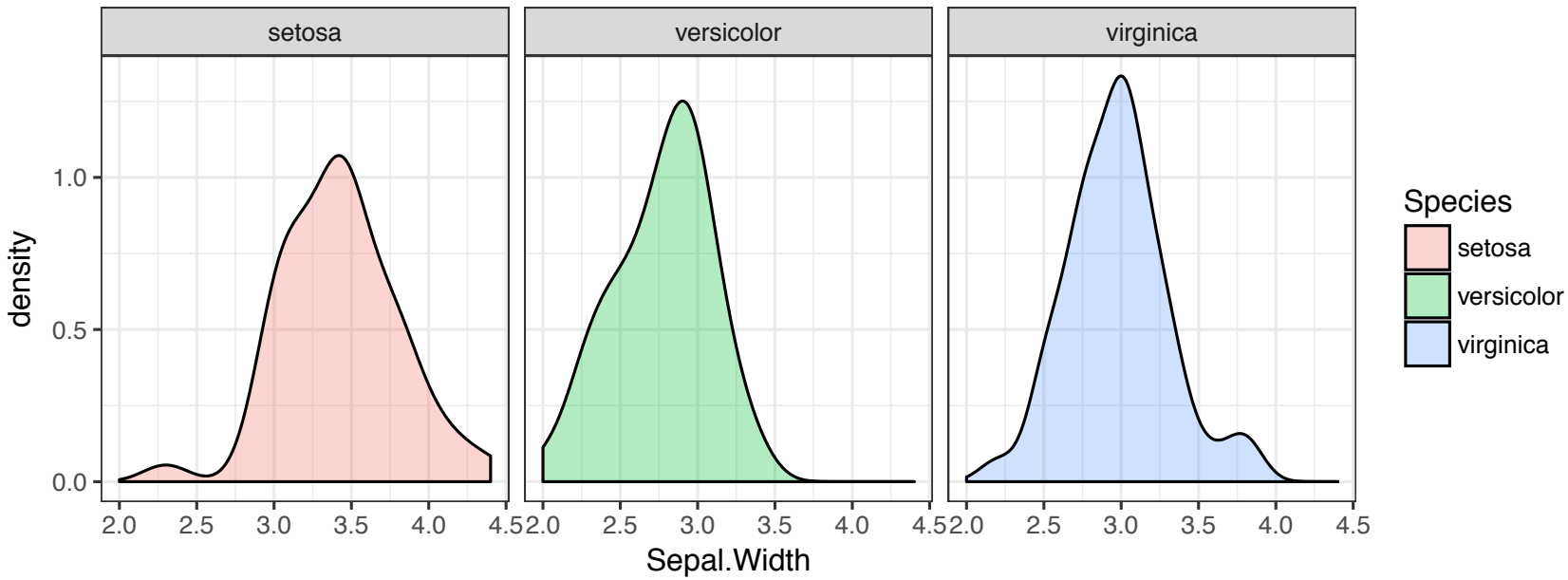
Density Plot (Fill with Color)

```
ggplot(data=iris,aes(x=Sepal.Width,fill=Species))+theme_bw()+
  geom_density(alpha=0.3)+xlab("Sepal Width")+ylab("Density")
```



Density Plot (Facet)

```
ggplot(data=iris,aes(x=Sepal.Width,fill=Species))+theme_bw()+
  geom_density(alpha=0.3)+facet_grid(~Species)
```

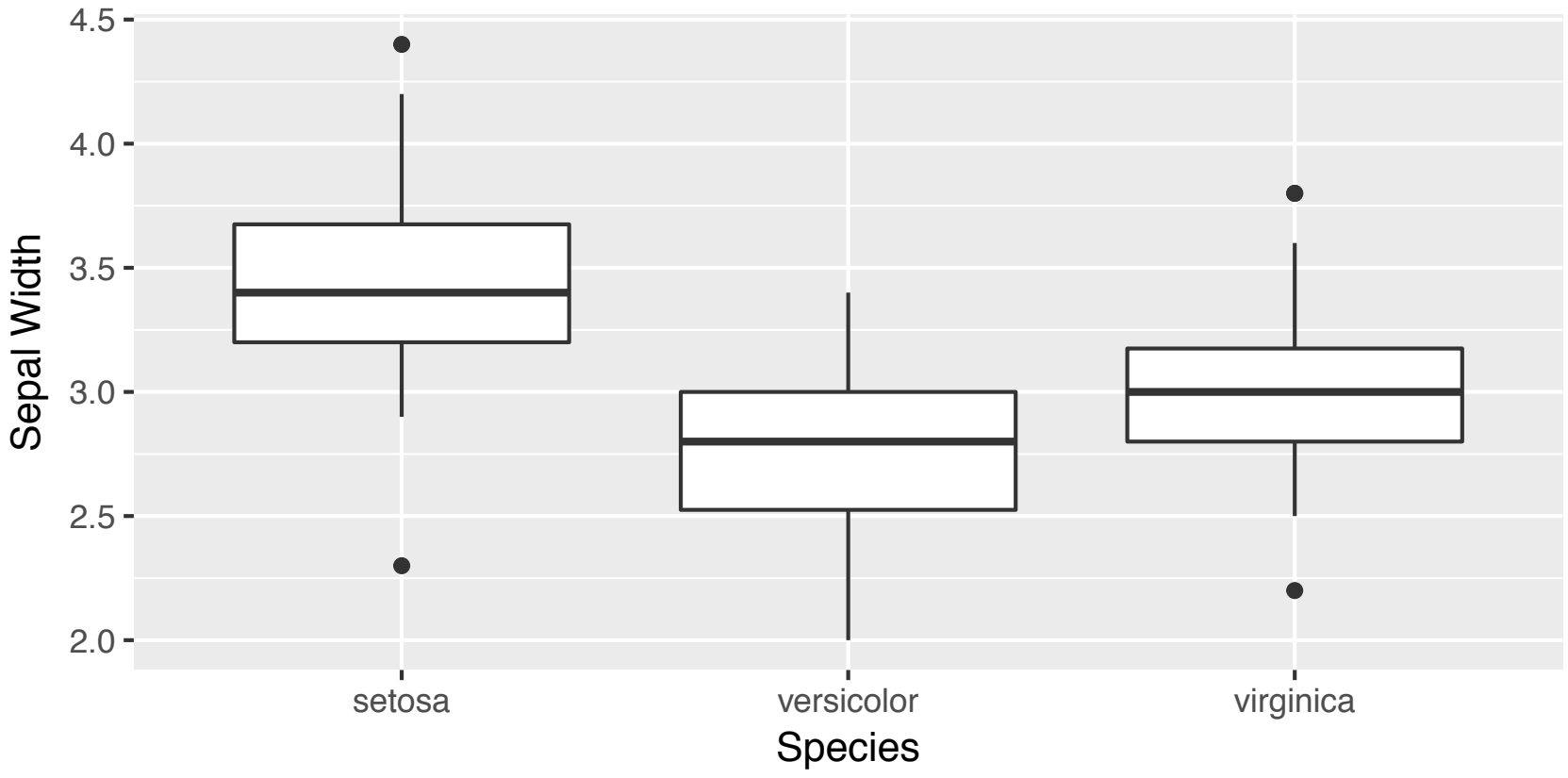


Density Plot + Histogram

```
ggplot(data=iris,aes(x=Sepal.Width,y=..density..))+
  geom_histogram(binwidth=0.5, alpha=0.9,fill="cornsilk",colour="grey60")+
  geom_density()+theme_bw()+xlab("Sepal Width")+ylab("Density")
```

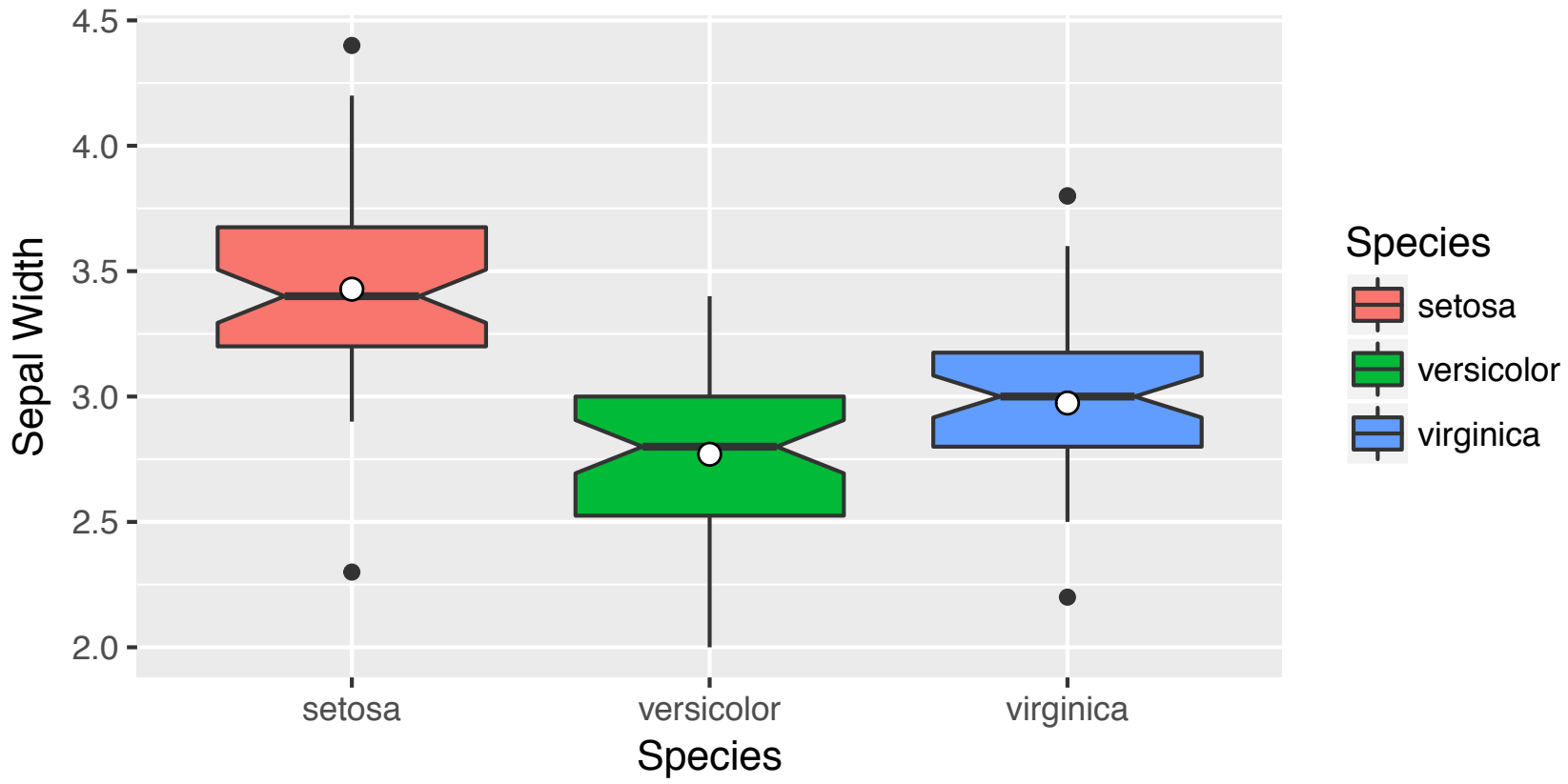
Box Plot

```
ggplot(data=iris,aes(x=Species,y=Sepal.Width))+
  geom_boxplot()+ylab("Sepal Width")
```



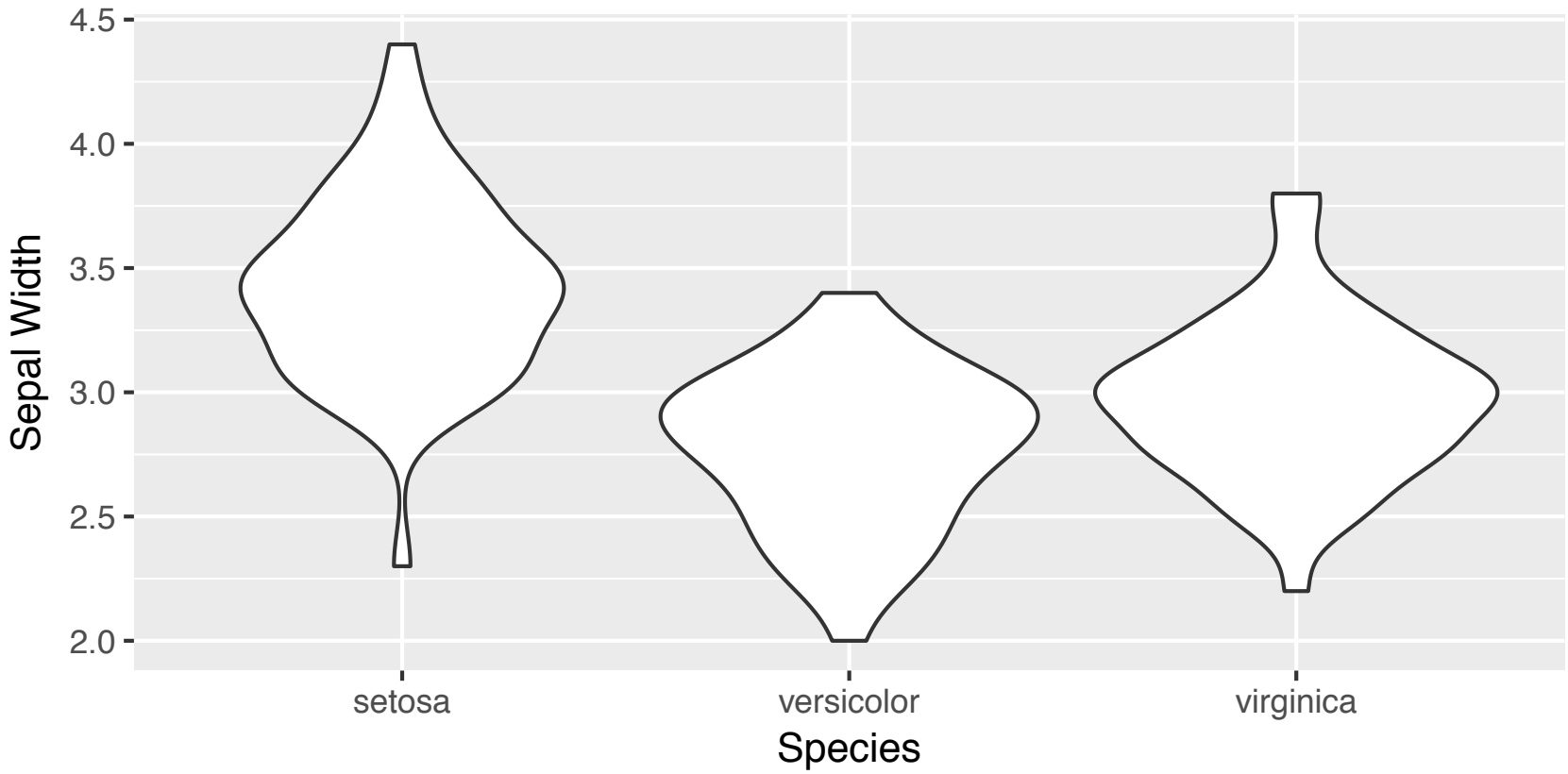
Box Plot (Notch, Fill and Stats)

```
ggplot(data=iris,aes(x=Species,y=Sepal.Width,fill=Species))+
  geom_boxplot(notch=TRUE) +
  stat_summary(fun.y=mean, geom="point", fill="white",
  shape=21, size=2.5)+ylab("Sepal Width")
```



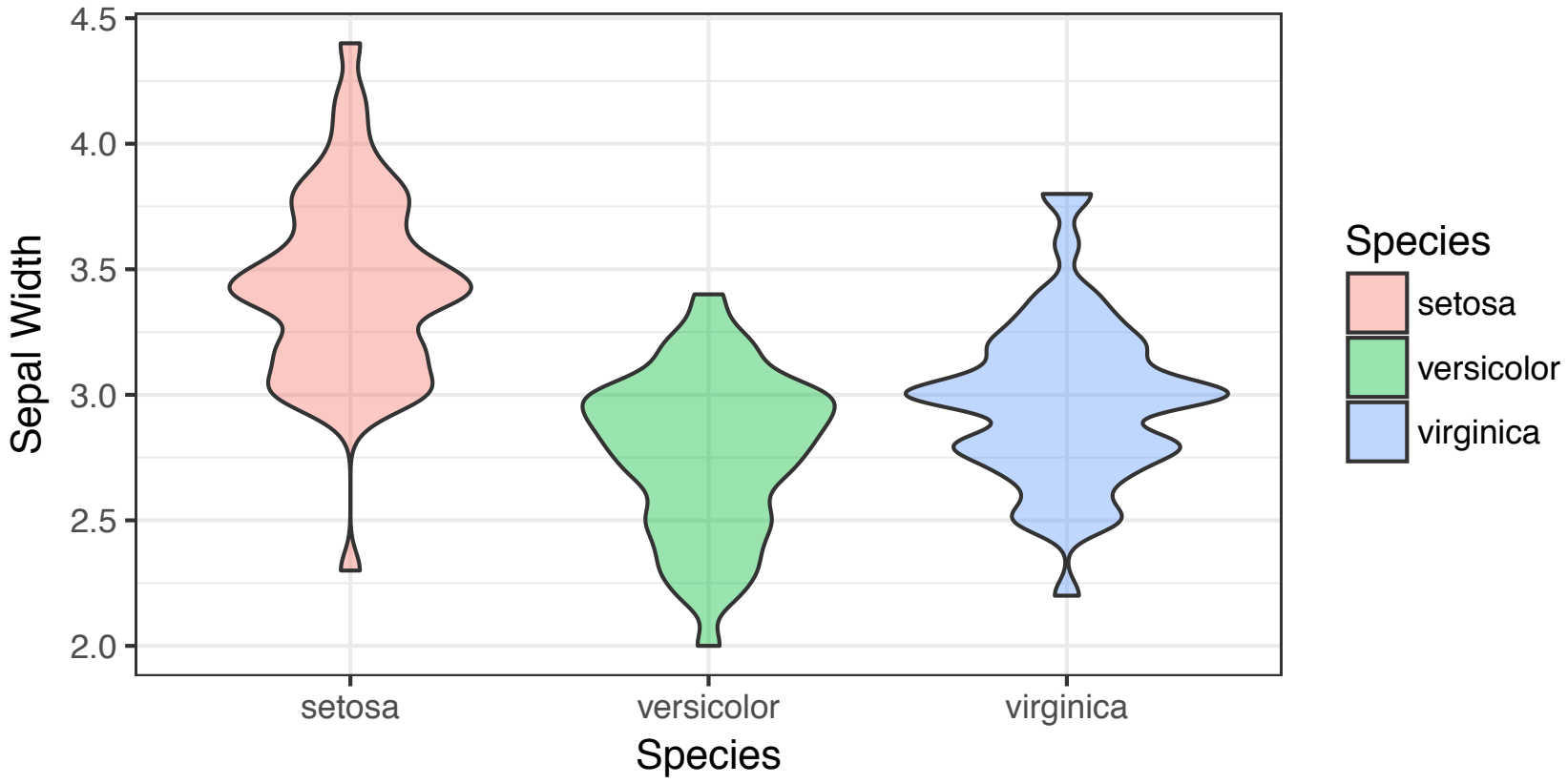
Violin Plot

```
ggplot(data=iris,aes(x=Species,y=Sepal.Width))+
  geom_violin()+ylab("Sepal Width")
```



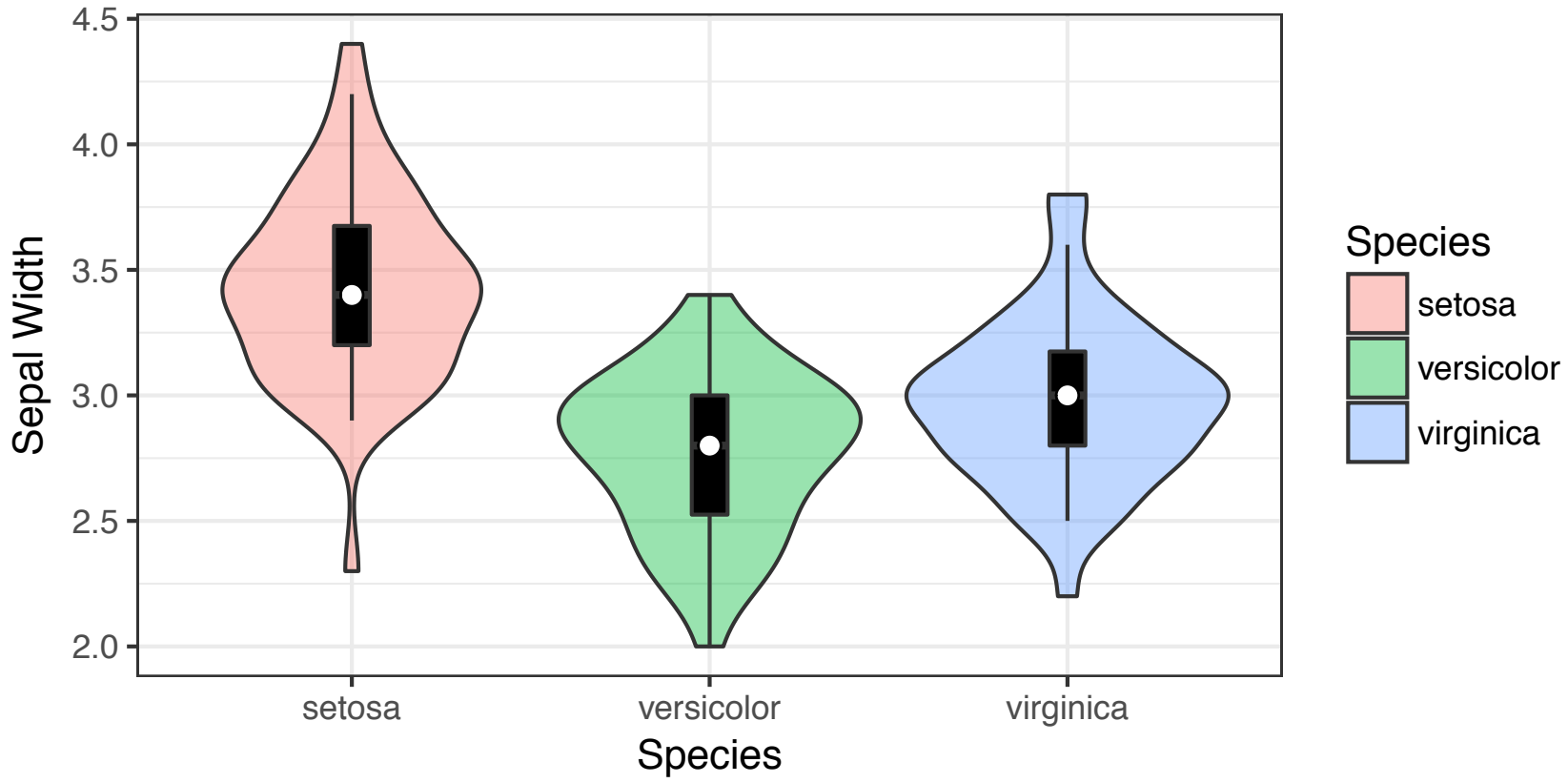
Violin Plot (Fill and Smoothing)

```
ggplot(data=iris,aes(x=Species,y=Sepal.Width))+
  geom_violin(aes(fill=Species), alpha=0.4,adjust=0.5)+
  theme_bw()+ylab("Sepal Width")
```



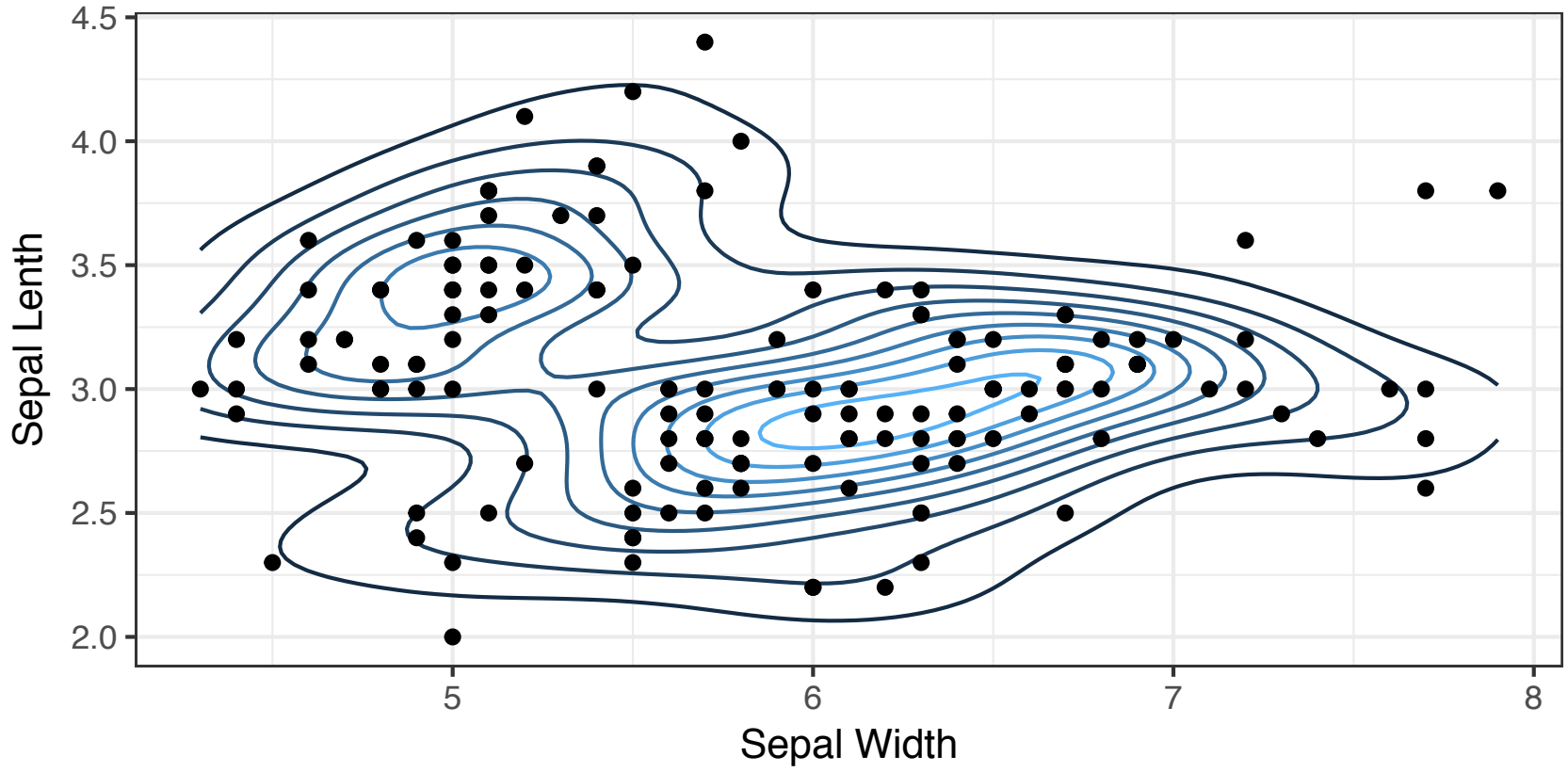
Box + Violin Plot

```
ggplot(data=iris,aes(x=Species,y=Sepal.Width))+
  geom_violin(aes(fill=Species), alpha=0.4)+
  geom_boxplot(width=.1, fill="black", outlier.colour=NA) +
  stat_summary(fun.y=median, geom="point", fill="white",
  shape=21, size=2.5)+theme_bw()+ylab("Sepal Width")
```



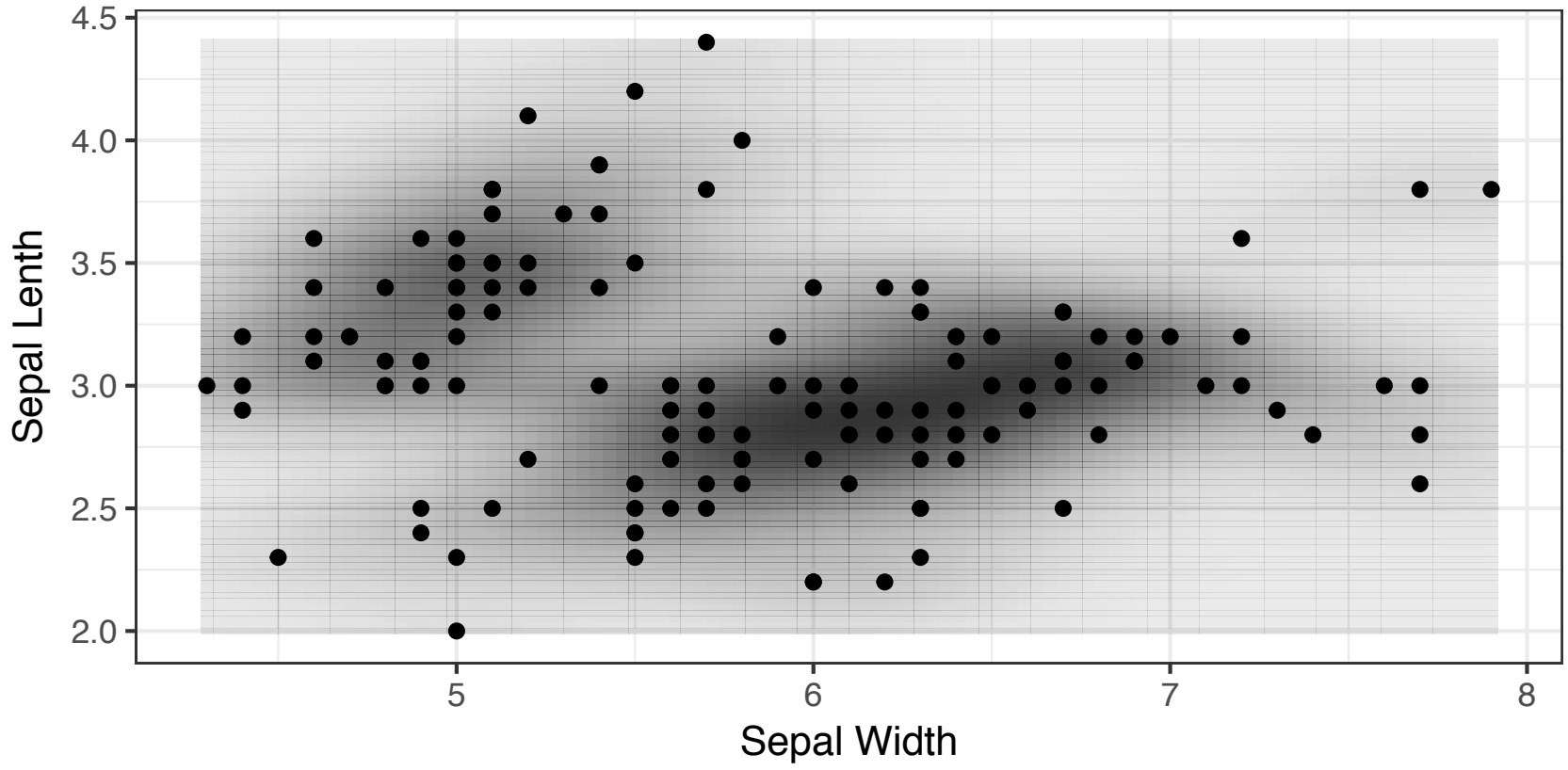
Two-Dimensional Density

```
ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))+
  stat_density2d(aes(color=..level..))+geom_point()+
  theme_bw()+ylab("Sepal Lenth")+xlab("Sepal Width")+
  guides(color=FALSE)
```



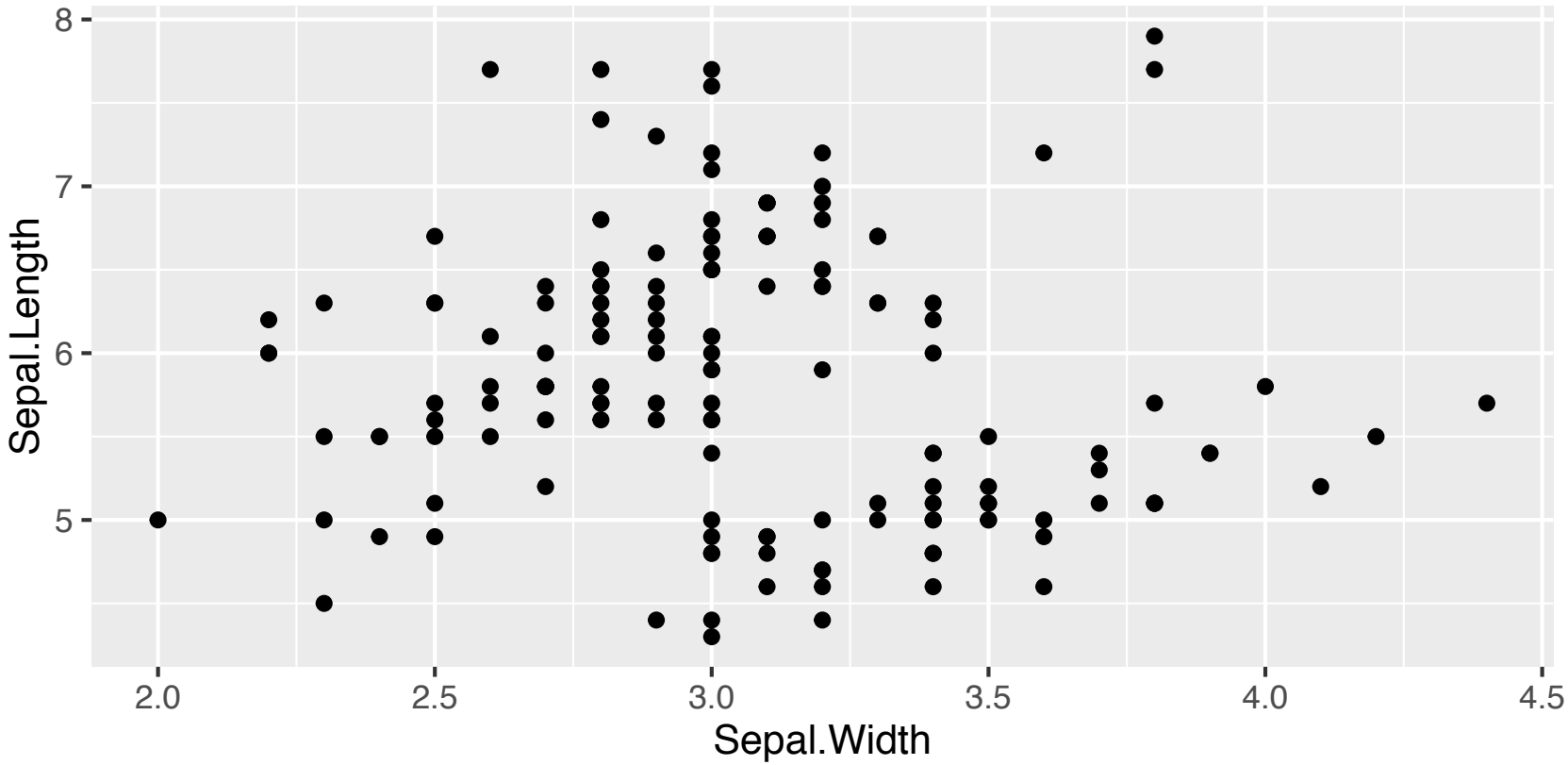
Two-Dimensional Density

```
ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))+
  stat_density2d(aes(alpha=..density..), geom="tile", contour=FALSE)+
  geom_point()+  guides(alpha=FALSE)+
  theme_bw()+ylab("Sepal Lenth")+xlab("Sepal Width")
```



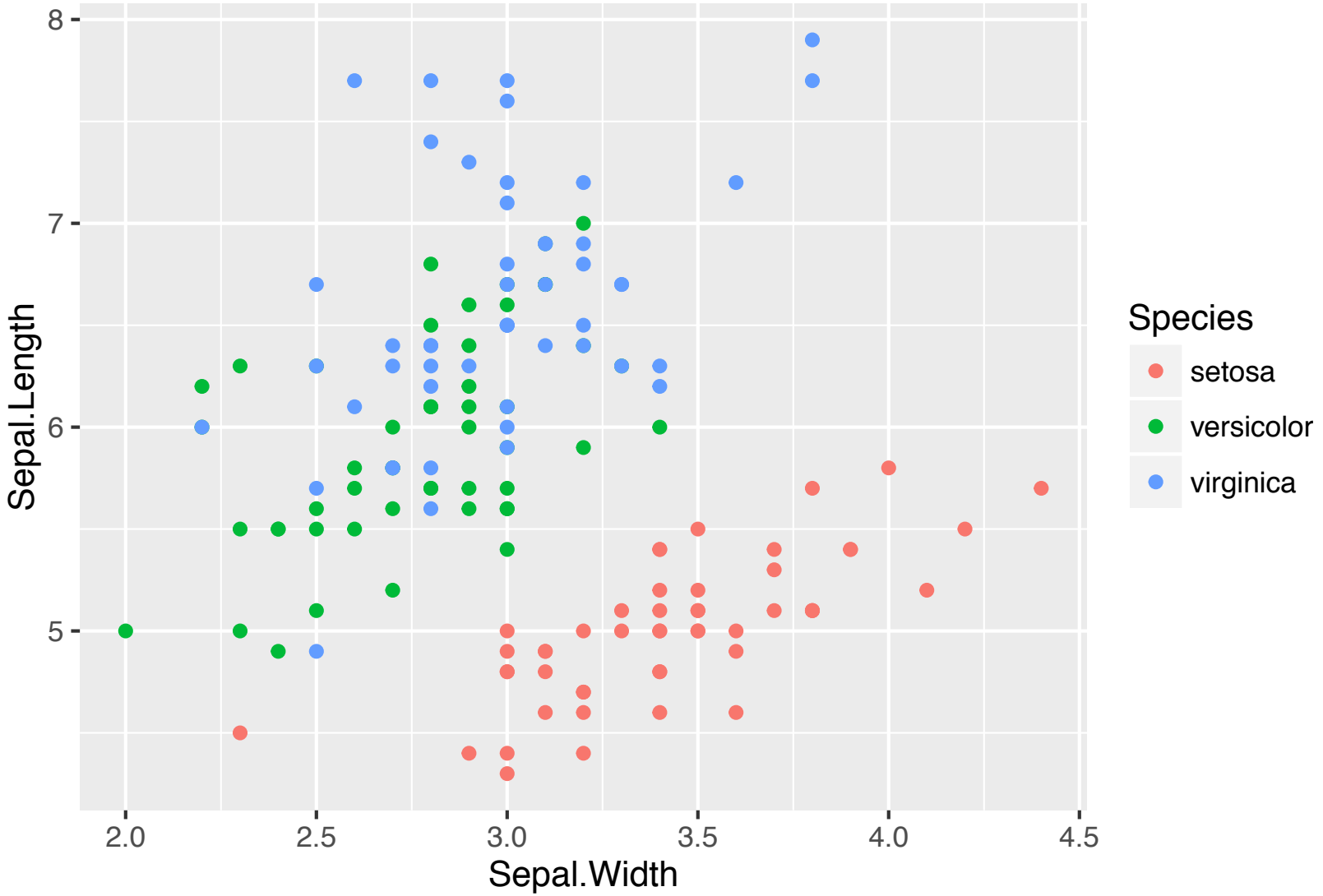
Basic Scatterplot - 2 variables

```
myplot <- ggplot(data=iris)
myplot + geom_point(aes(x=Sepal.Width, y=Sepal.Length))
```



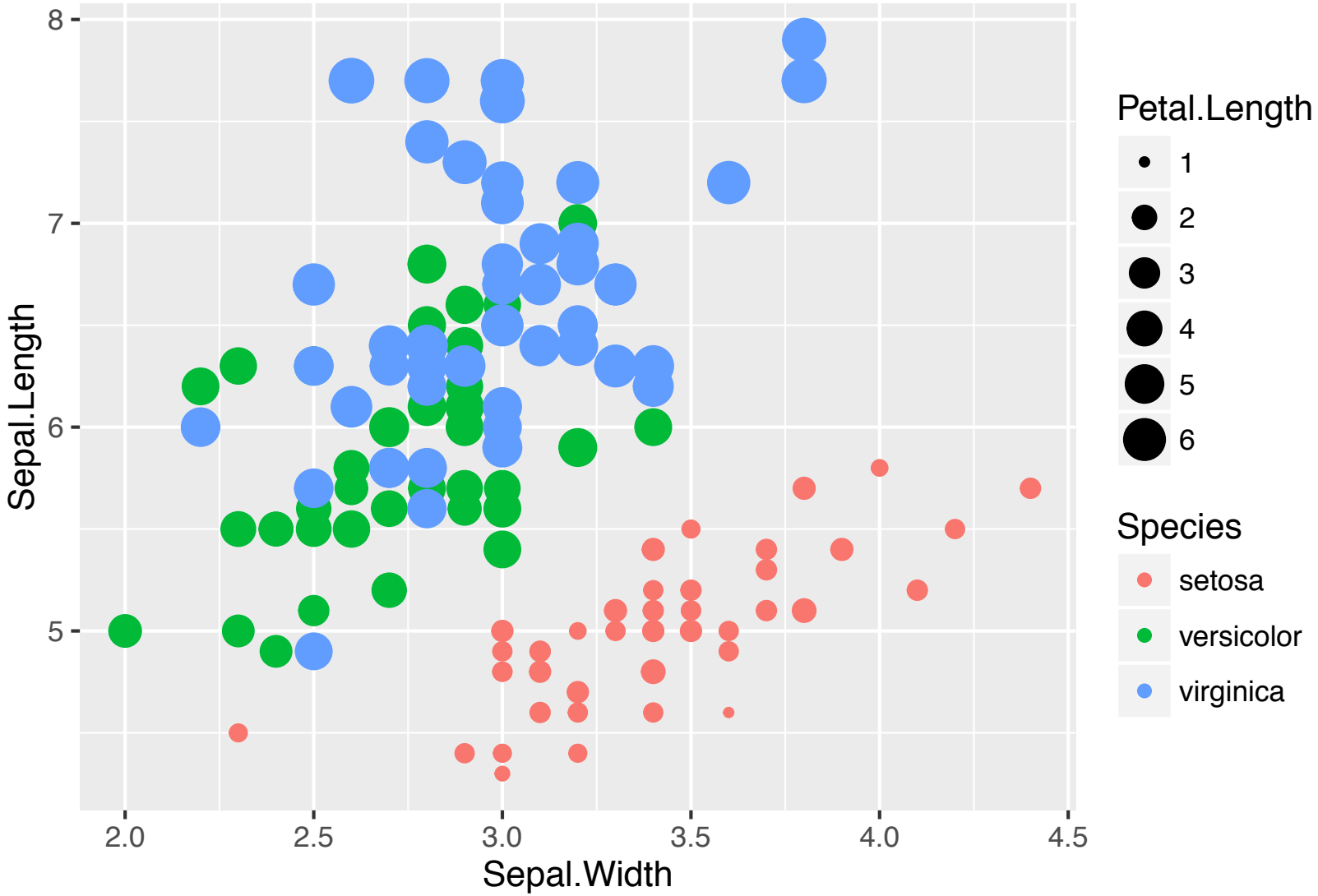
Enhanced Scatterplot - 3 variables

```
myplot <- ggplot(data=iris)
myplot +
geom_point(aes(x=Sepal.Width, y=Sepal.Length, colour=Species))
```



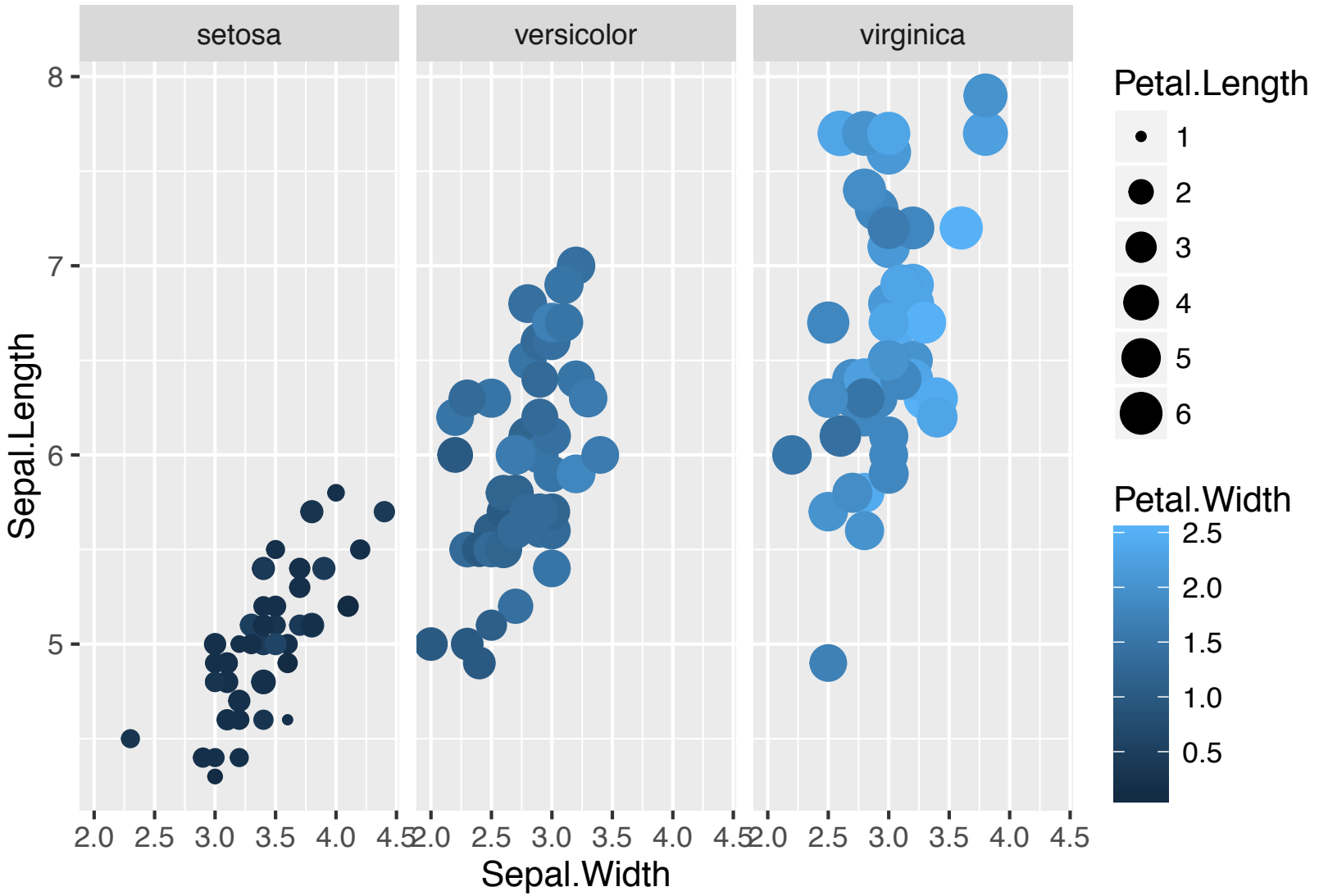
Enhanced Scatterplot - 4 variables

```
myplot <- ggplot(data=iris)
myplot +
geom_point(aes(x=Sepal.Width,y=Sepal.Length,colour=Species,size=Petal.Length))
```



Enhanced Scatterplot - 5 variables

```
myplot <- ggplot(data=iris, aes(x=Sepal.Width, y=Sepal.Length))
myplot <- myplot + geom_point(aes(colour=Petal.Width, size=Petal.Length))
myplot + facet_wrap(~Species)
```



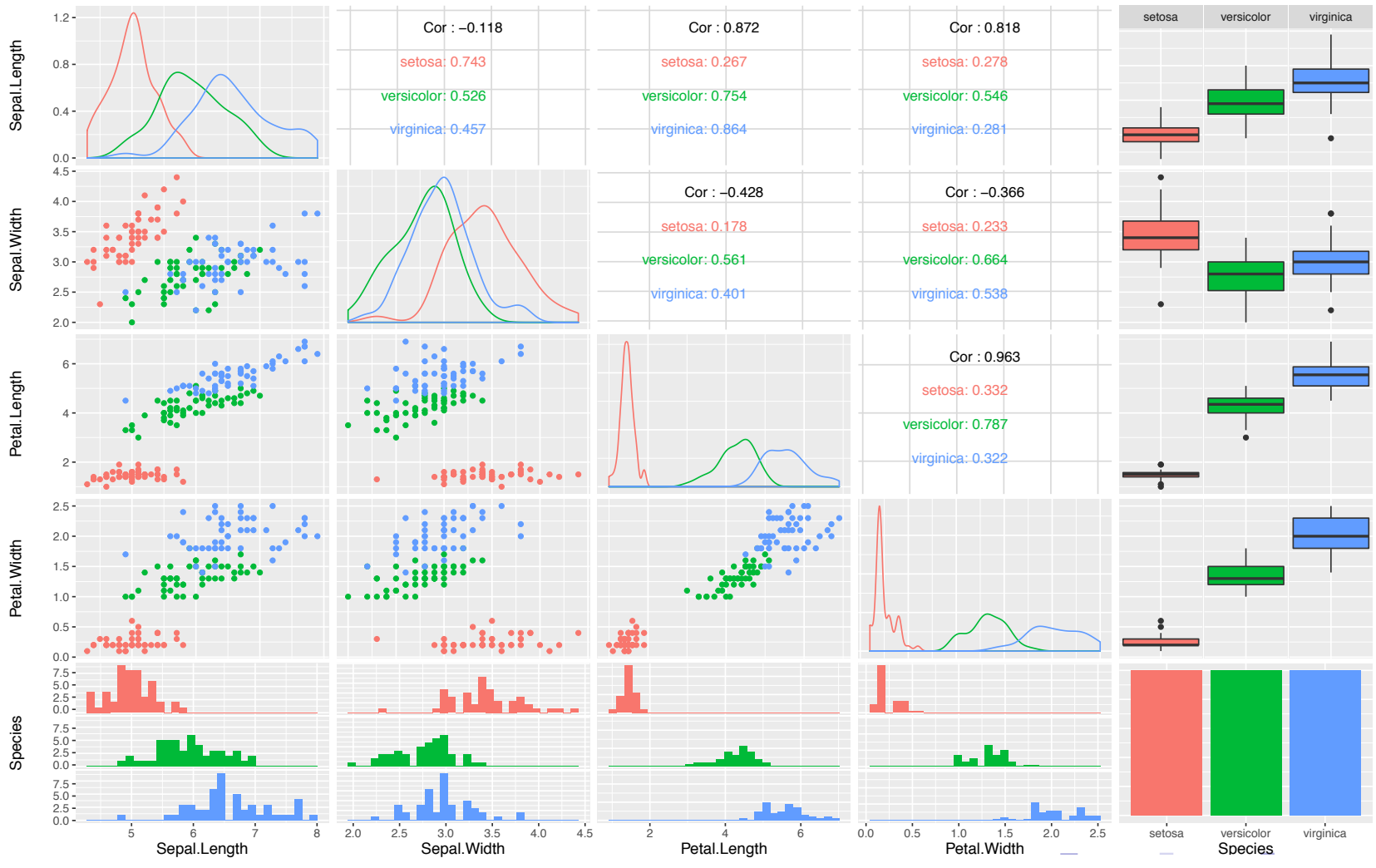
SPLOM - Scatterplot Matrix

- A scatter plot displays the correlation between a pair of variables. Given a set of n variables, there are n-choose-2 pairs of variables, and thus the same numbers of scatter plots.
- These scatter plots can be organized into a matrix(SPLOM), making it easy to look at all pairwise correlations in one place.
- SPLOM is able to assess the relationships between multiple variables simultaneously.
- R package "GGally" is used to generate SPLOM easily. It is designed to be a helper to "ggplot2".

```
library(GGally)
?ggpairs
```

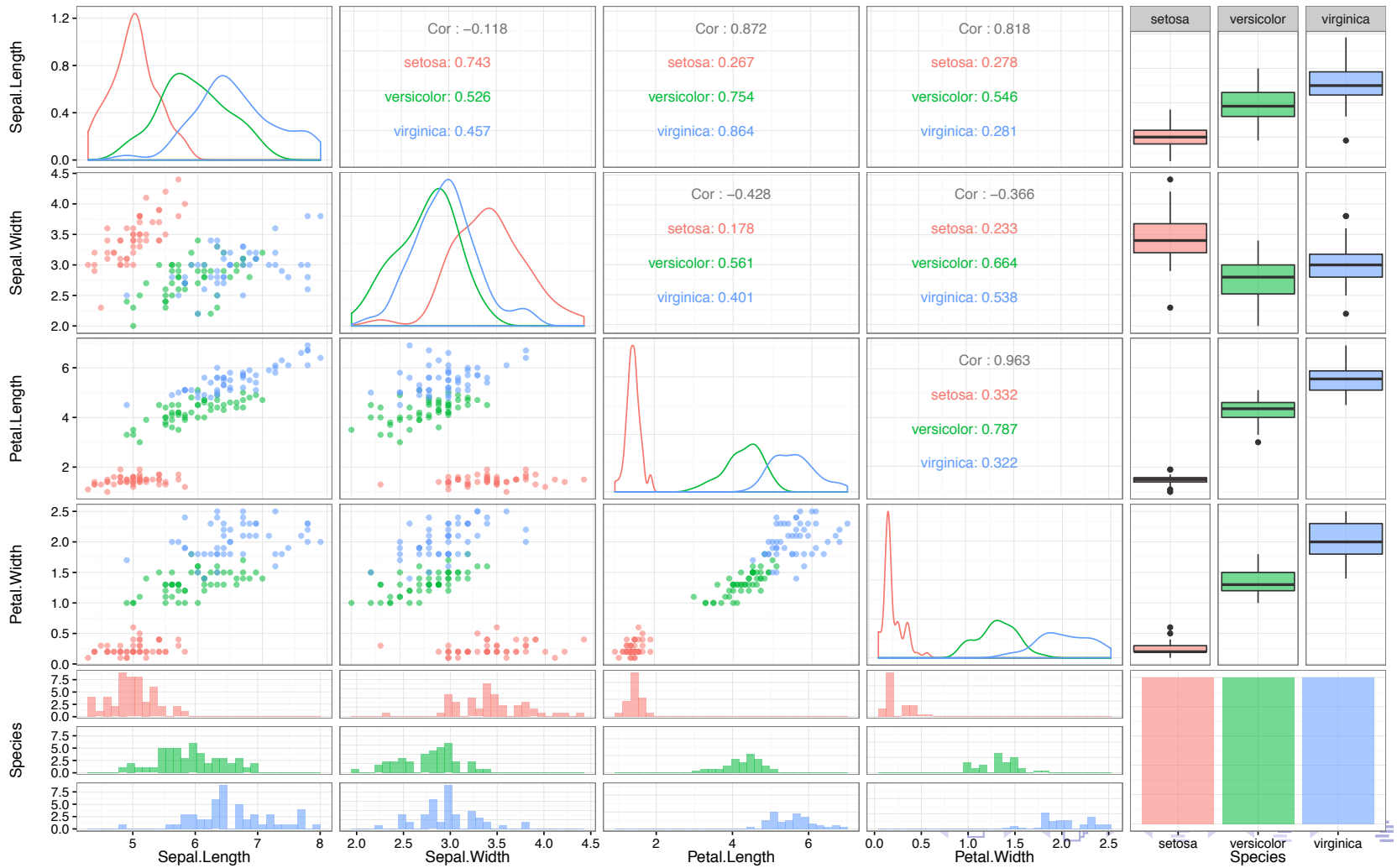
SPLOM - Scatterplot Matrix

```
# SPLOM1. Default settings with ggpairs():
ggpairs(data = iris , mapping = ggplot2::aes(colour = Species))
```



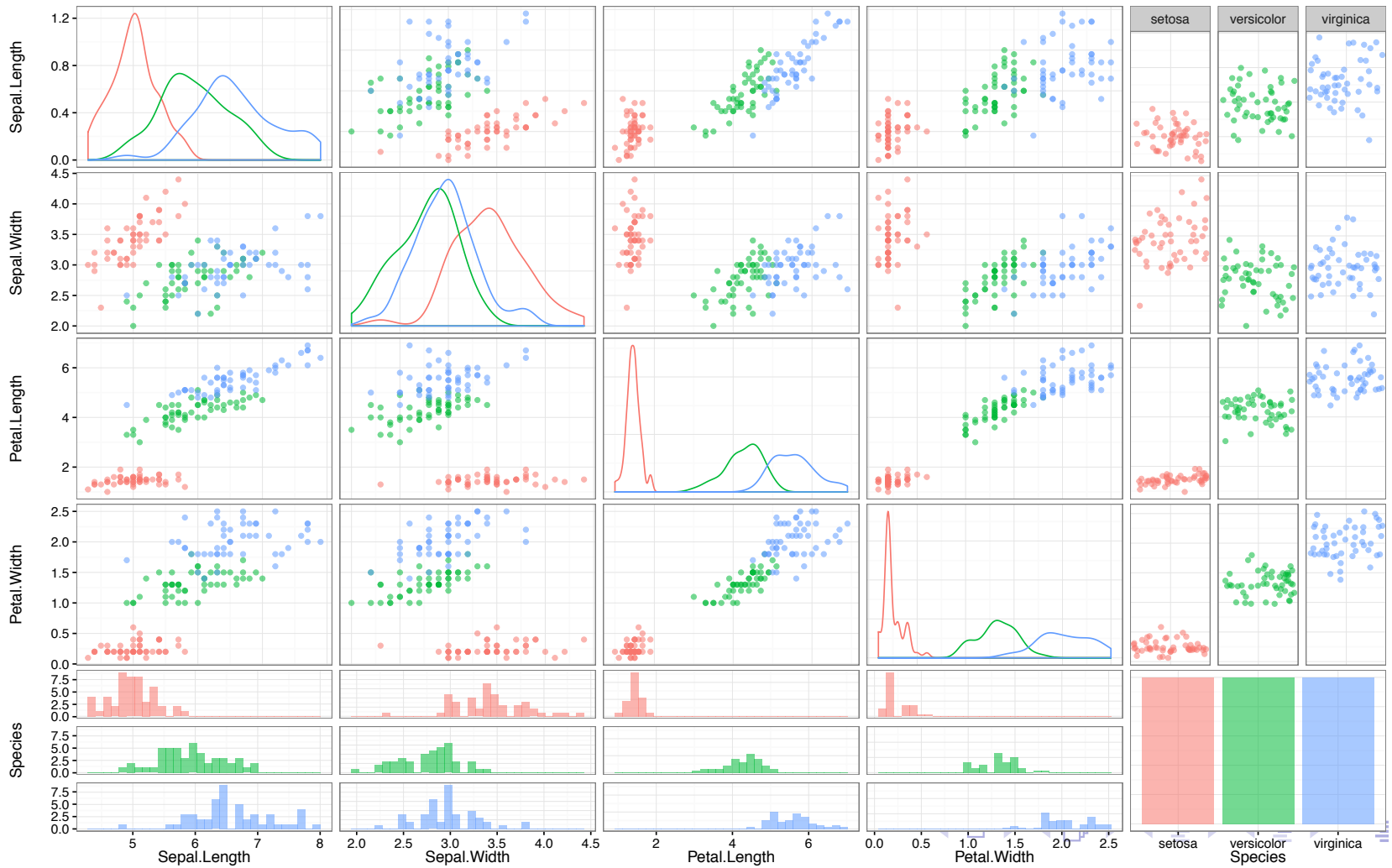
SPLOM - Scatterplot Matrix

```
# SPLOM2. Aesthetics settings with transparency and background theme
ggpairs(data = iris ,
        mapping = ggplot2::aes(colour = Species, alpha = 0.5)) +
        theme_bw()
```



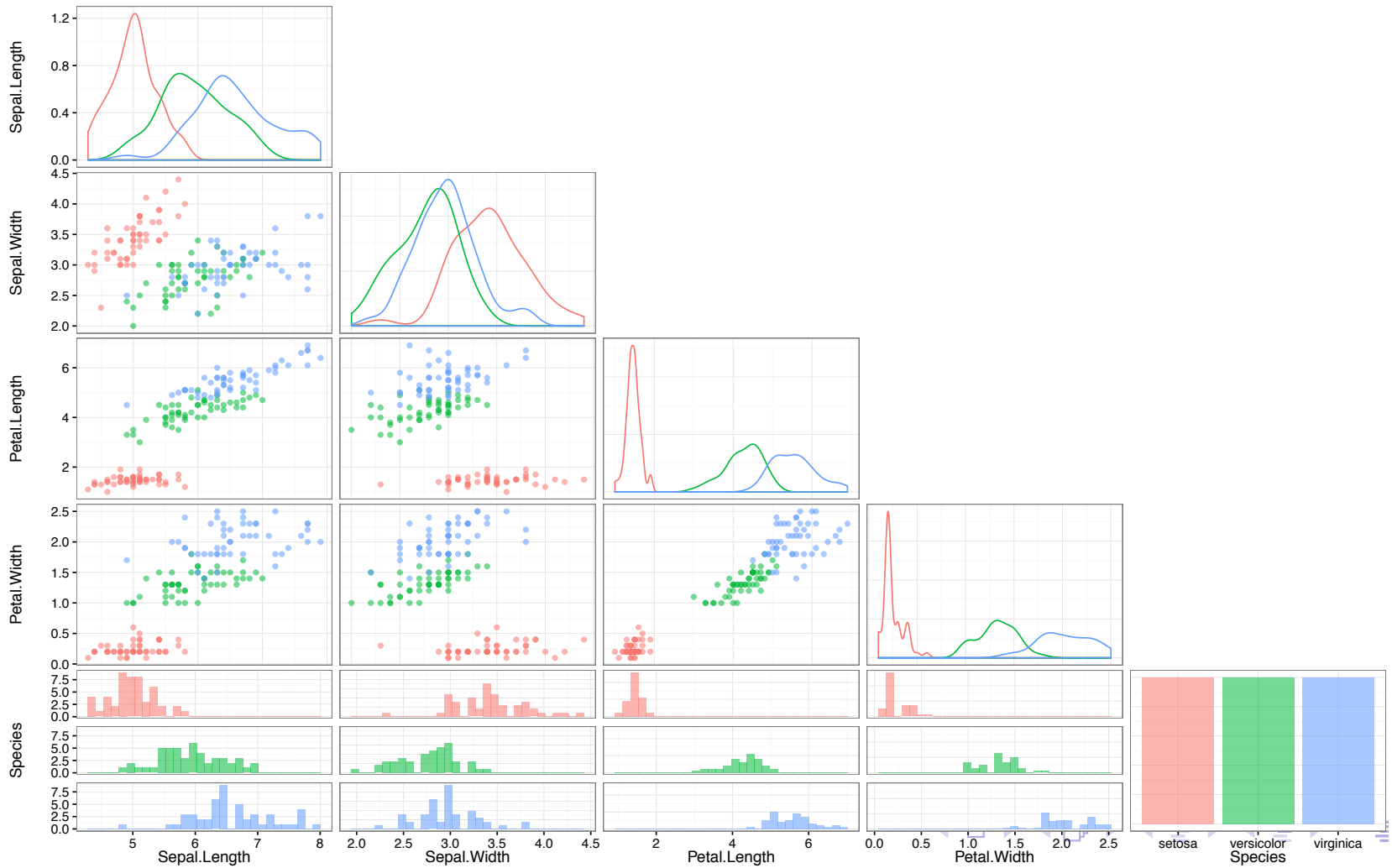
SPLOM - Scatterplot Matrix

```
#SPLOM3. Matrix settings:
ggpairs(data = iris , upper = list(continuous = "points", combo = "dot"),
        mapping = ggplot2::aes(colour = Species, alpha = 0.5)) +
        theme_bw()
```



SPLOM - Scatterplot Matrix

```
#SPLOM4. Matrix settings :
ggpairs(data = iris , upper = "blank",
        mapping = ggplot2::aes(colour = Species, alpha = 0.5)) +
        theme_bw()
```

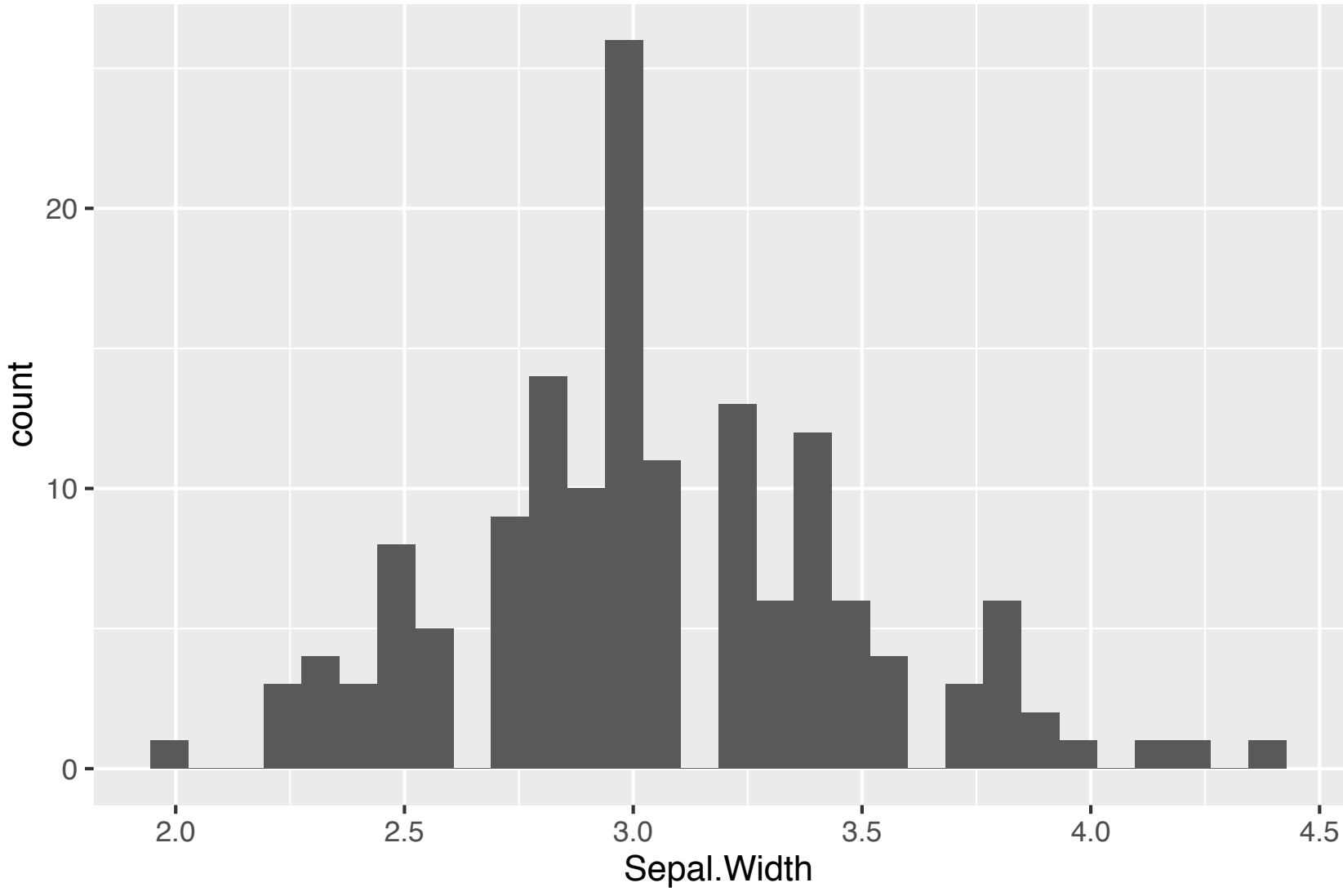


SPLOM - Scatterplot Matrix

Final touchup: you can always use photoshop or paint to quickly fix the desgin for the best visual display:

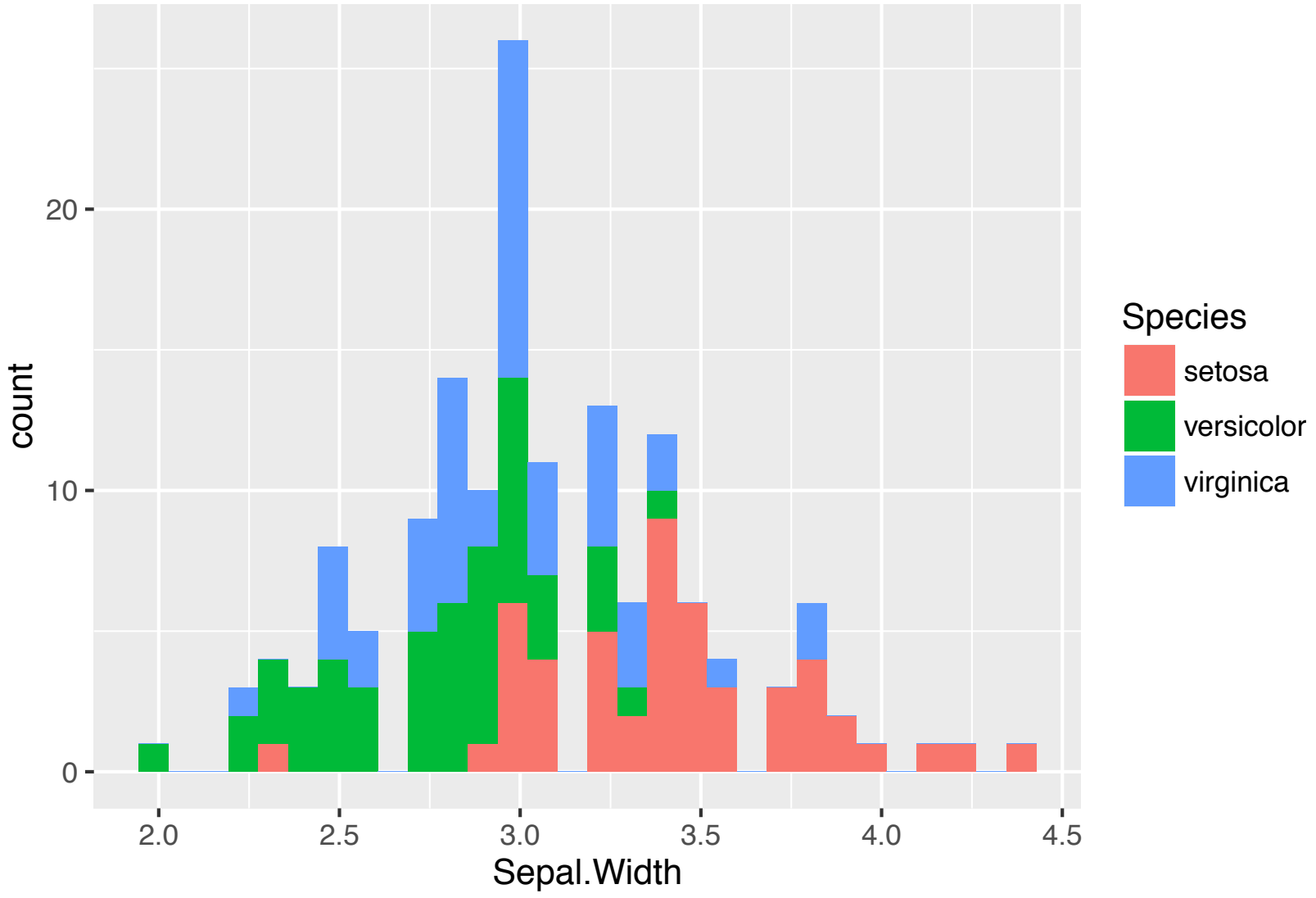
Histogram - Sepal Widths

```
myplot <- ggplot(data=iris)
myplot + geom_histogram(aes(x=Sepal.Width))
```



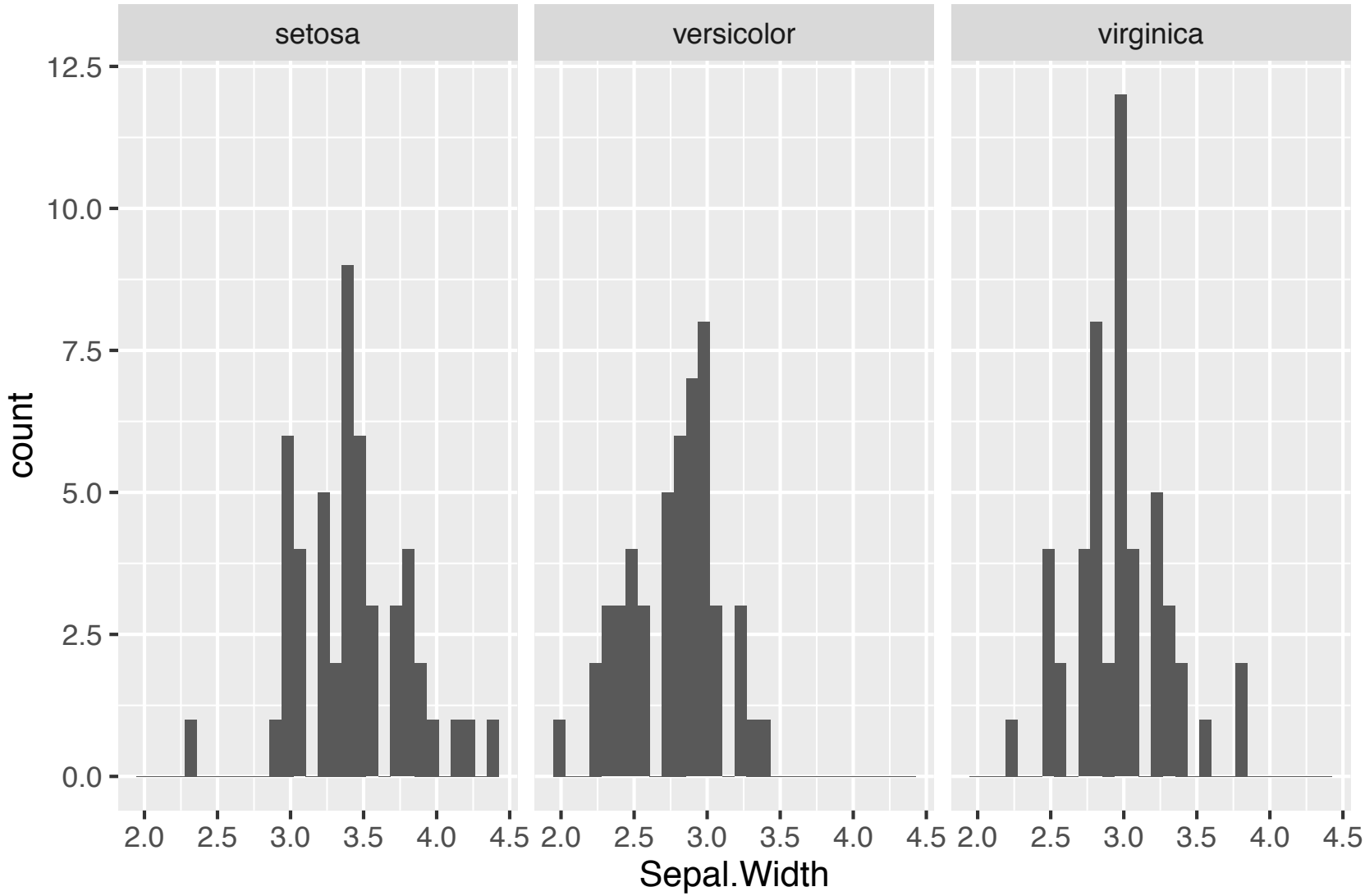
Histogram - Sepal Width and Species (using colors)

```
myplot <- ggplot(data=iris)
myplot + geom_histogram(aes(x=Sepal.Width, fill=Species))
```



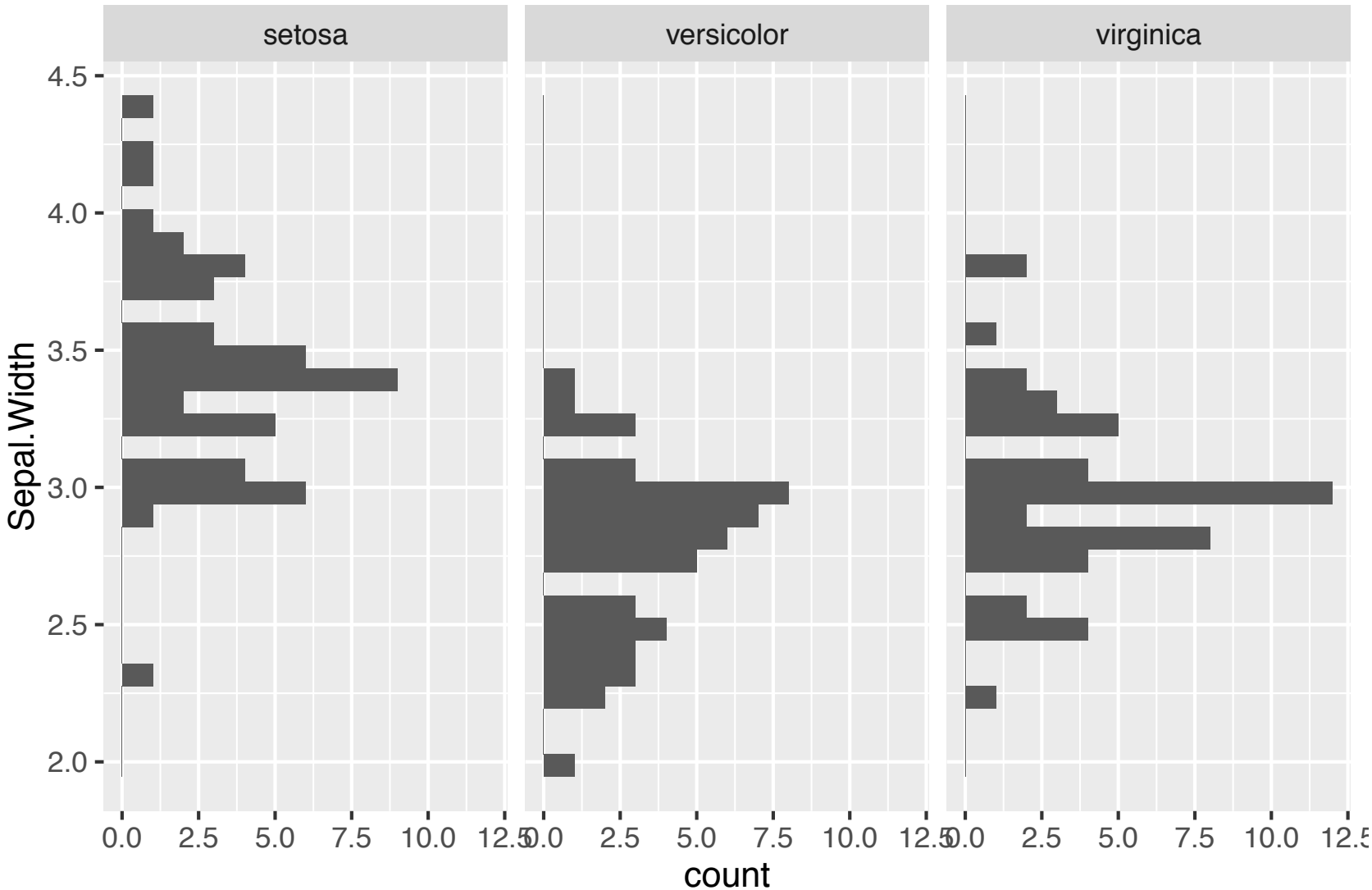
Histogram and Faceting

```
myplot <- ggplot(data=iris)
myplot + geom_histogram( aes(x=Sepal.Width)) +
facet_wrap(~Species)
```



Histogram and Faceting (flipping coordinates)

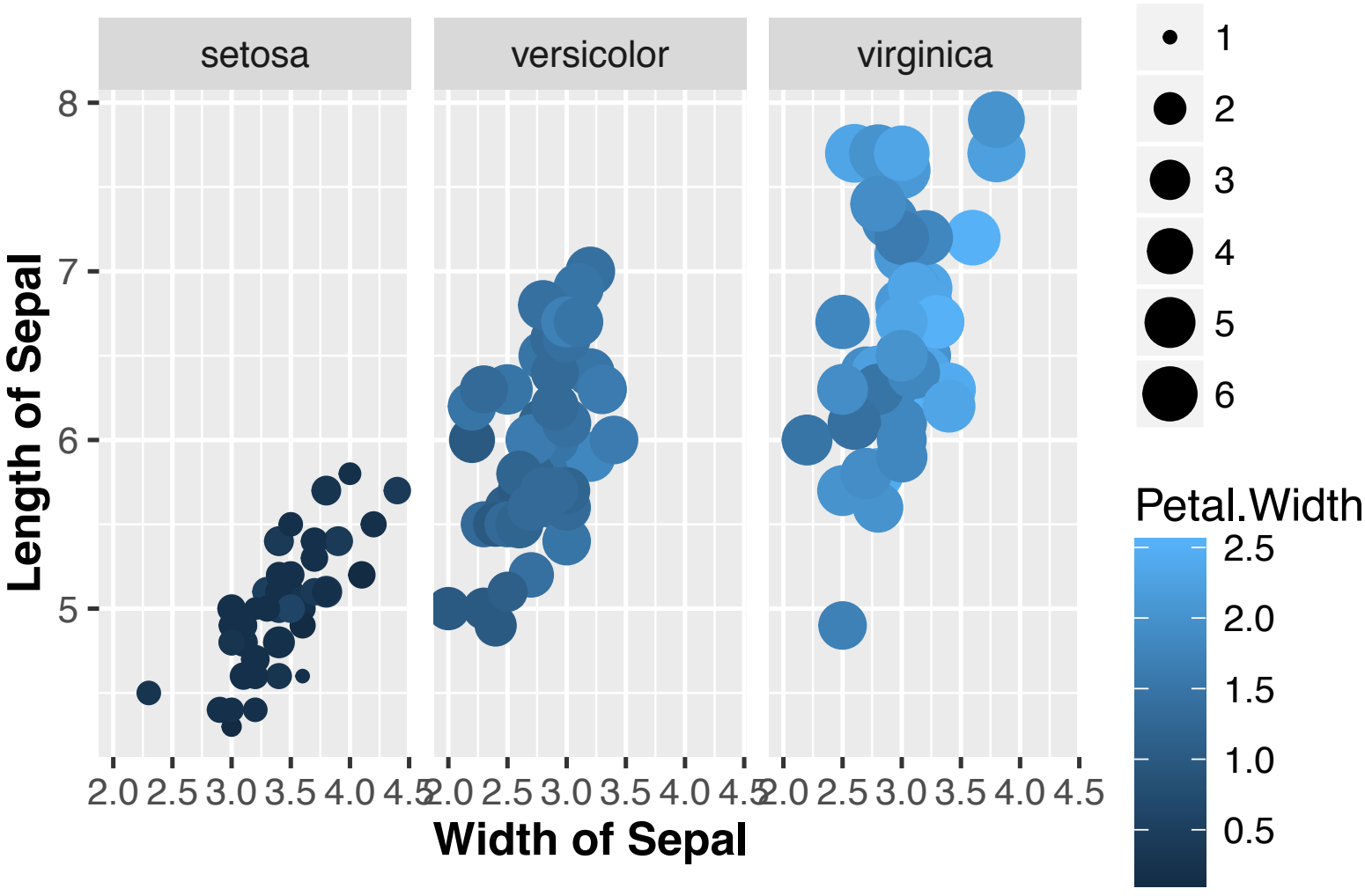
```
myplot <- ggplot(data=iris)
myplot + geom_histogram( aes(x=Sepal.Width)) +
facet_wrap(~Species) + coord_flip()
```



Customizing the graphs

```
myplot <- ggplot(data=iris, aes(x=Sepal.Width, y=Sepal.Length))
myplot <- myplot + geom_point(aes(colour=Petal.Width, size=Petal.Length))
myplot <- myplot + facet_wrap(~Species)
myplot <- myplot + xlab('Width of Sepal') + ylab('Length of Sepal')
myplot <- myplot + theme(axis.title.y = element_text( face='bold'))
myplot <- myplot + theme(axis.title.x = element_text( face='bold'))
myplot
```

Customizing the graphs



Mini Challenge # 1

1. Install (if you have not done so) the ggplot2 library (type *install.packages('ggplot2')*).
2. Load the library ggplot2 (type *library(ggplot2)*).
3. Download the swain1971 dataset from the course website.
4. Imagine that each point in the dataset represents the amount of people (weight) living at that location (x and y coordinates). If you were to open a distribution center to serve the entire population (e.g. a supermarket) and you want people to move as little as possible from their location to the distribution center, where would you open the distribution center? Assume that the distribution center needs to be placed in the same location as one of the demand points.
5. Provide the id of the location where you would open the distribution center and any graphs generated to reach your conclusion.

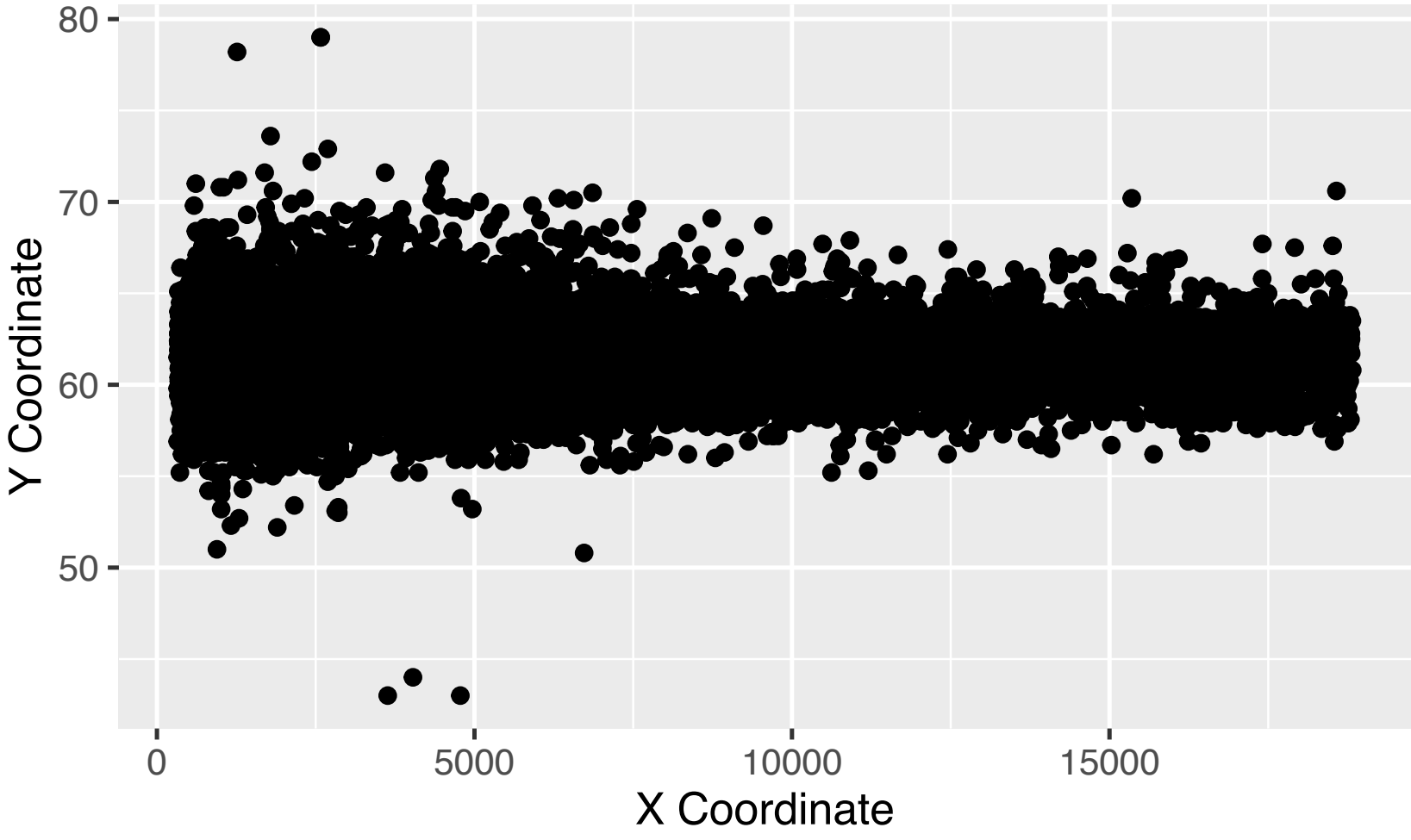
Case Study: *diamonds dataset*

Explore the dataset:

```
str(diamonds)
head(diamonds)
```

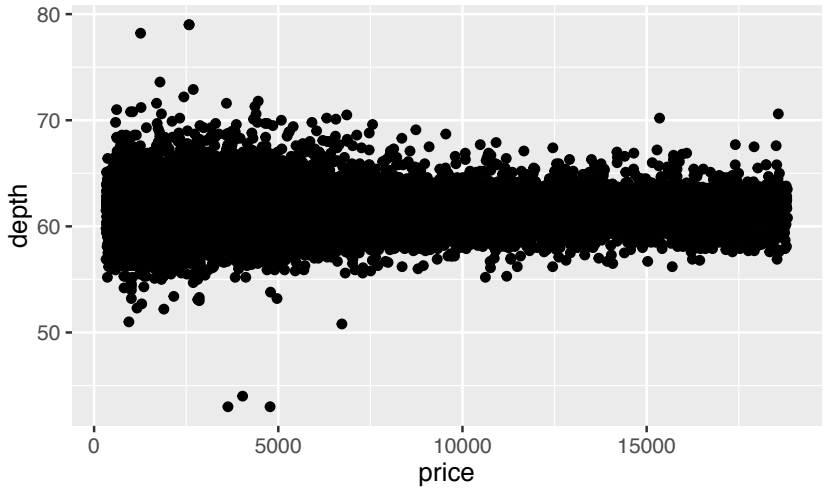
Let us display the price vs depth

```
myplot <- ggplot(data = diamonds) +
  geom_point(aes(x = price, y = depth))
myplot <- myplot + xlab('X Coordinate') + ylab('Y Coordinate')
myplot
```

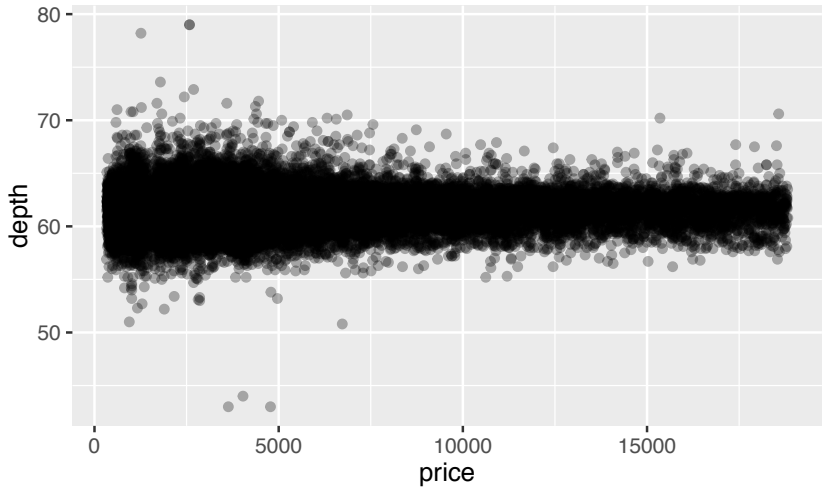


Detecting where points overlap: alpha blending

```
myplot <- ggplot(data = diamonds)
myplot + geom_point(
  aes(x = price,y = depth))
```

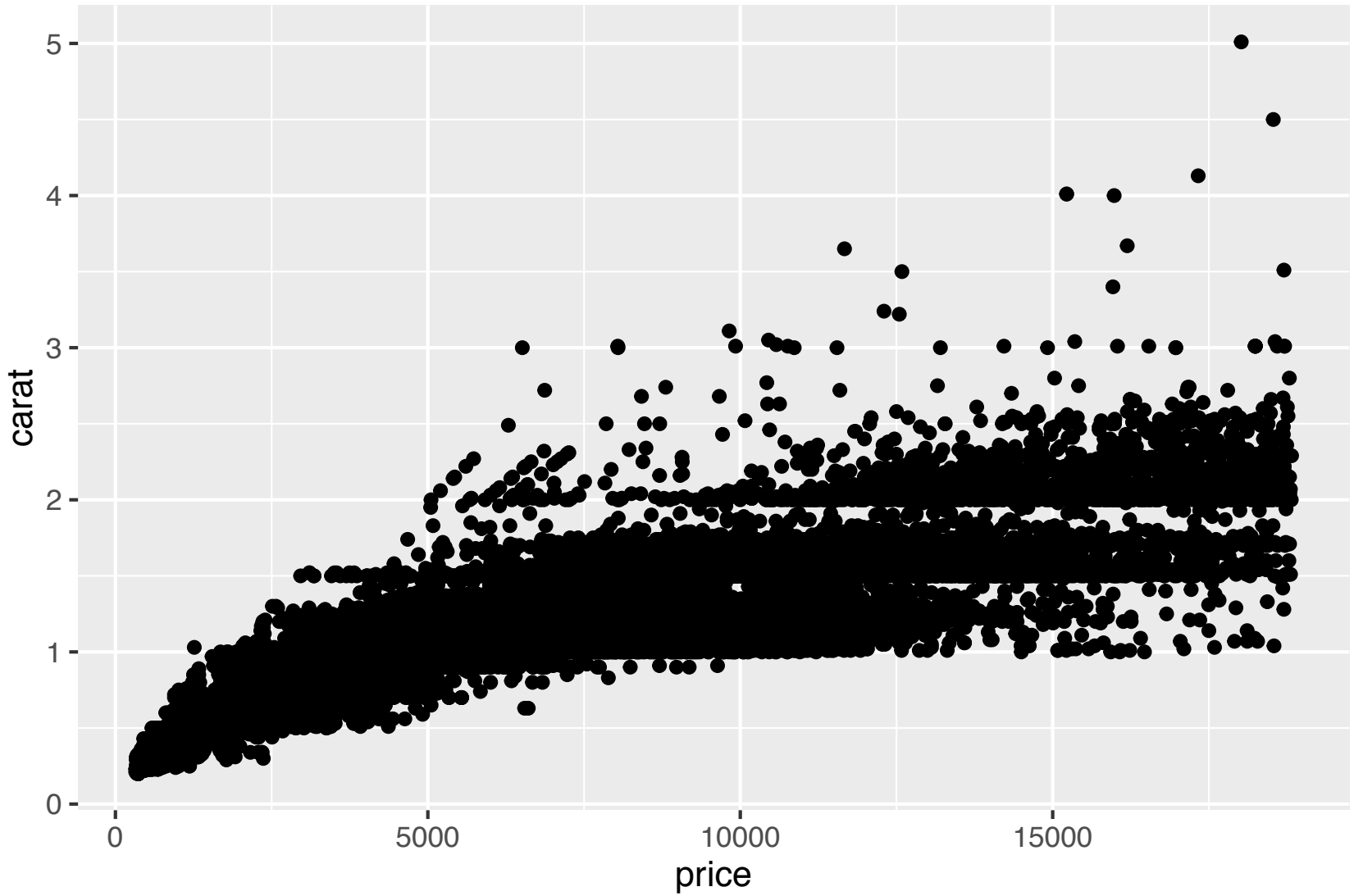



```
myplot <- ggplot(data = diamonds)
myplot + geom_point(
  aes(x = price,y = depth), alpha = 0.3)
```



Let us display the price vs carat

```
ggplot(data = diamonds) + geom_point(aes(x = price, y = carat))
```

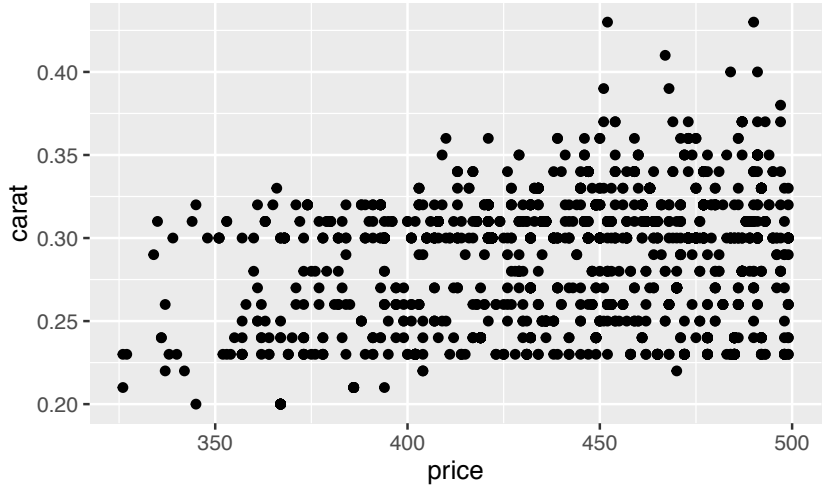


How many points do we have in the really dark areas?

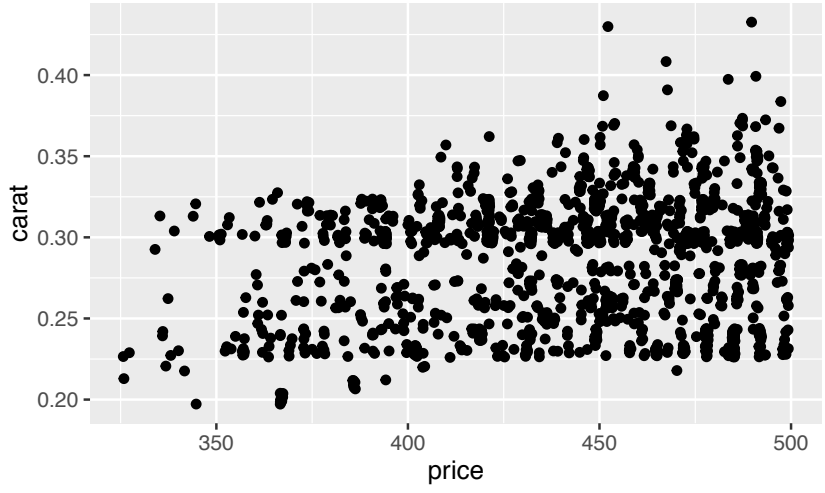
Moving points that overlap: jittering

```
#let us work with a subset of the data
mysubset <- subset(diamonds, price < 500)
```

```
myplot <- ggplot(data = mysubset)
myplot + geom_point(
  aes(x = price,y = carat))
```

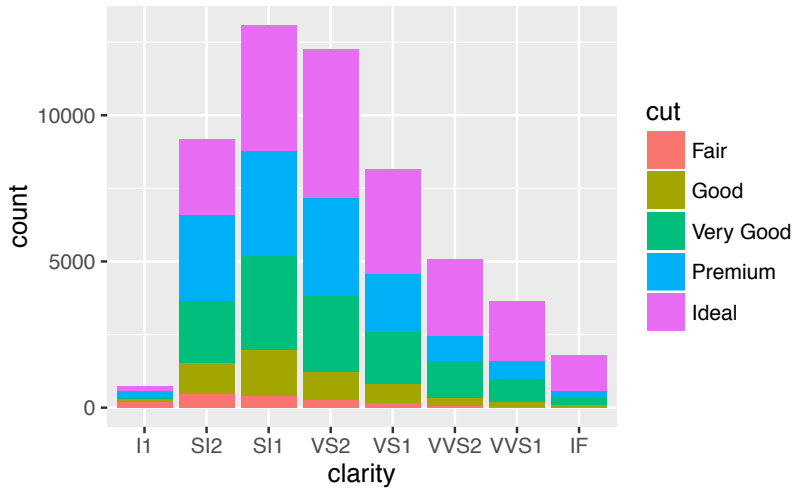



```
myplot <- ggplot(data = mysubset)
myplot + geom_jitter(
  aes(x = price,y = carat))
```

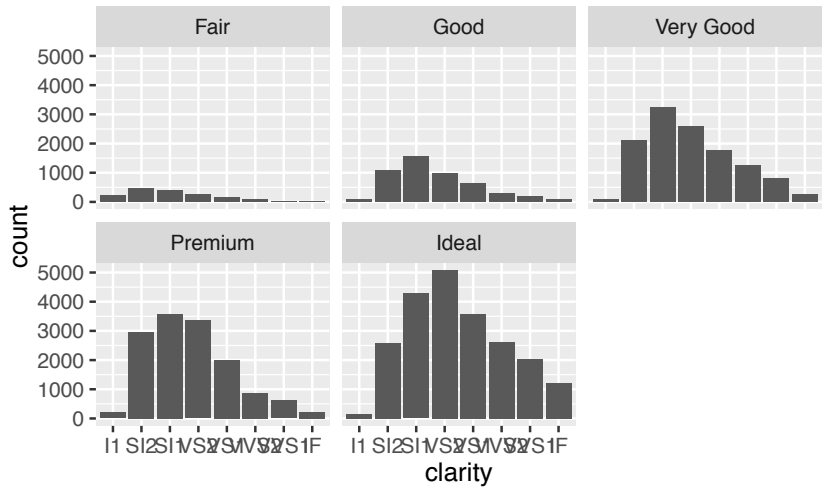


Using bars to display data

```
ggplot(data=diamonds) + geom_bar(
aes(x = clarity, y =..count.., fill = cu
```



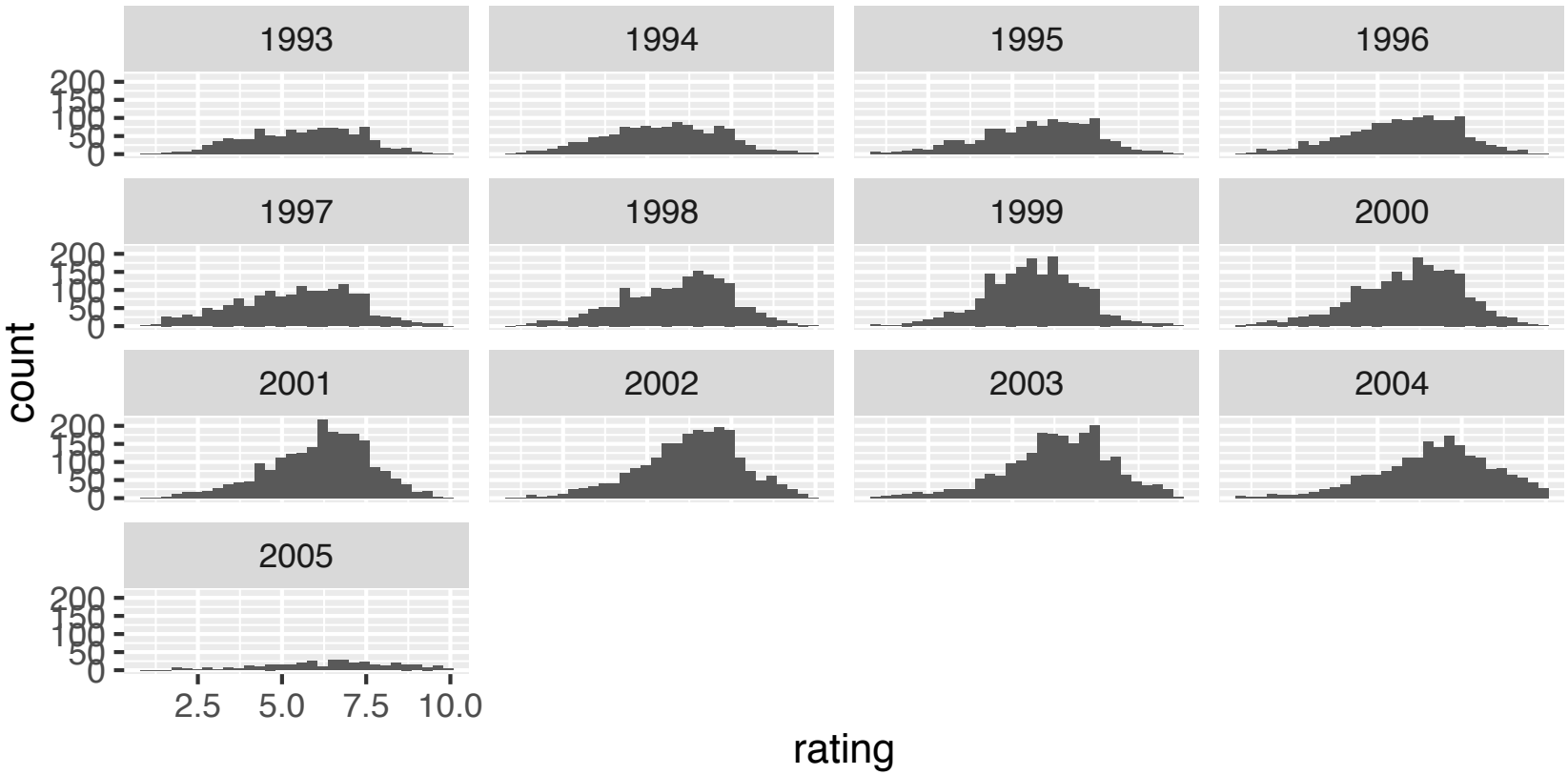

```
ggplot(data = diamonds) +
geom_bar(aes(x = clarity, y =..count..)) +
facet_wrap(~cut)
```



Case study: *movies* dataset

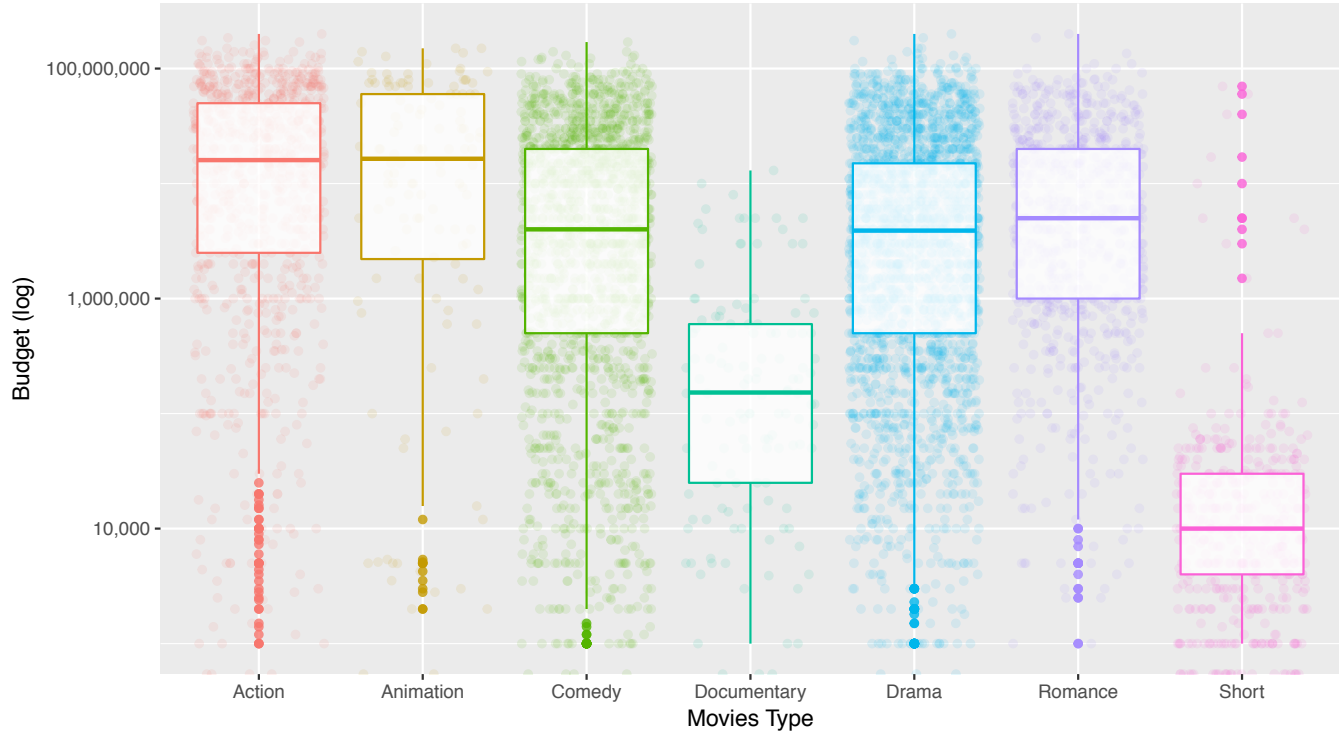
How have movie ratings evolve in time ?

```
#Data set movies in package (ggplot2movies)
 # install.packages("ggplot2movies")
  library("ggplot2movies")
  small <- subset(movies, year > 1992)
ggplot(data = small) + geom_histogram(aes(x = rating)) + facet_wrap(~year)
```



Movie Example

```
# clean the data before plotting
ggplot(data=movie_data,
       aes(type,budget,color=type)) +
  geom_jitter(alpha=0.1) +
  geom_boxplot(alpha=0.8)+
  scale_y_log10(labels = scales::comma)+
  xlab('Movies Type')+ylab('Budget (log)')+ guides(color=FALSE)
```



Homework

Hospital readmission occurs when a patient who had been discharged from a hospital is admitted again within a specified time interval. Decreasing readmissions has the potential to simultaneously lower costs and improve quality. Hospital Readmissions Reduction Program requires CMS to reduce payments to IPPS hospitals with excess 30-days readmissions.[1] We are interested in how hospitals' readmissions have beeb changed since the establishment of the program in 2012. Readmission data has been downloaded and aggregated from Year 2013 to 2016 from Hospital Compare datasets[2]. However, the dataset requires further transformation (using gather function from tidyr library) before plotting. Please generate the boxplot, violin plot, density plot, summarize readmission changes, and discuss your perference among boxplot, violin plot and density plot for visualizing data distribution.

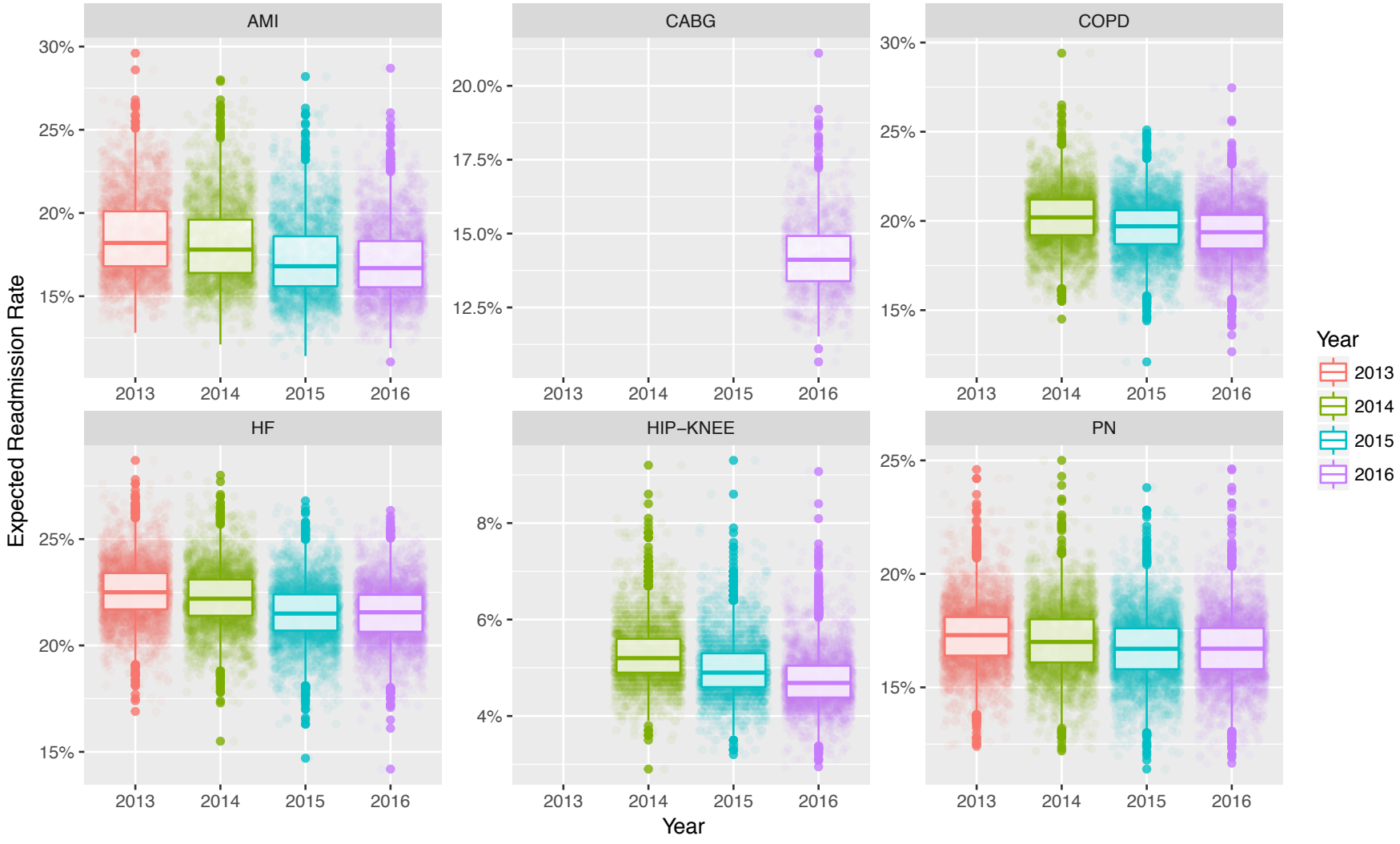
---

[1] https://www.cms.gov/medicare/medicare-fee-for-service-payment/acuteinpatientpps/readmissions-reduction-program.html

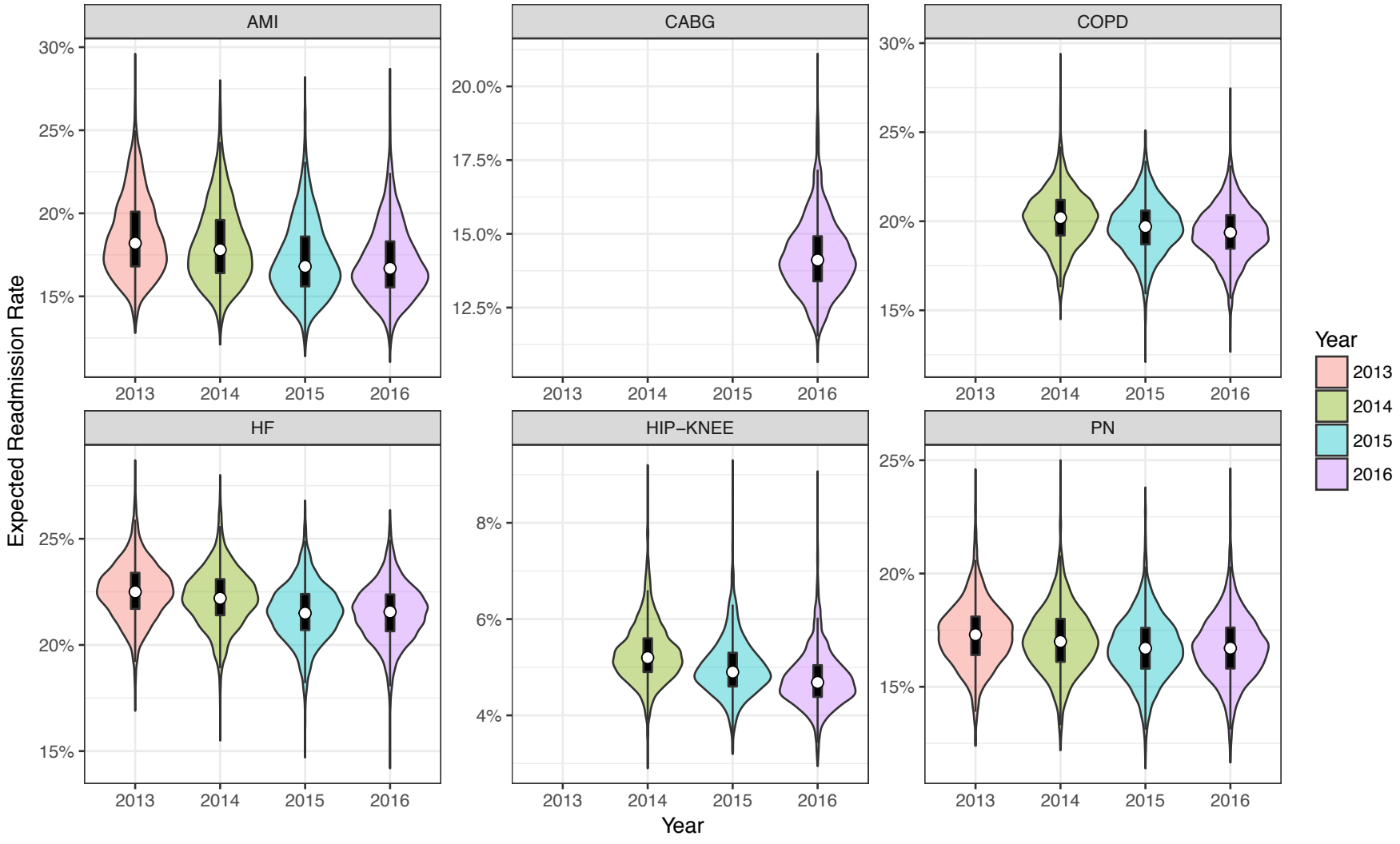
[2] https://data.medicare.gov/Hospital-Compare/Hospital-Readmissions-Reduction-Program/9n3s-kdb3

Homework: box plot

Set x = Year, y = Expected Readmission Rate, color = Year, facet = Measure Name, and create a jitter and box plot. Be aware of the scales and labels in y axis.

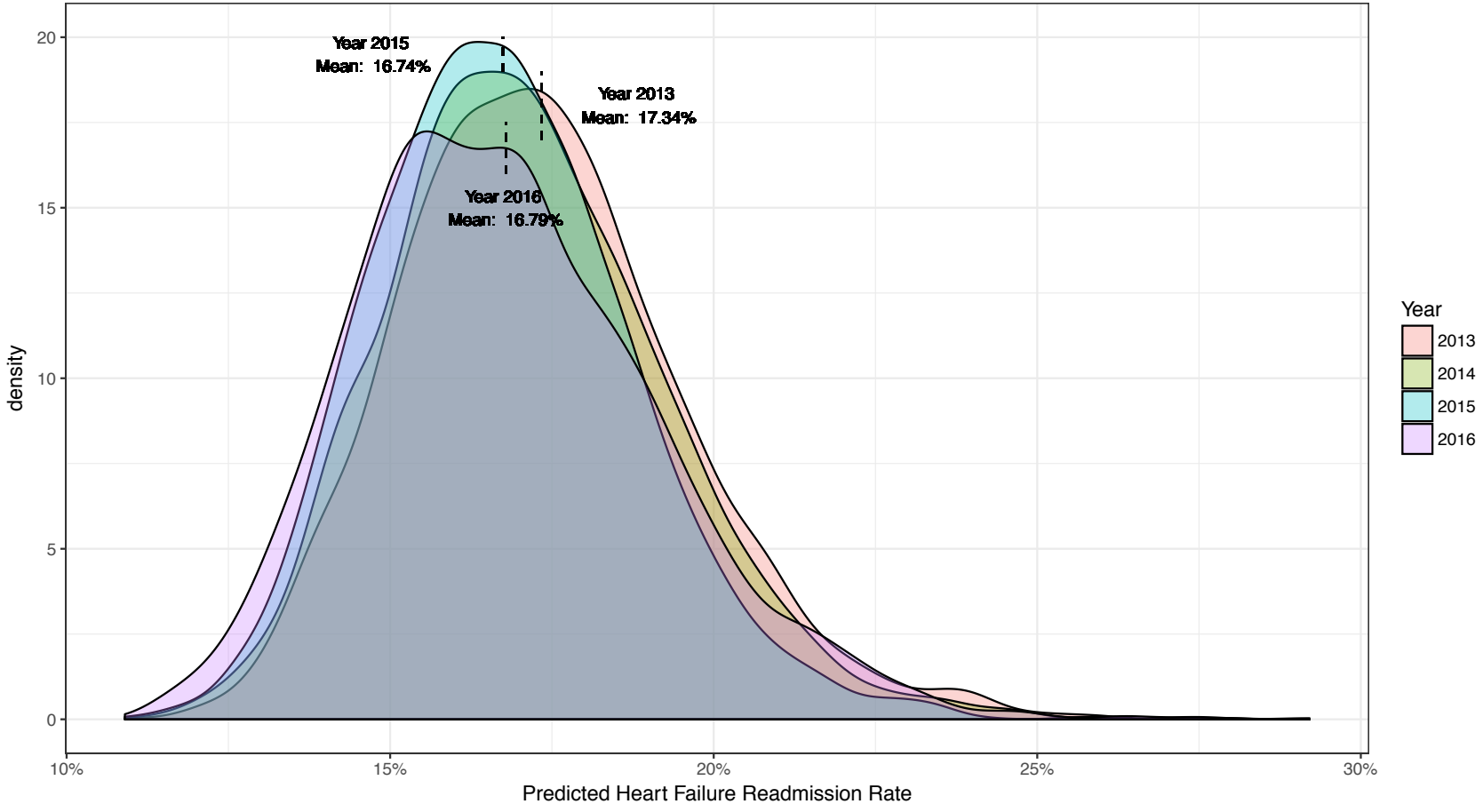


Homework: violin plot



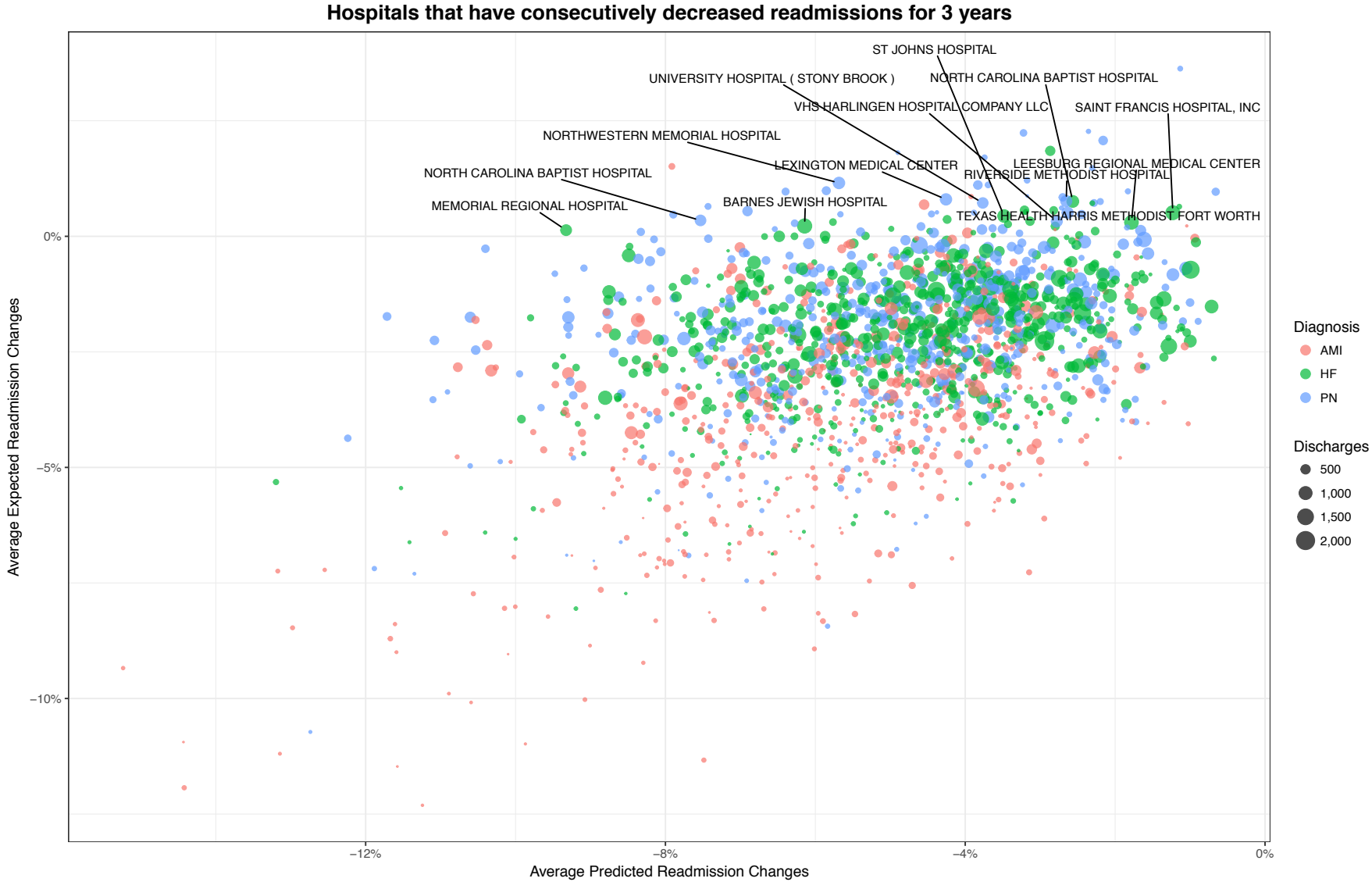
Homework: density plot

Choose one measure and predicted or expected readmission rate for all 4 year, create a density plot and add meaningful annotations (do not need to be exactly the same as the figure below).



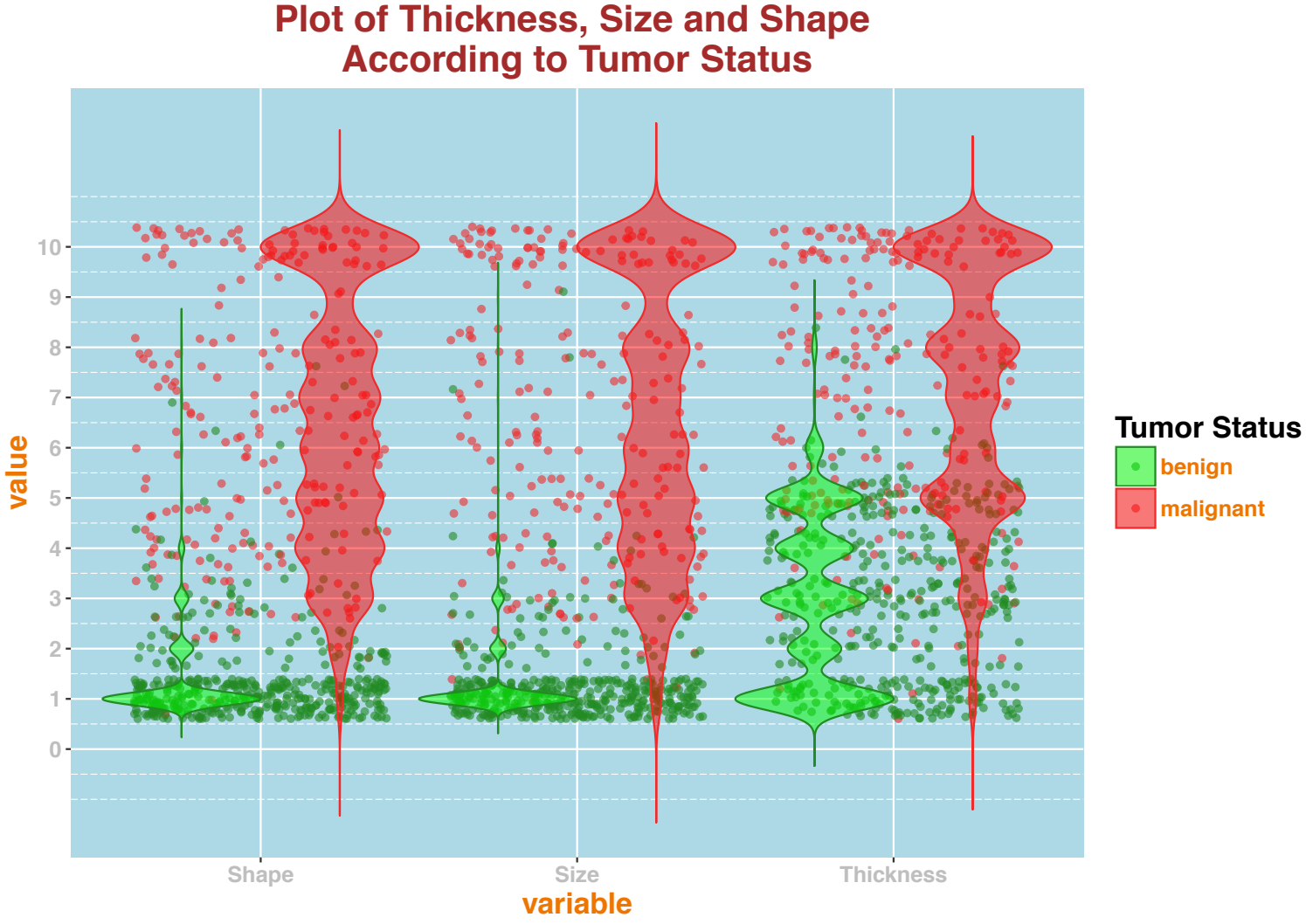
Homework (bonus)

Feel free to explore the dataset and come up with other plots



Homework (bonus)

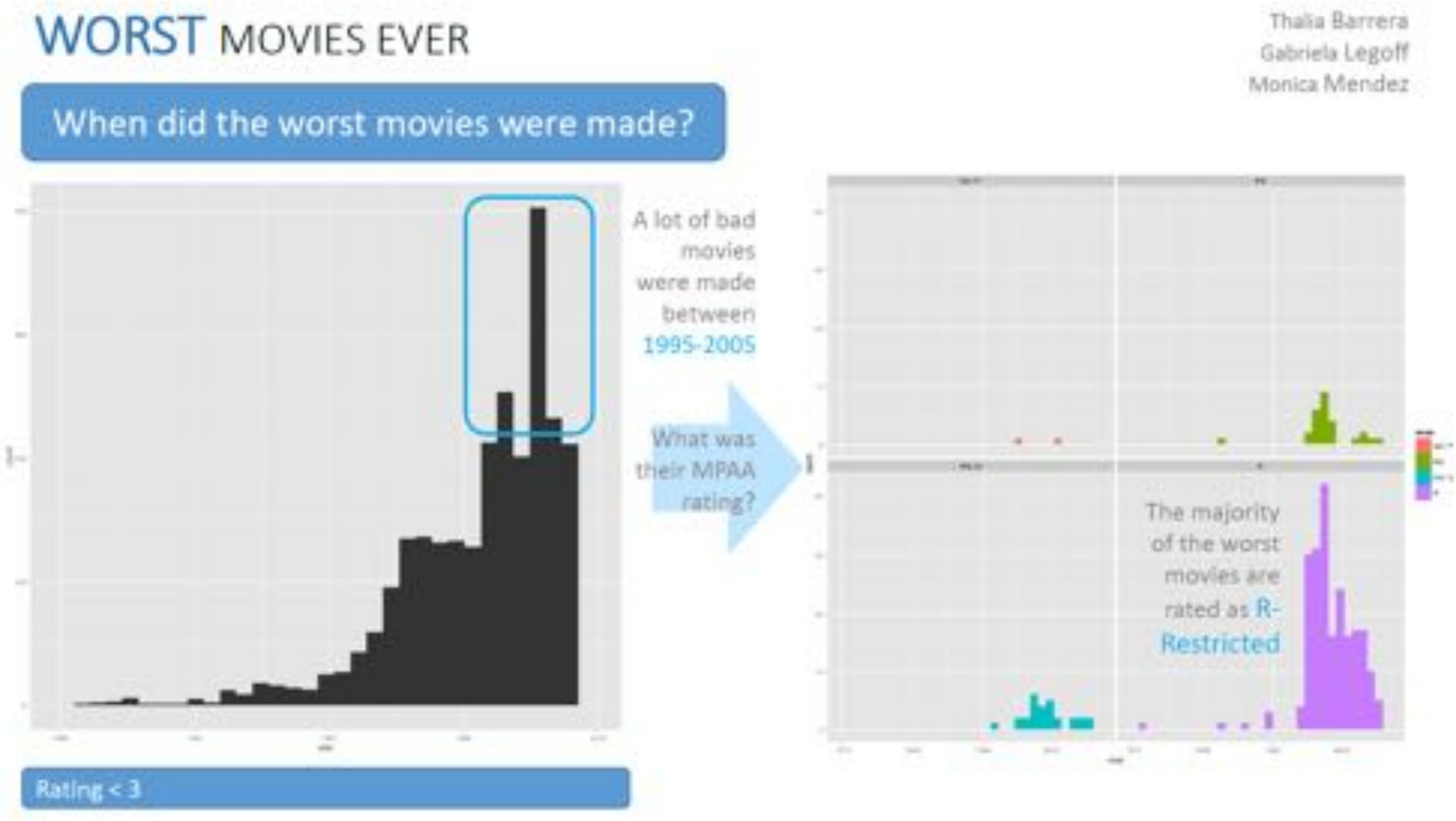
Replicate the following violin plot (cancer_data is ready for plotting).
This violin plot was initially created by Ji Gu from Class 2016 Fall.



Challenge #1 - Design your data story

1. Explore the *movies* dataset.
2. Come up with an interesting question that the dataset can help you answer.
3. Create a visualization that answers the question or tells a story about the dataset.

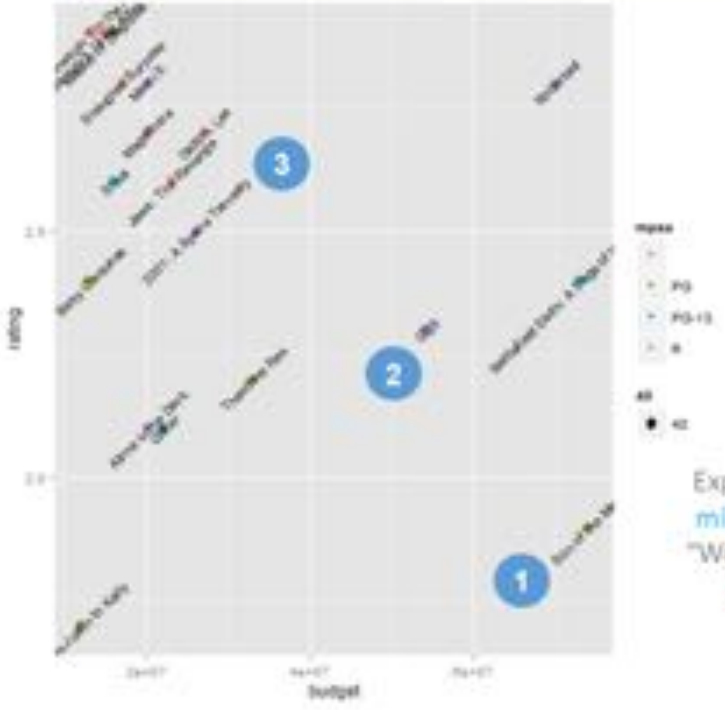
Challenge #1 - Design your data story



Challenge #2 - Design your data story

WORST MOVIES EVER

What are the most expensive?



3

2

1

The Most Expensive (~70 million dollars) "Worst Movie": Son of the Mask

The Godfather had ~5 million dollars budget (9.2/10)

Homework

1. Download the Breast Cancer Dataset from the course website.
2. Take some time to explore the dataset and learn about its context (`http://archive.ics.uci.edu/ml/datasets/Breast+Cancer`).
3. Create a visualization that will help you classify the type of breast cancer.
4. Show the graph that you propose as being the most helpful one for classifying breast cancer.

Resources

1. Edwin Chen (Data scientist at Twitter), ggplot2 tutorial:

   `https://github.com/echen/ggplot2-tutorial`

2. Ramon Saccilotto (Basel Institute for Clinical Epidemiology and Biostatistics), ggplot2 tutorial:

   `http://www.ceb-institute.org/bbs/wp-content/uploads/2011/09/handout_ggplot2.pdf`

3. ggplot2 documentation: `http://docs.ggplot2.org/current/`

What have we learned?

1. R is a free programming tool that allows people to manipulate and analyze data.
2. ggplot allows us to create plots layer by layer.
3. The main graphs for analyzing data are barplots, scatterplots, line plots and histograms.
4. We can use faceting, blending, jittering, colours and sizes to create better visualizations.