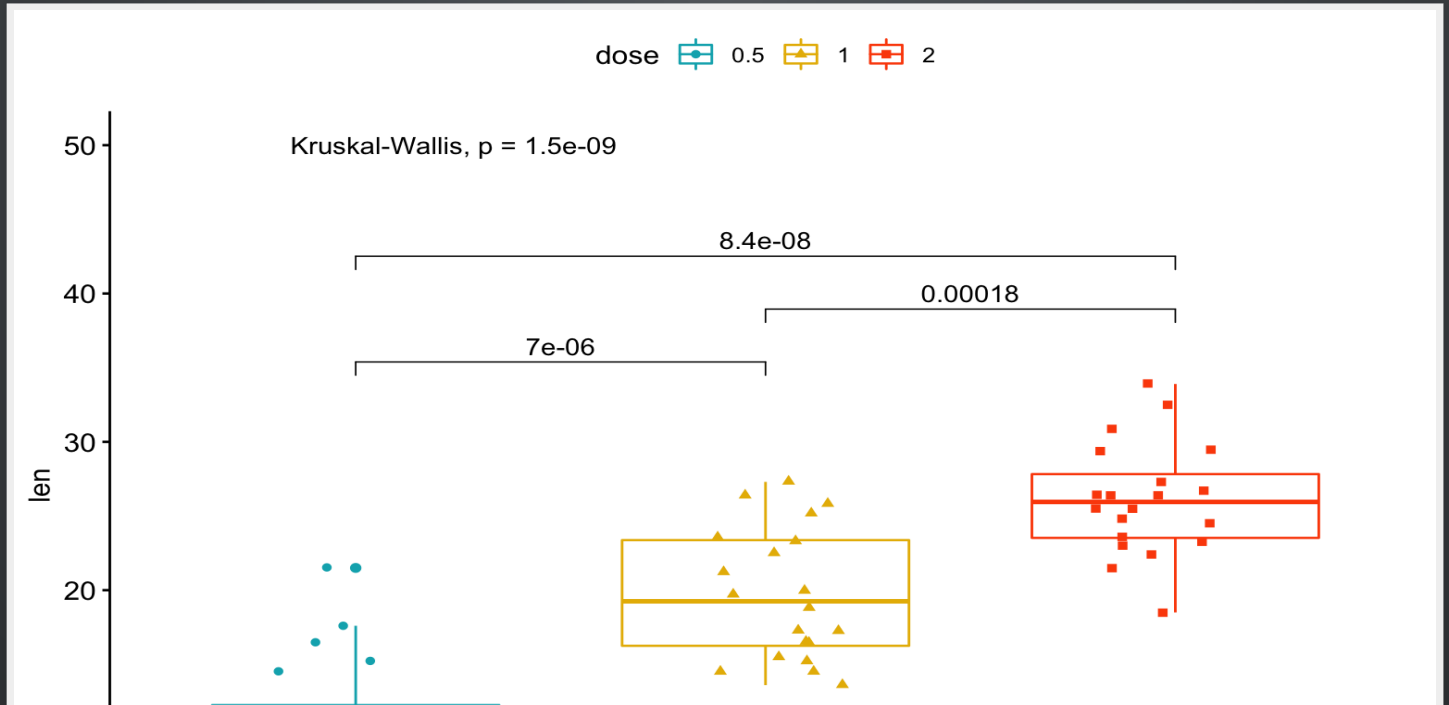
LECTURE 4: DATA VISUALIZATION

BIOF 339

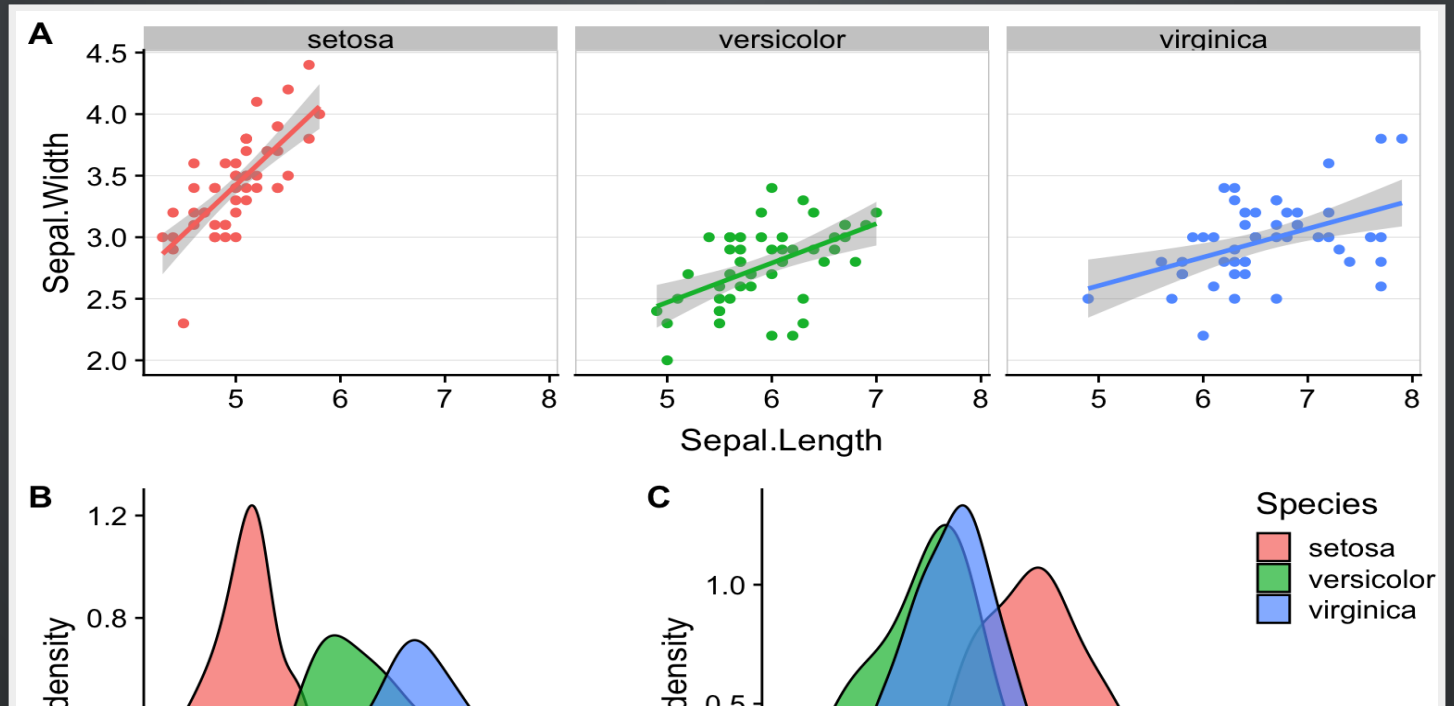
OCTOBER 3, 2018

DATA VISUALIZATION IN R

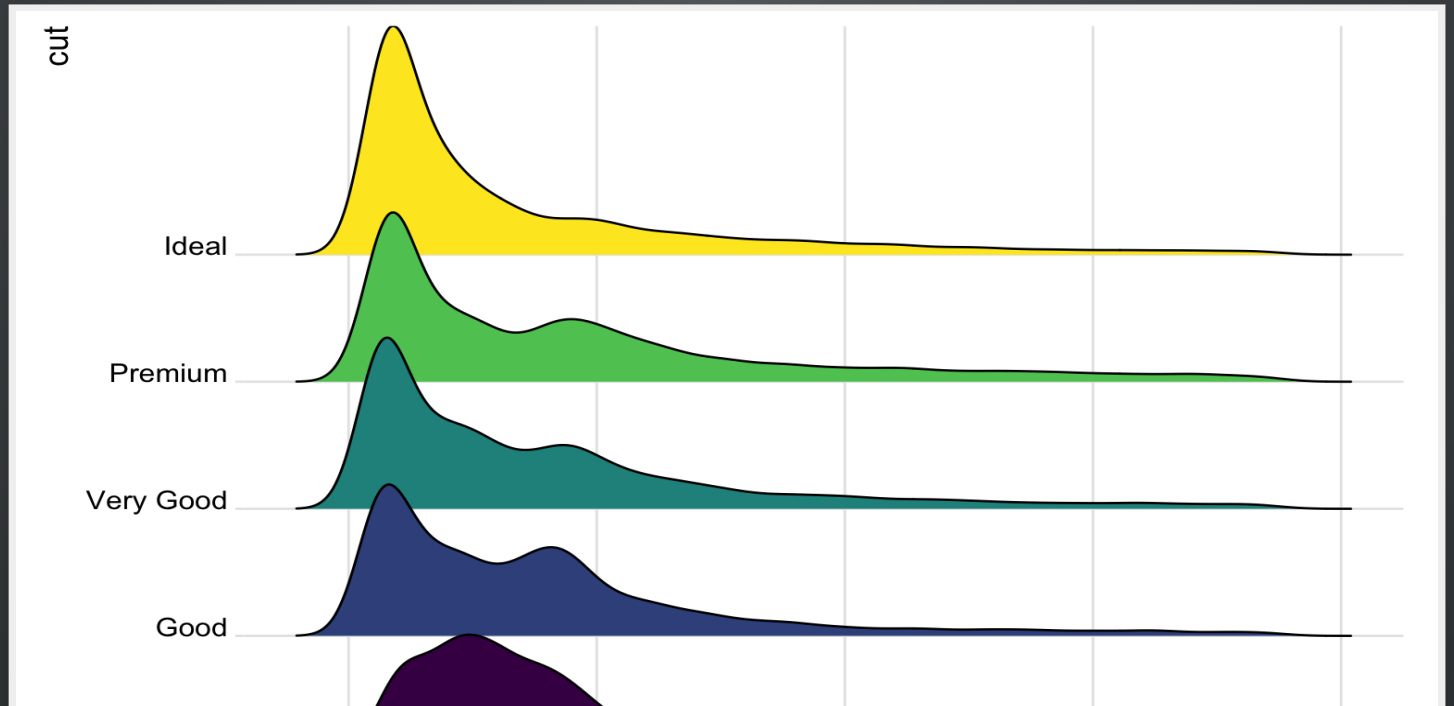
GALLERY


dose
0.5
1
2
Kruskal-Wallis, p = 1.5e-09
8.4e-08
0.00018
7e-06
len
50
40
30
20

GALLERY

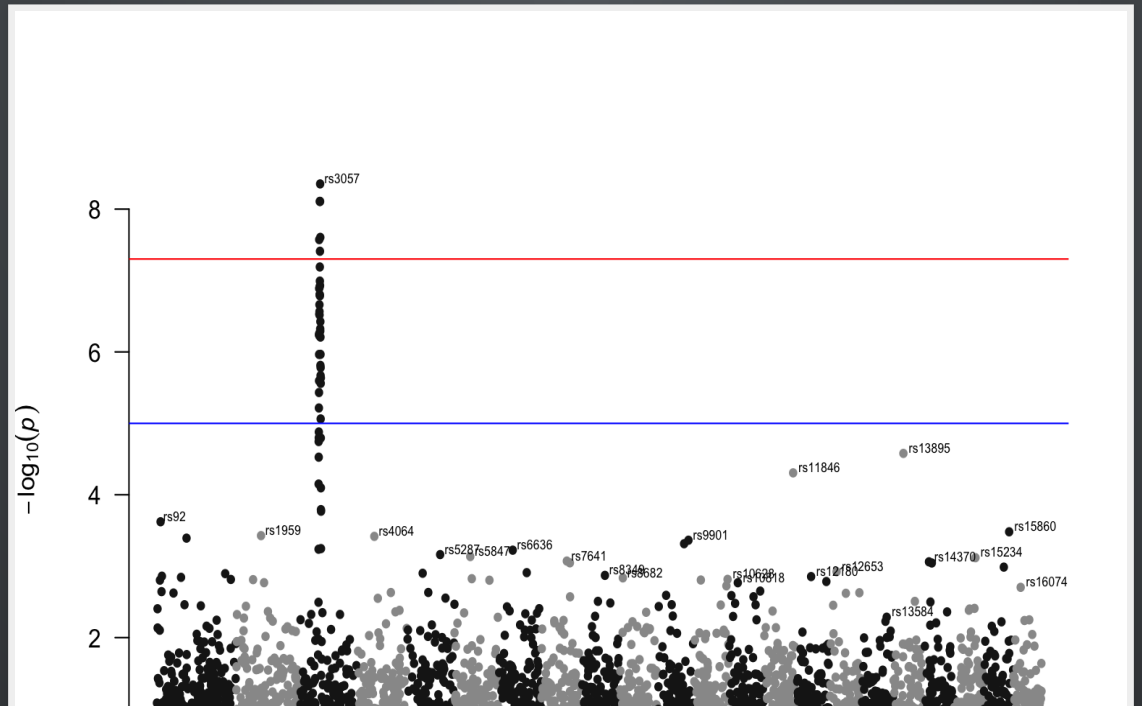

A
setosa
versicolor
virginica
Sepal.Width
Sepal.Length
B
density
C
density
Species
setosa
versicolor
virginica

GALLERY



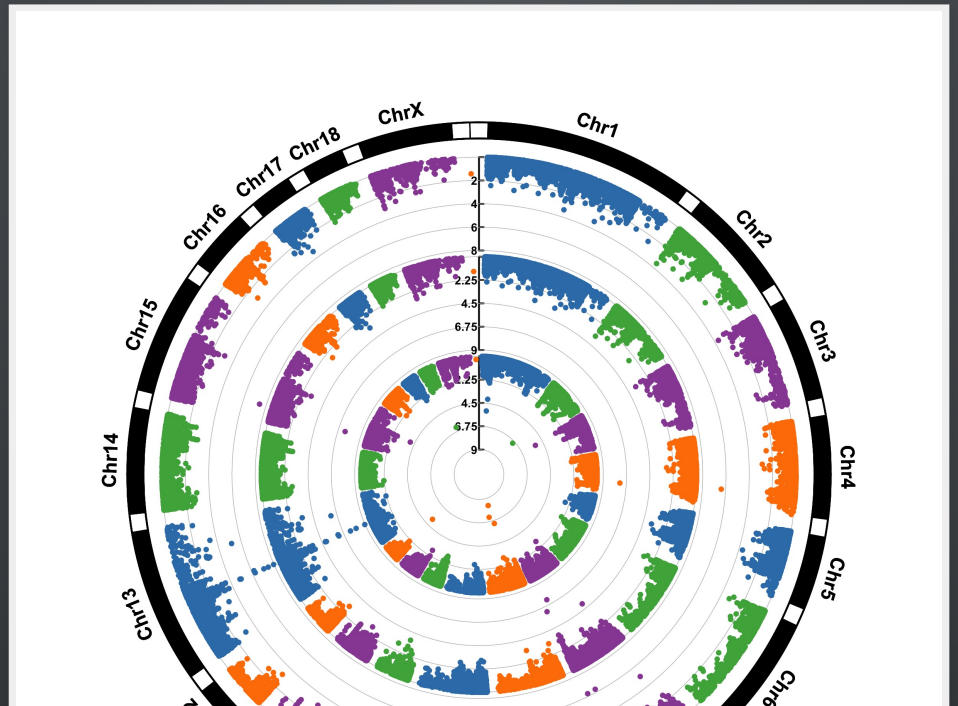
GALLERY

MANHATTAN PLOT



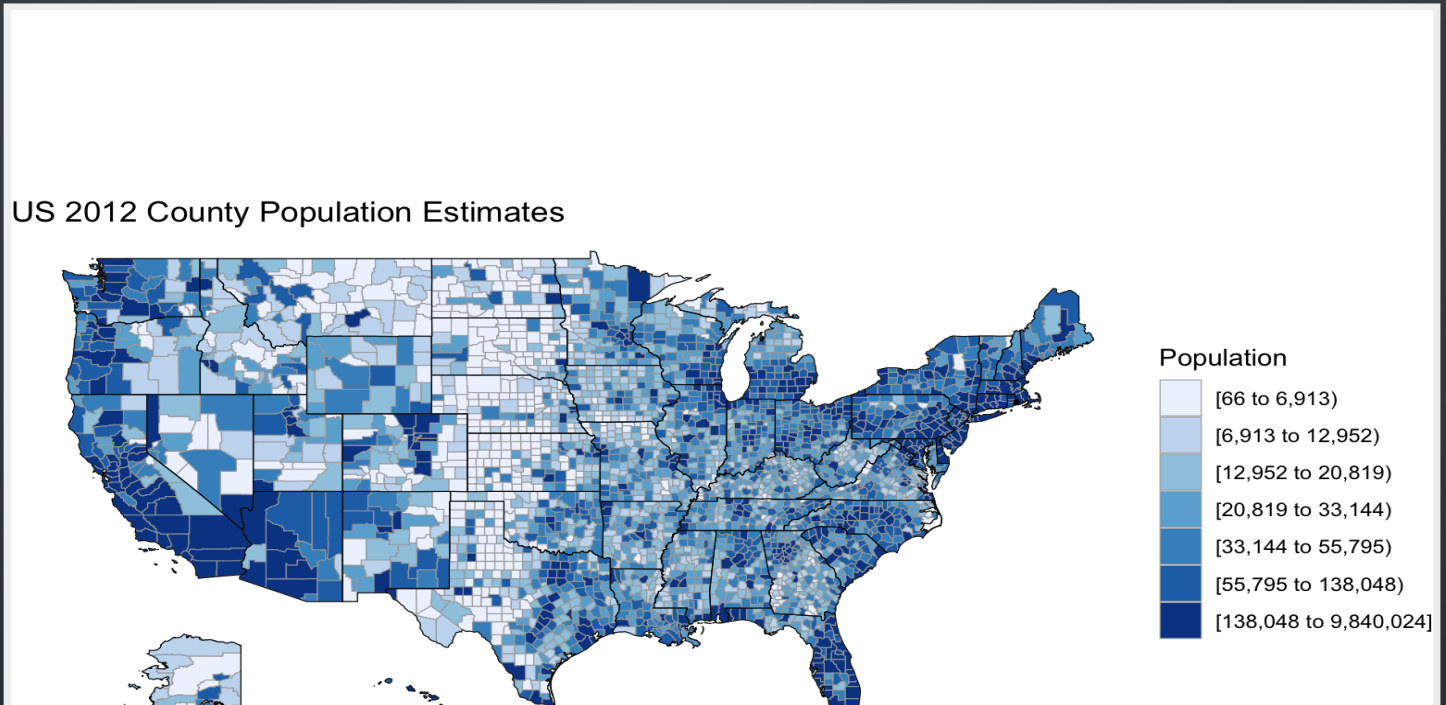
GALLERY

CIRCULAR MANHATTAN PLOT



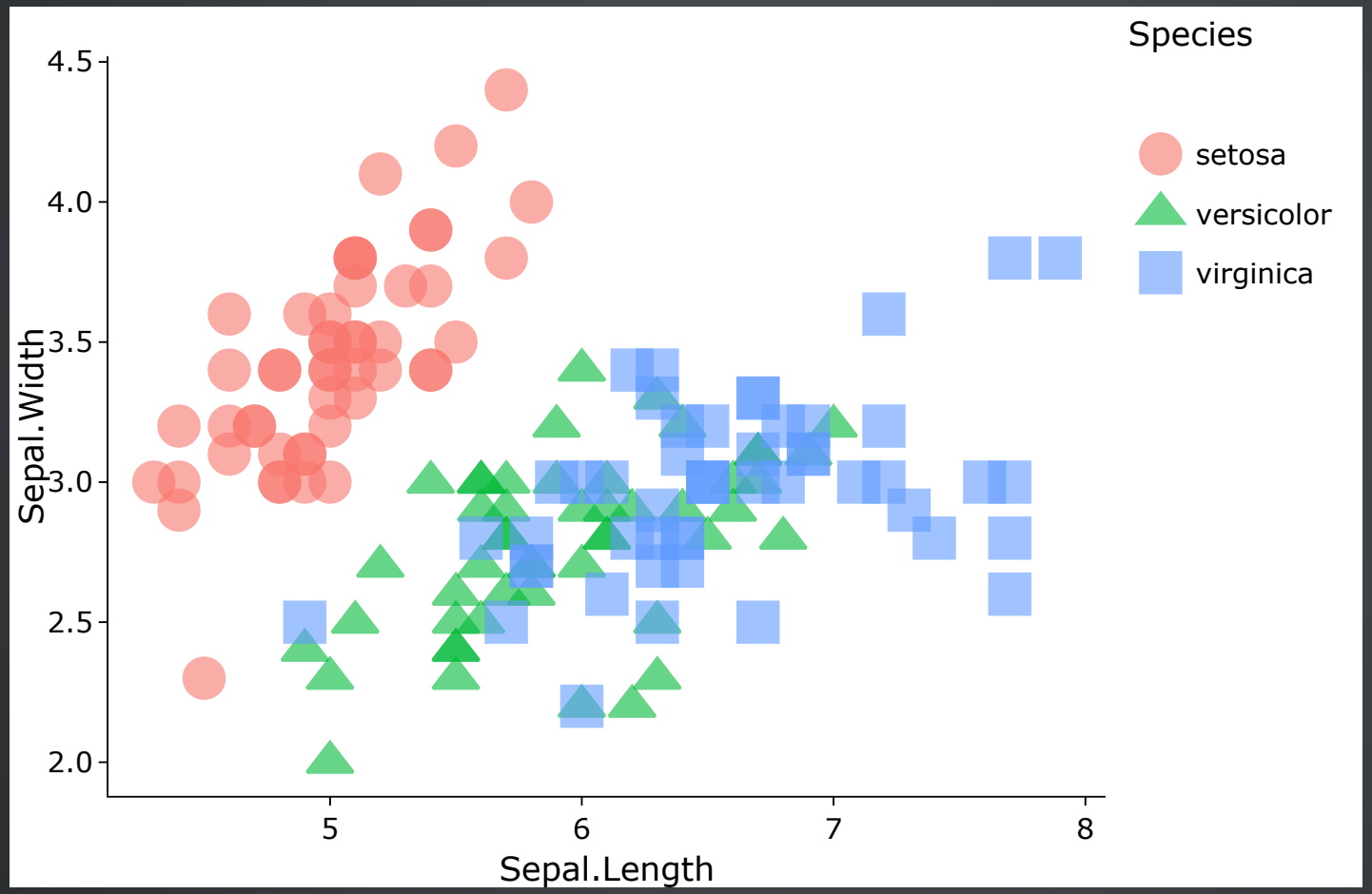
GALLERY

MAPS



GALLERY

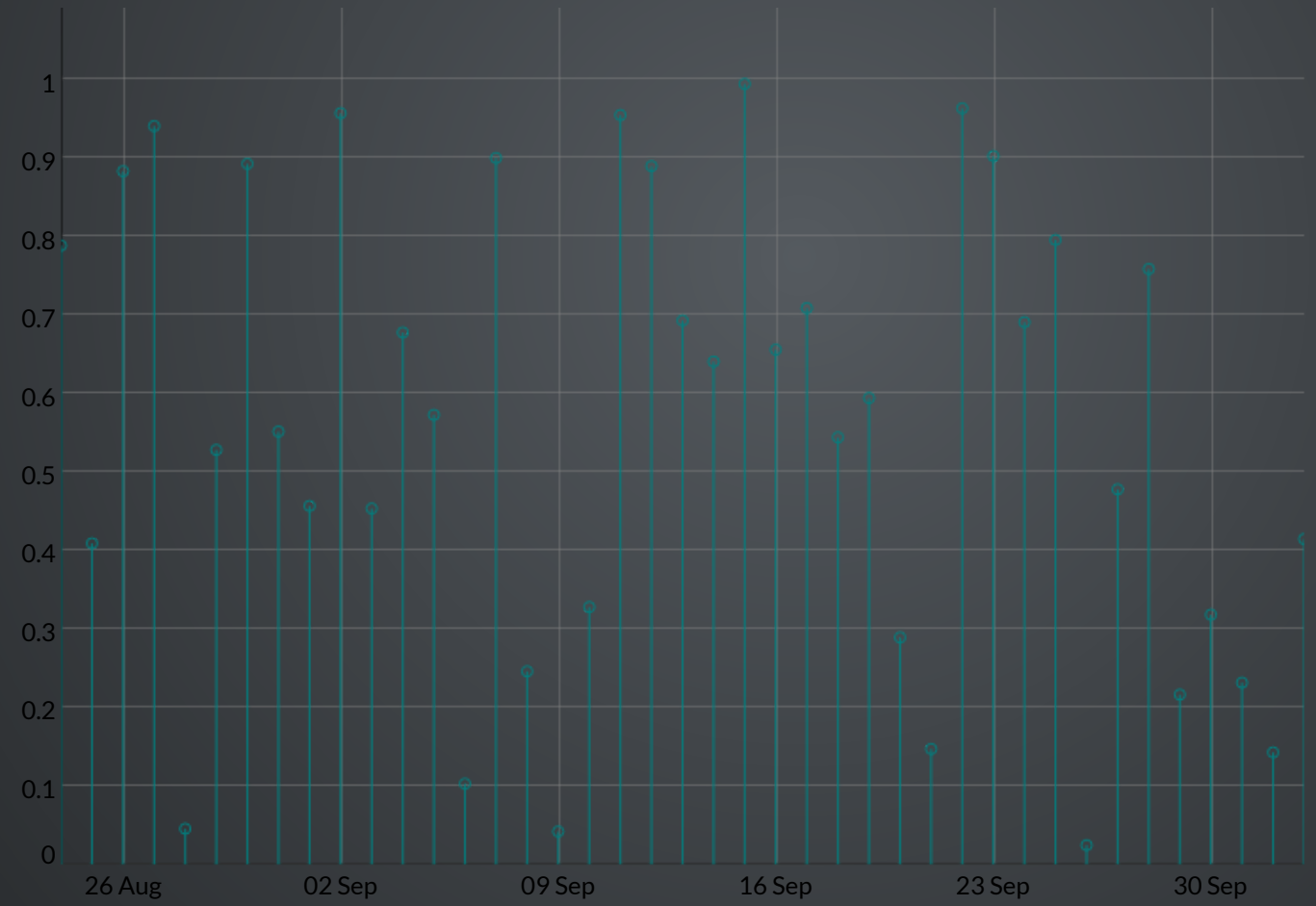
INTERACTIVE GRAPHS



GALLERY

INTERACTIVE GRAPHS

```
#     Date[1:41], format: "2018-08-24" "2018-08-25" "2018-08-26" "2018-08-
```



GALLERY

ANIMATION

GGPLOT2

We’re making the decision to use `ggplot2` for my graphics

- Makes pretty good formatting choices out of the box
- Works like pipes!!
- Is declarative (tell it what you want) without getting caught up in minutae
- Strongly leverages data frames (good practice)
- Fast enough
- There are good templates if you want to change the look

INTRODUCTION TO GGPLOT2

```
# install.packages('tidyverse')
library(ggplot2)
```

INTRODUCTION TO GGPLOT2

The `ggplot2` package is a very flexible and (to me) intuitive way of visualizing data. It is based on the concept of layering elements on a canvas.

> *This idea of layering graphics on a canvas is, to me, a nice way of building graphs*

INTRODUCTION TO GGPLOT2

You need:

- A `data.frame` object
- *Aesthetic mappings* (aes) to say what data is used for what purpose in the viz
  - x- and y-direction
  - shapes, colors, lines
- A *geometry object* (geom) to say what to draw
  - You can “layer” geoms on each other to build plots

INTRODUCTION TO GGPLOT2

`ggplot` used pipes before pipes were a thing. However, it uses the `+` symbol for piping rather than the `%>%` operator, since it pre-dates the `tidyverse`

INTRODUCTION TO GGPLOT2

```
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()
```

- A `data.frame` object: mtcars
- Aesthetic mapping:
  - x-axis: wt
  - y-axis: mpg
- Geometry:
  - geom_point: draw points

INTRODUCTION TO GGPLOT2

```
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()+ geom_smooth()
```

- A `data.frame` object: mtcars
- Aesthetic mapping:
  - x-axis: wt
  - y-axis: mpg
- Geometry:
  - geom_point: draw points
  - geom_smooth: Add a layer which draws a best-fitting line

GGPLOT2 EXAMPLES

We will use the two data sets:

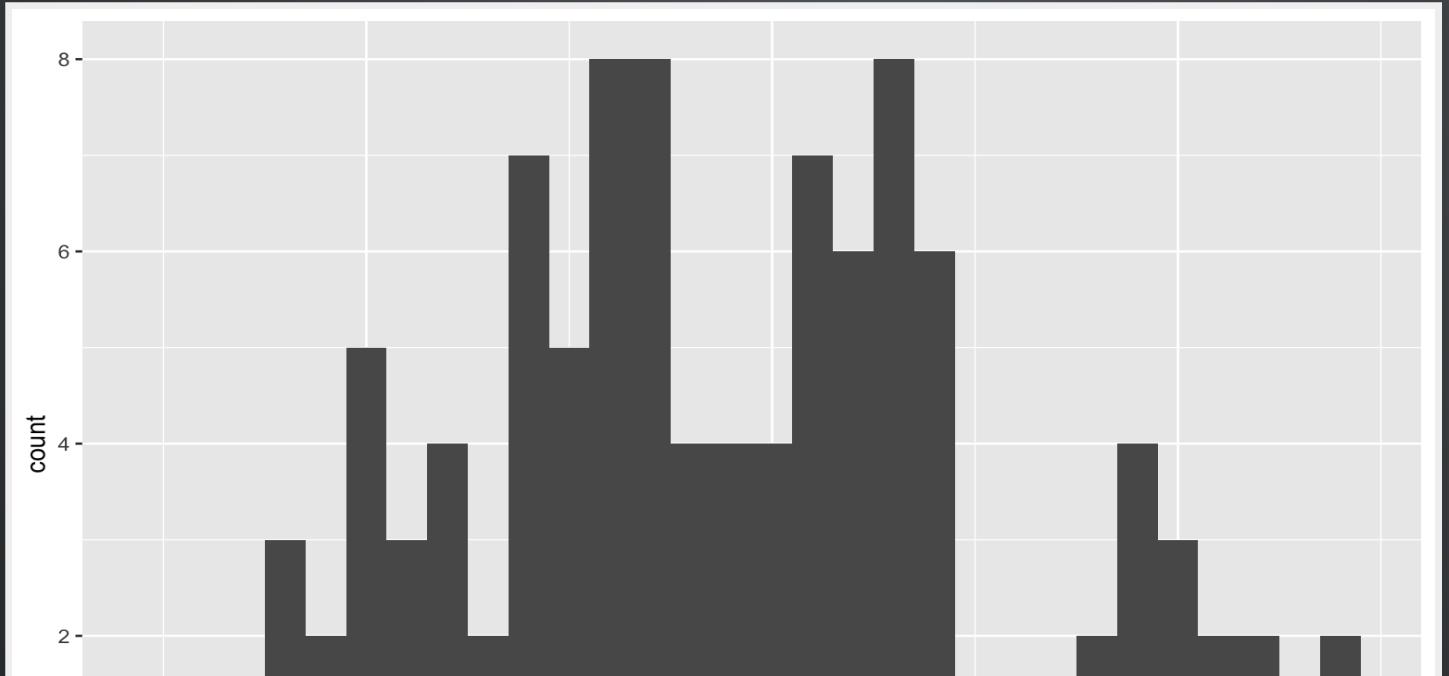
```
data_spine <- read.csv('http://www.araastat.com/BIOF339_PracticalR/
                        Lectures/lecture2_data/Dataset_spine.csv',
                        stringsAsFactors = F)


data_brca <- read.csv('http://www.araastat.com/BIOF339_PracticalR/
                       Lectures/lecture2_data/
                       clinical_data_breast_cancer_modified.csv',
                       stringsAsFactors = F)
```

PLOTTING ONE VARIABLE

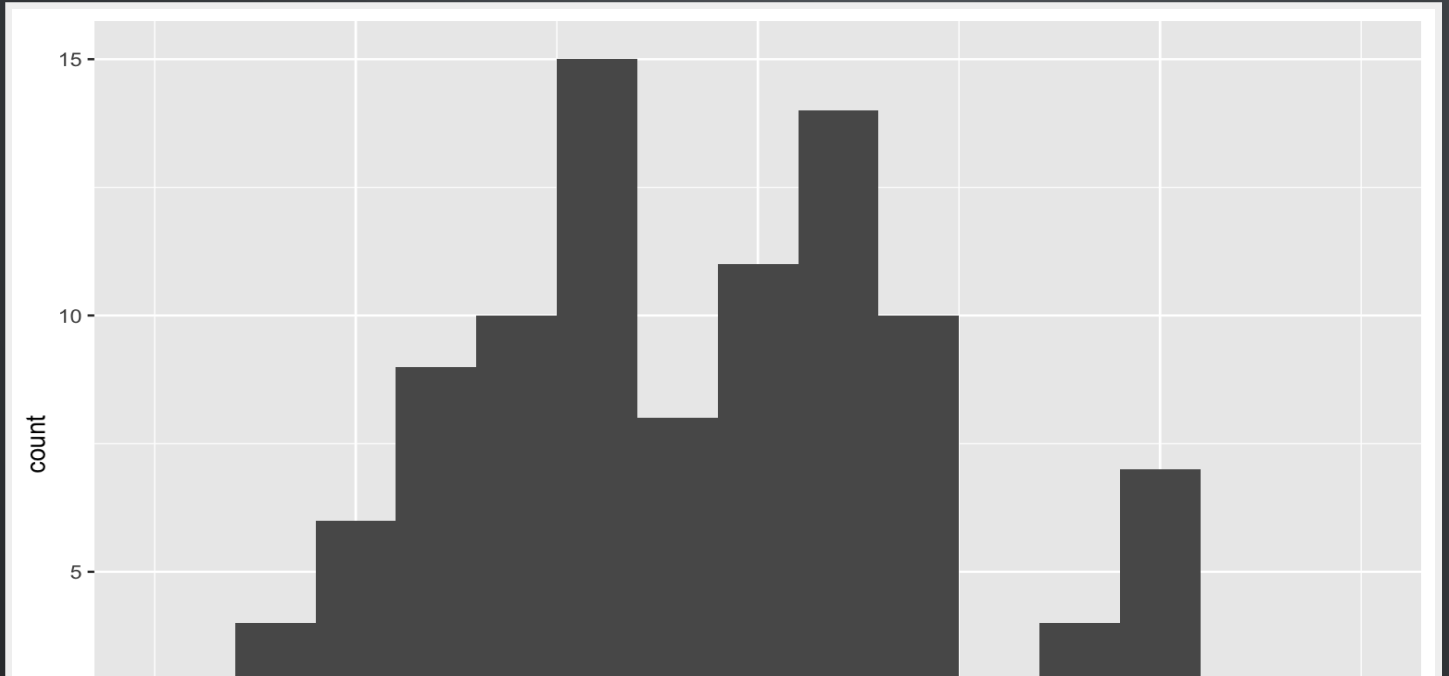
HISTOGRAMS

```
ggplot(data_brca, aes(x = Age.at.Initial.Pathologic.Diagnosis)) +
  geom_histogram()
```



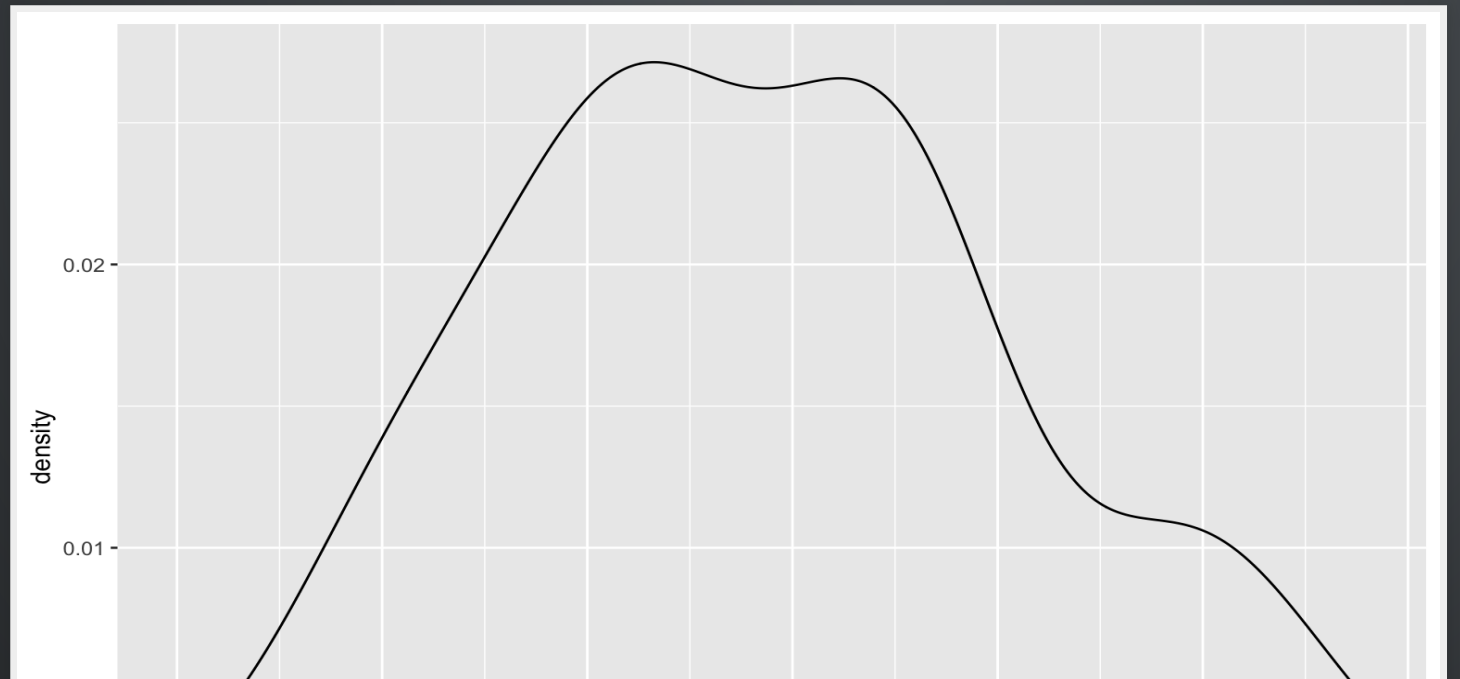
HISTOGRAMS

```
ggplot(data_brca, aes(x = Age.at.Initial.Pathologic.Diagnosis)) +
  geom_histogram(binwidth=4)
```



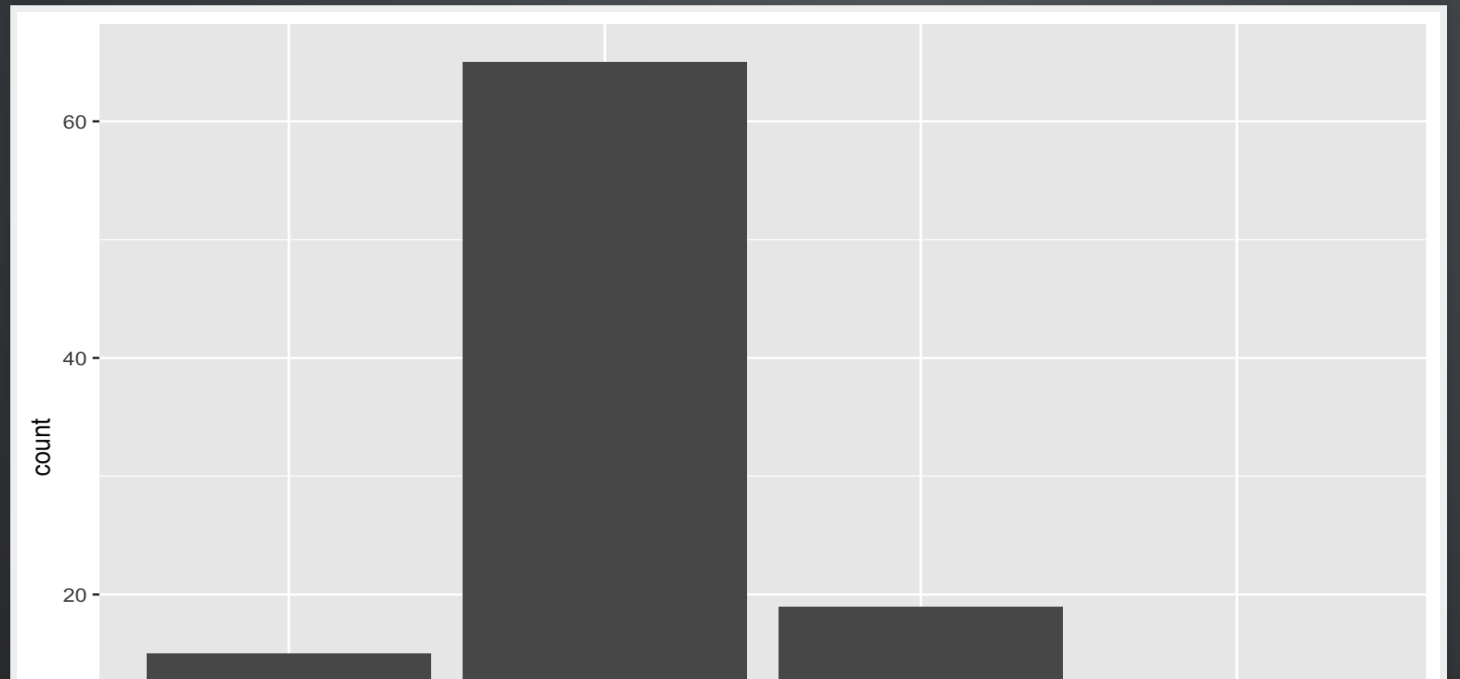
DENSITY PLOT

```
ggplot(data_brca, aes(x = Age.at.Initial.Pathologic.Diagnosis)) +
  geom_density()
```



BAR PLOT

```
ggplot(data_brca, aes(x = Tumor))+geom_bar()
```



EXERCISE

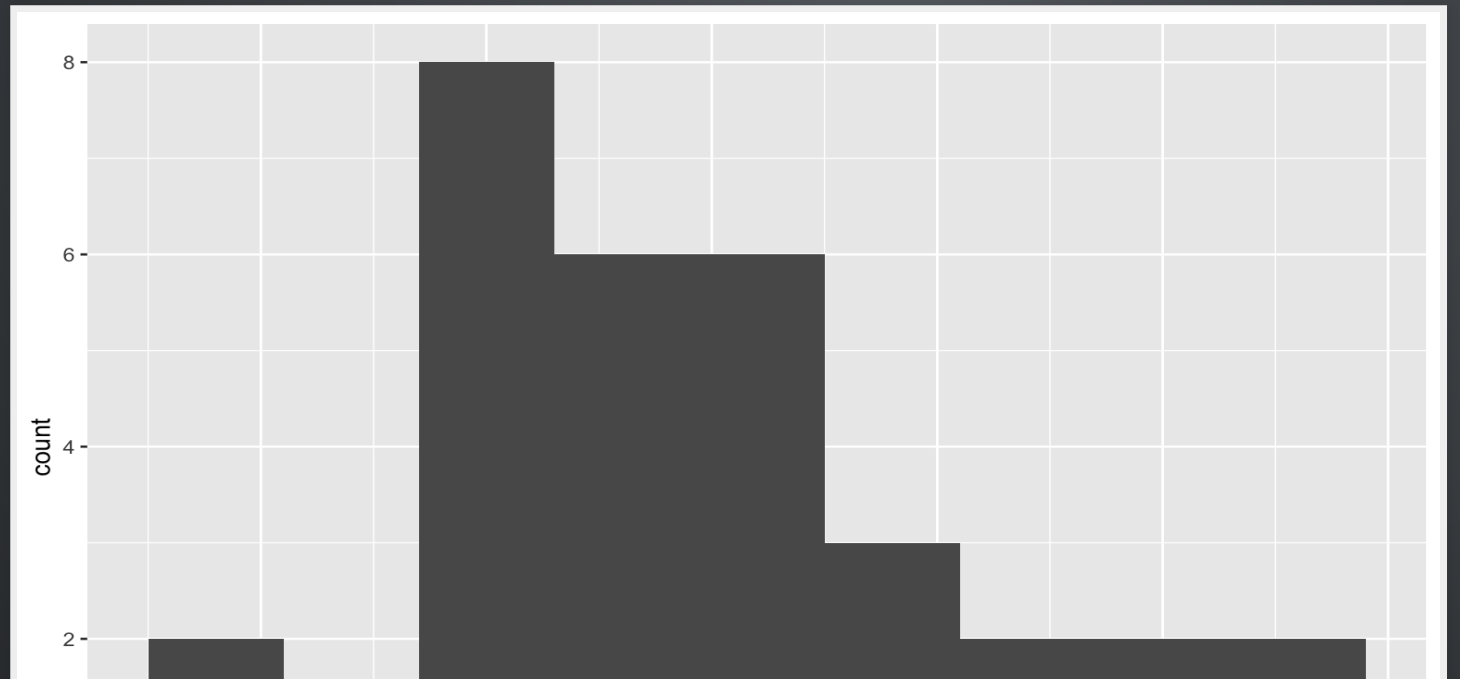
EXERCISE

Using the `mtcars` dataset in R, create:

1. A histogram of the fuel efficiences (`mpg`) in the data set
2. A bar plot of frequencies of number of cylinders (`cyl`) in the car

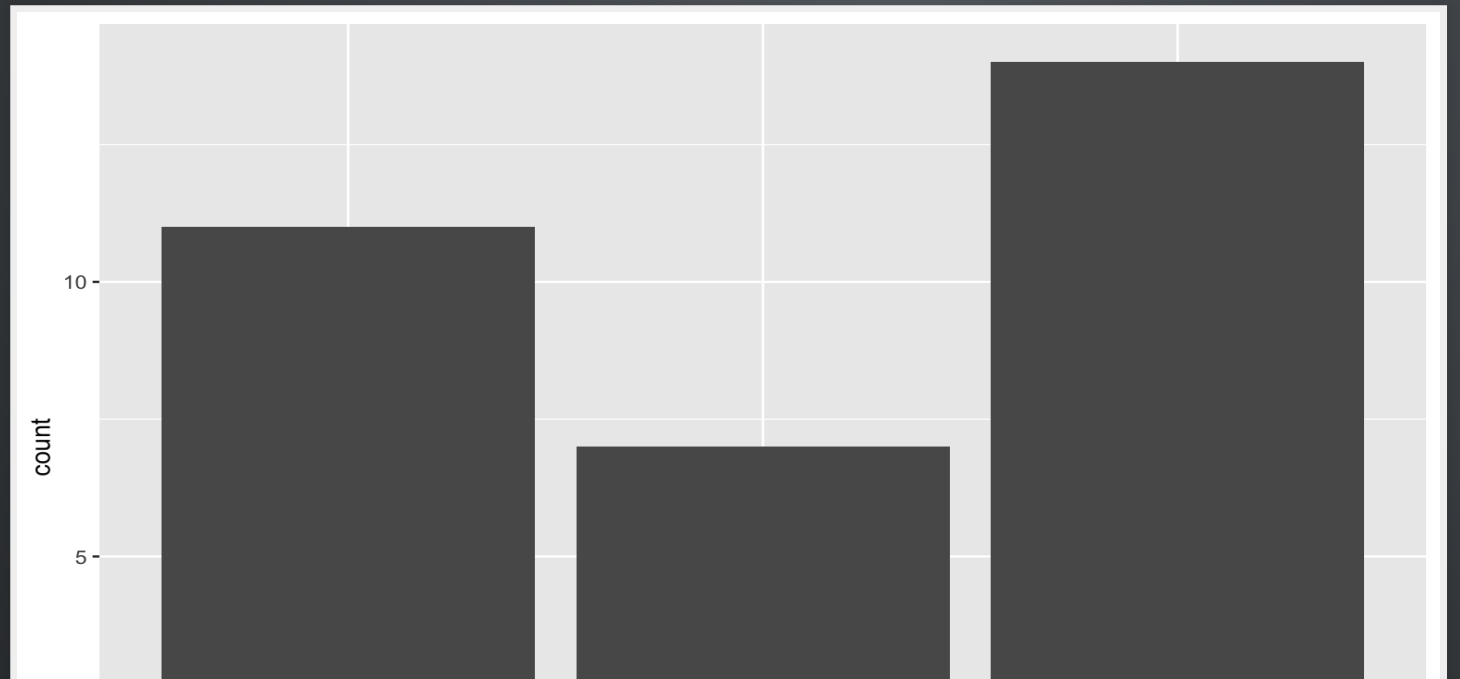
SOLUTION

```
ggplot(mtcars, aes(x = mpg)) + geom_histogram(binwidth=3)
```



SOLUTION

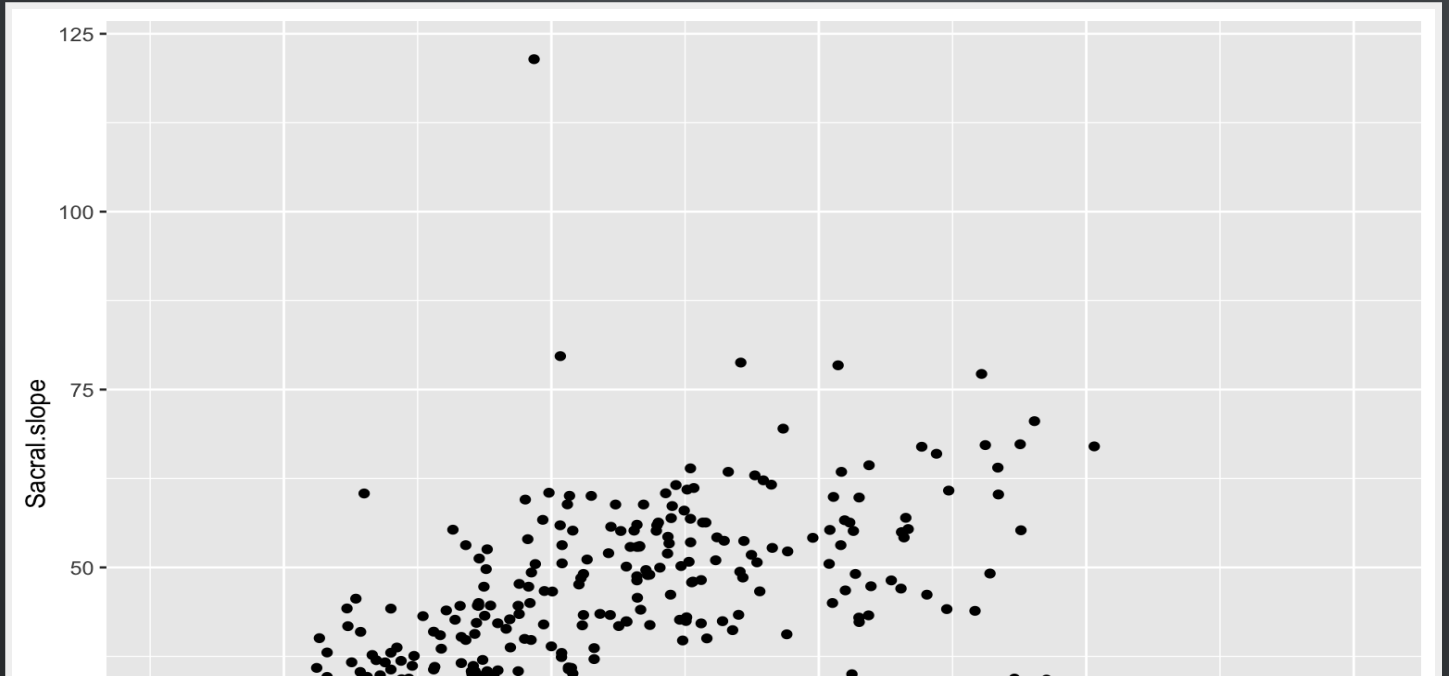
```
ggplot(mtcars, aes(x = factor(cyl))) + geom_bar()
```



TWO CONTINUOUS VARIABLES

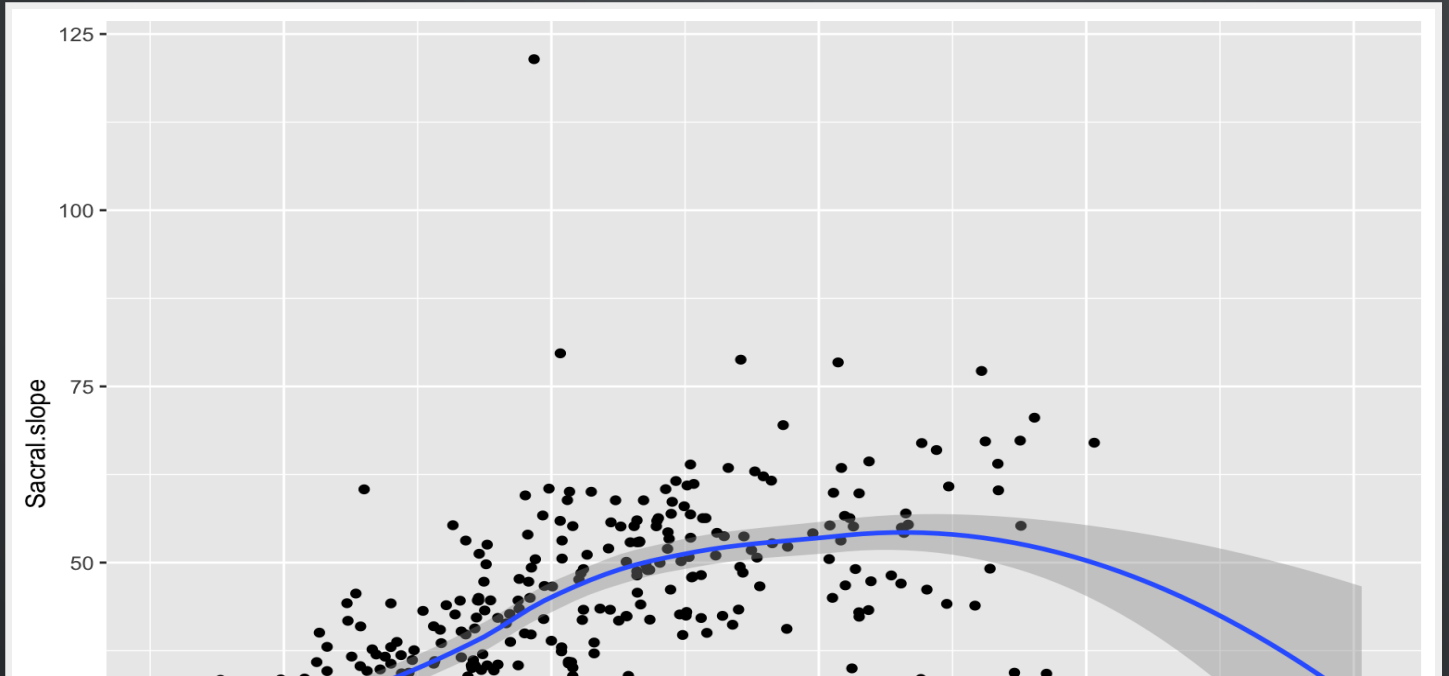
SCATTER PLOTS

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope)) +
  geom_point()
```



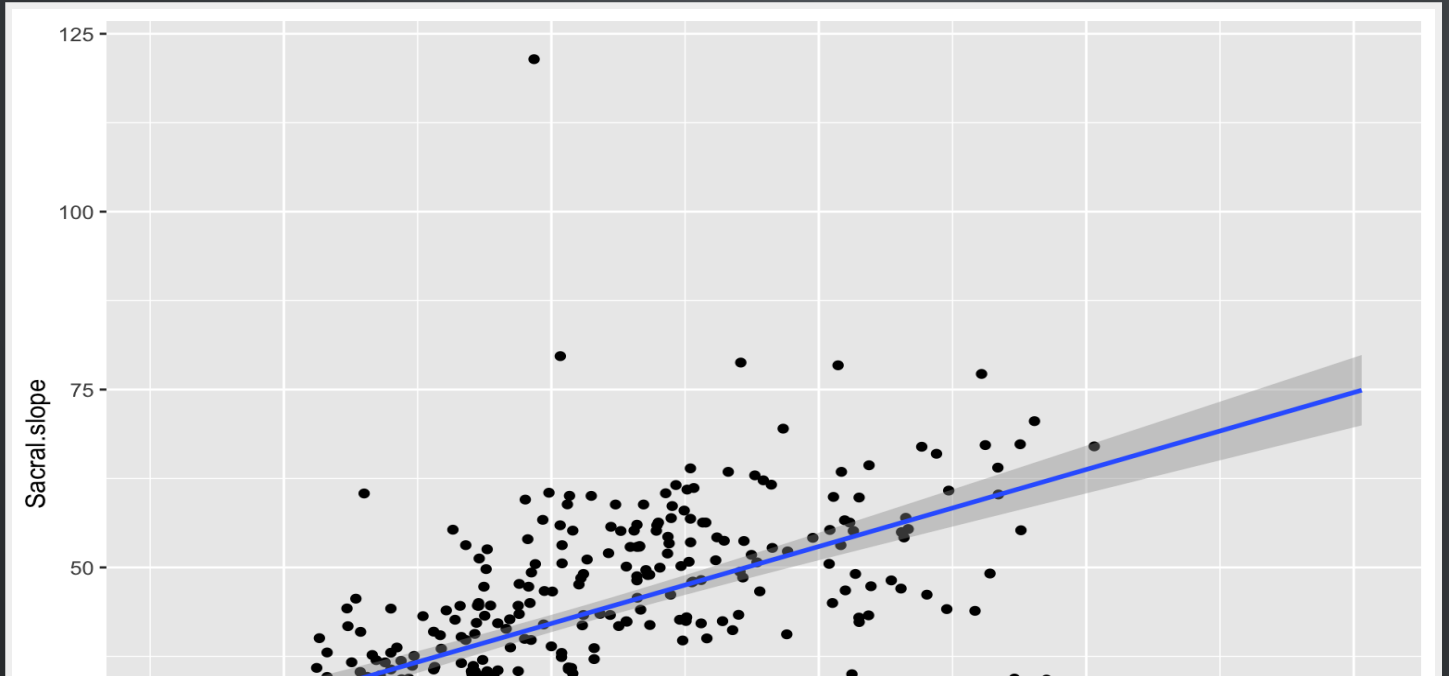
SCATTER PLOT WITH A SMOOTH LINE

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope))+
  geom_point() +
  geom_smooth()
```



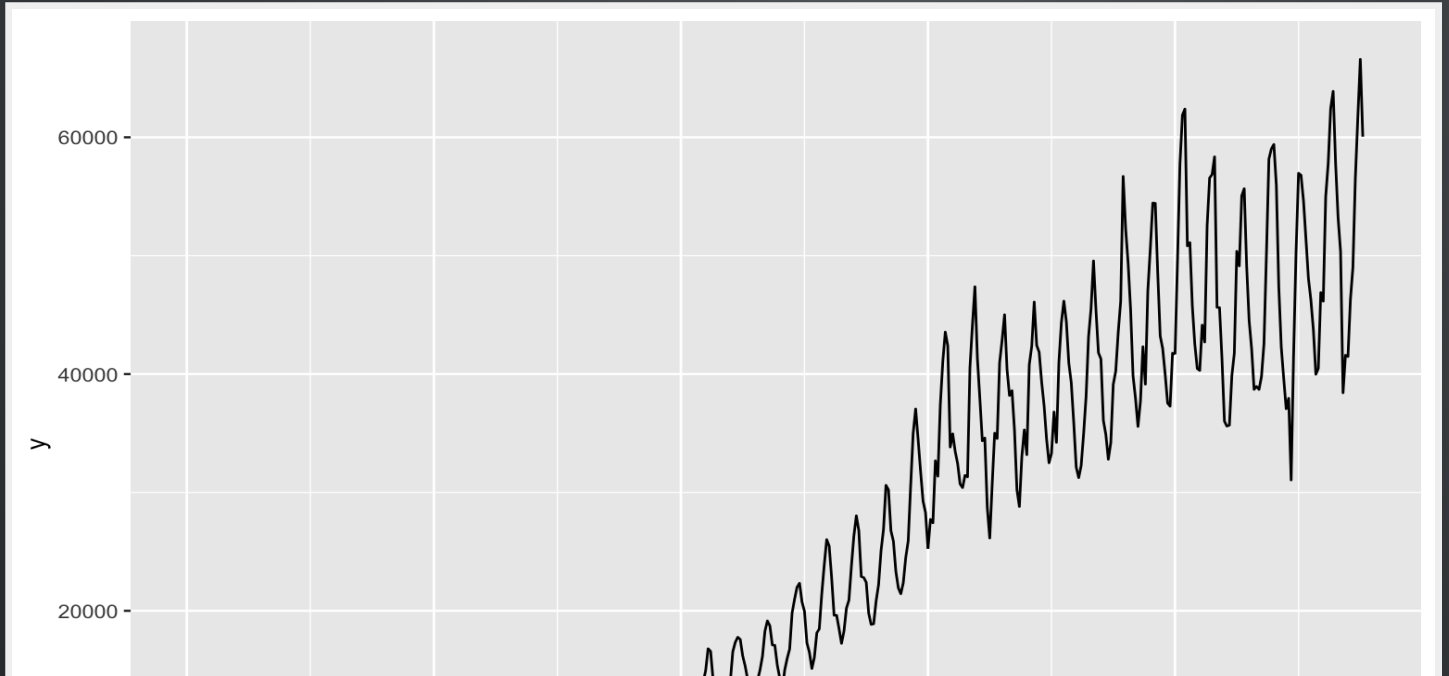
SCATTER PLOT WITH A SMOOTH STRAIGHT LINE

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope)) +
  geom_point()+
  geom_smooth(method='lm')
```



LINE PLOT (FOR TIME SERIES)

```
library(forecast)
d <- data.frame(x = 1:length(gas), y = gas) # Australian monthly gas pro
ggplot(d, aes(x, y)) + geom_line()
```



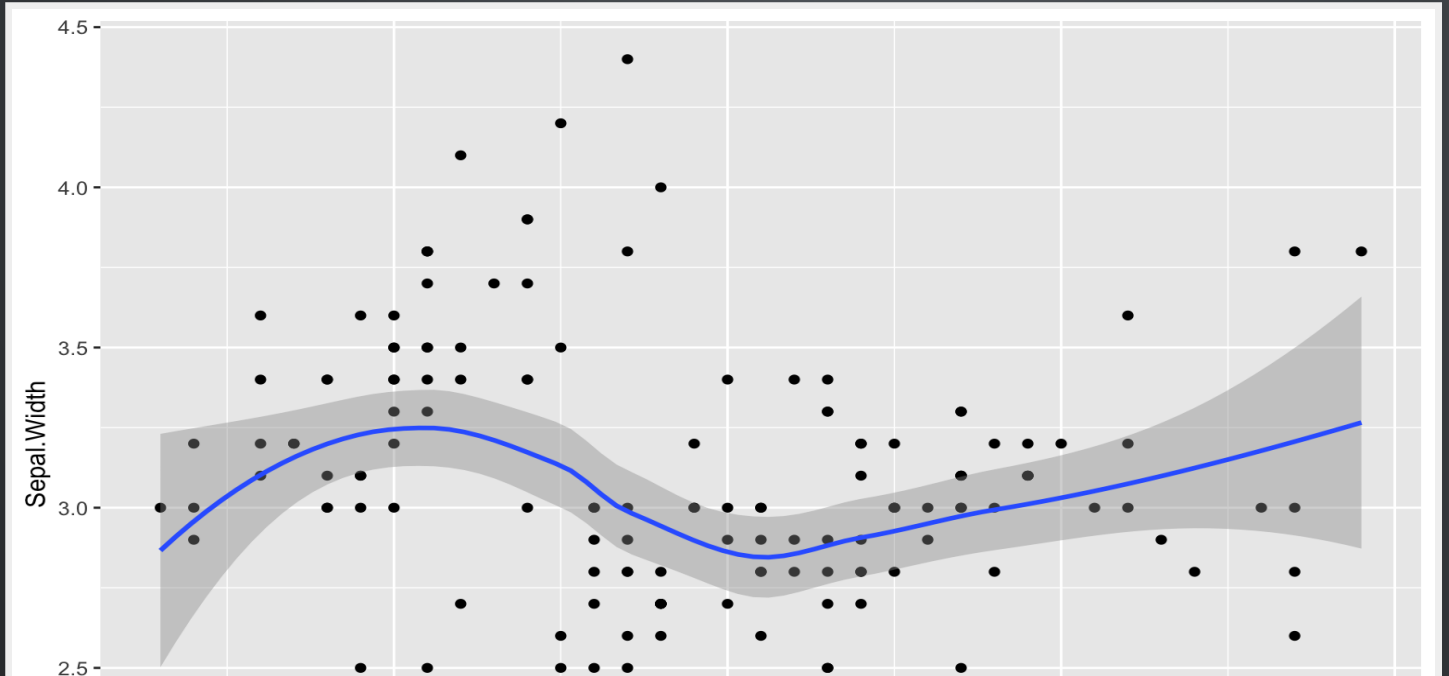
EXERCISE

EXERCISE

1. Create a scatter plot of sepal length and sepal width from the `iris` dataset, and add a smooth line through it

SOLUTION

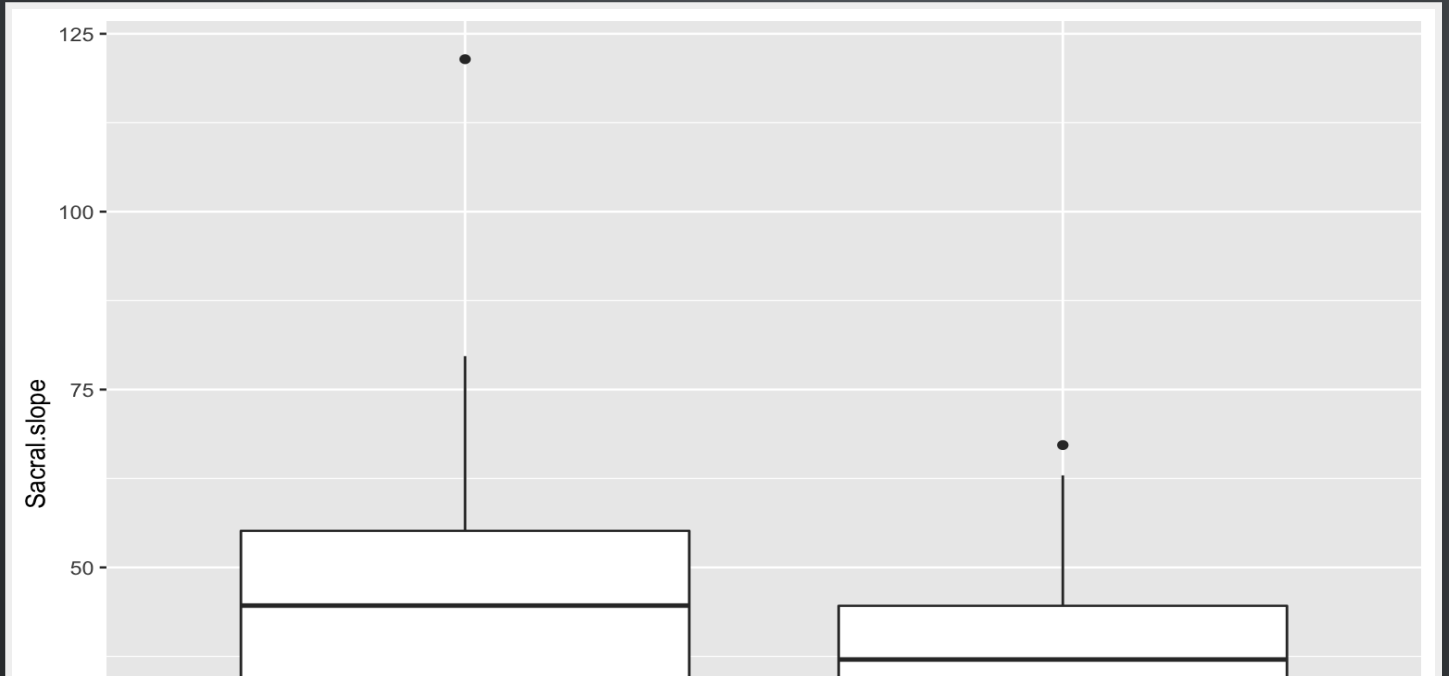
```
ggplot(iris, aes(Sepal.Length, Sepal.Width)) + geom_point() + geom_smoothl
```



CONTINUOUS VARIABLE WITH DISCRETE VARIABLE

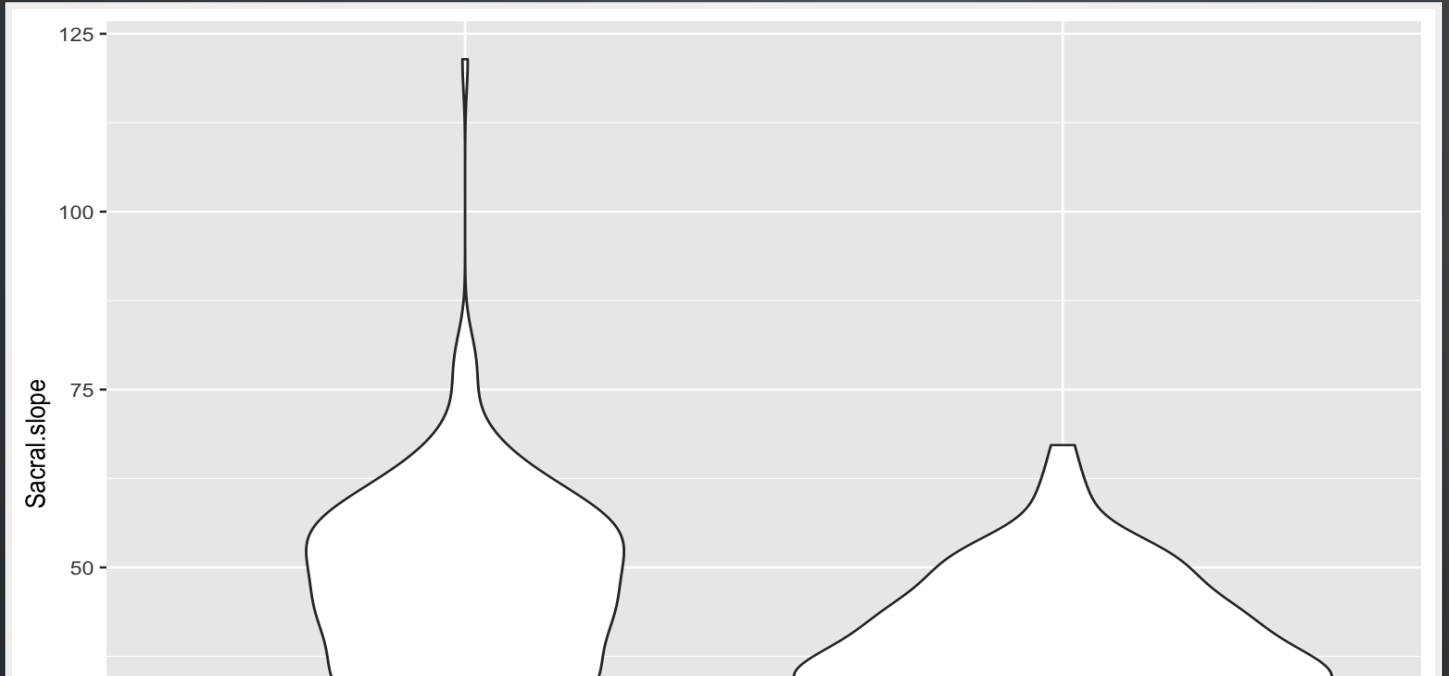
BOXPLOTS

```
ggplot(data_spine, aes(x = Class.attribute, y = Sacral.slope))+
  geom_boxplot()
```



VIOLIN PLOTS

```
ggplot(data_spine, aes(x = Class.attribute, y = Sacral.slope)) +
  geom_violin()
```



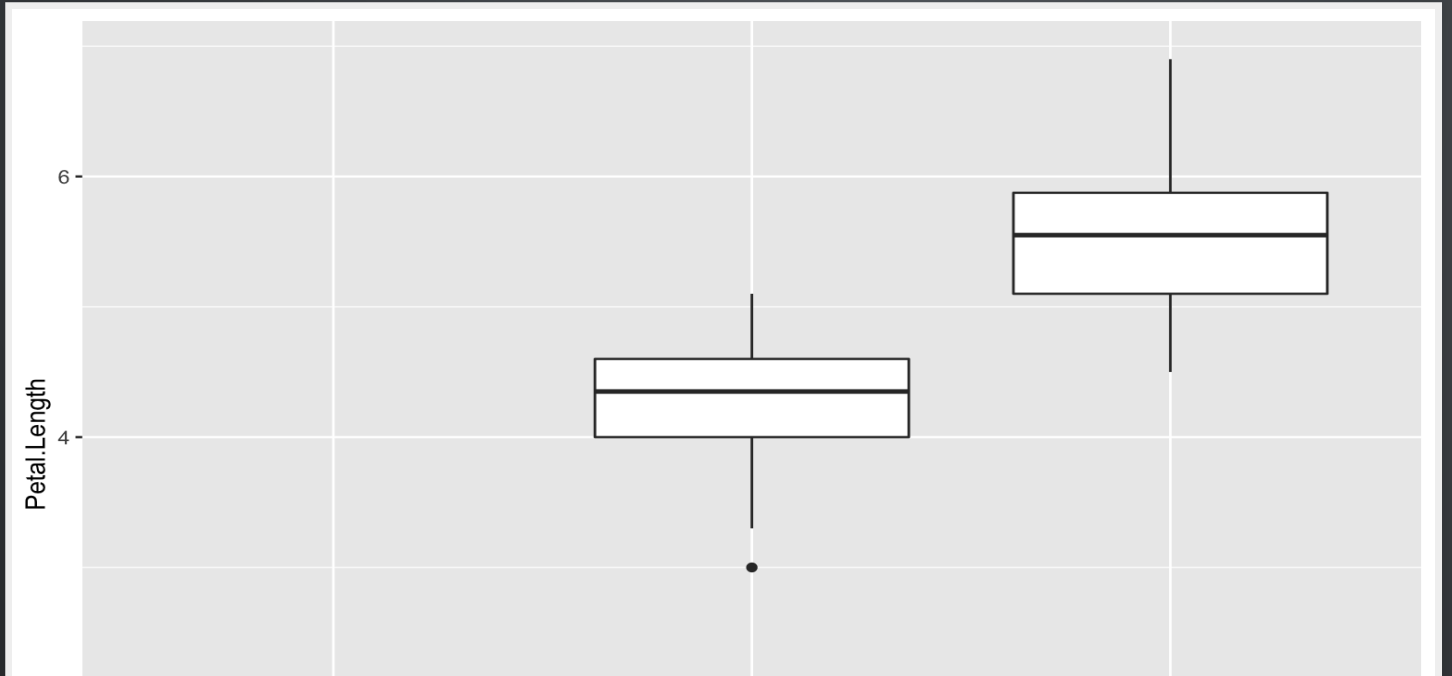
EXERCISE

EXERCISE

1. Plot a boxplot of petal length by species using the `iris` dataset

SOLUTION

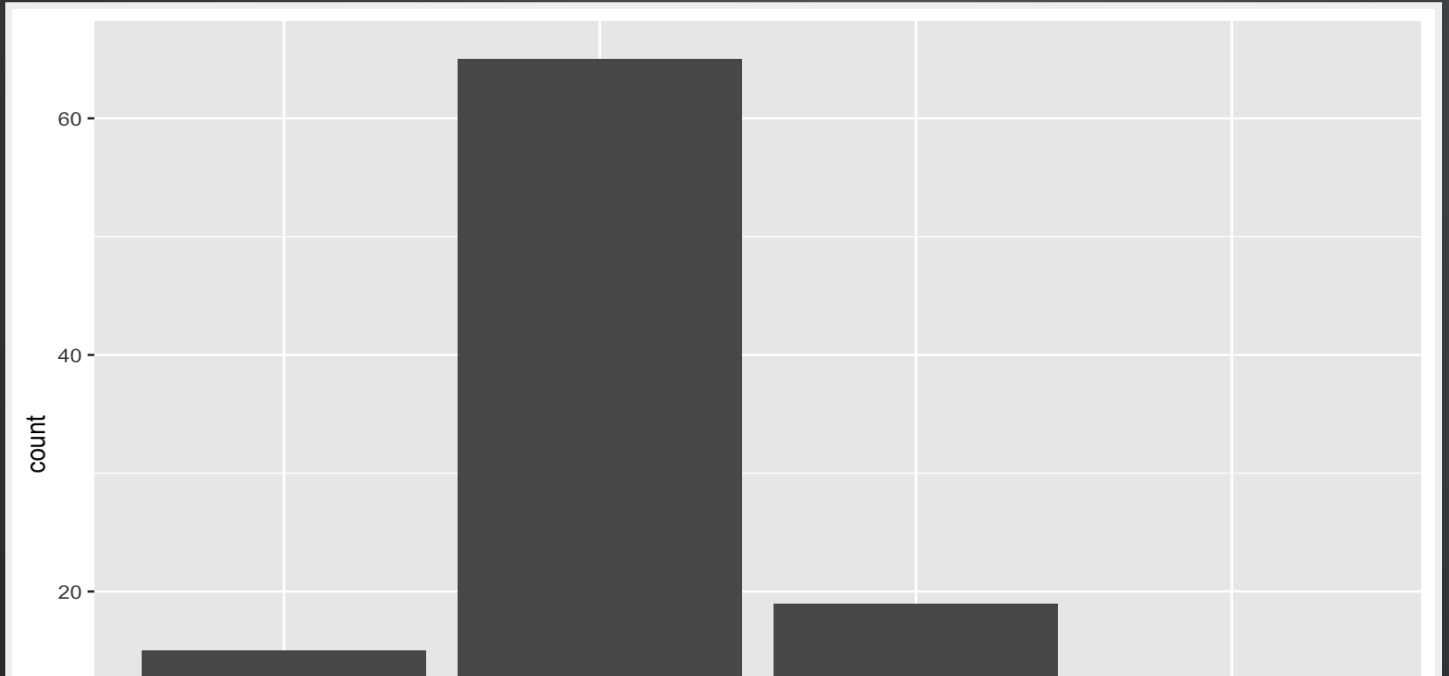
```
ggplot(iris, aes(x = Species, y = Petal.Length))+geom_boxplot()
```



FLIPPING AXES

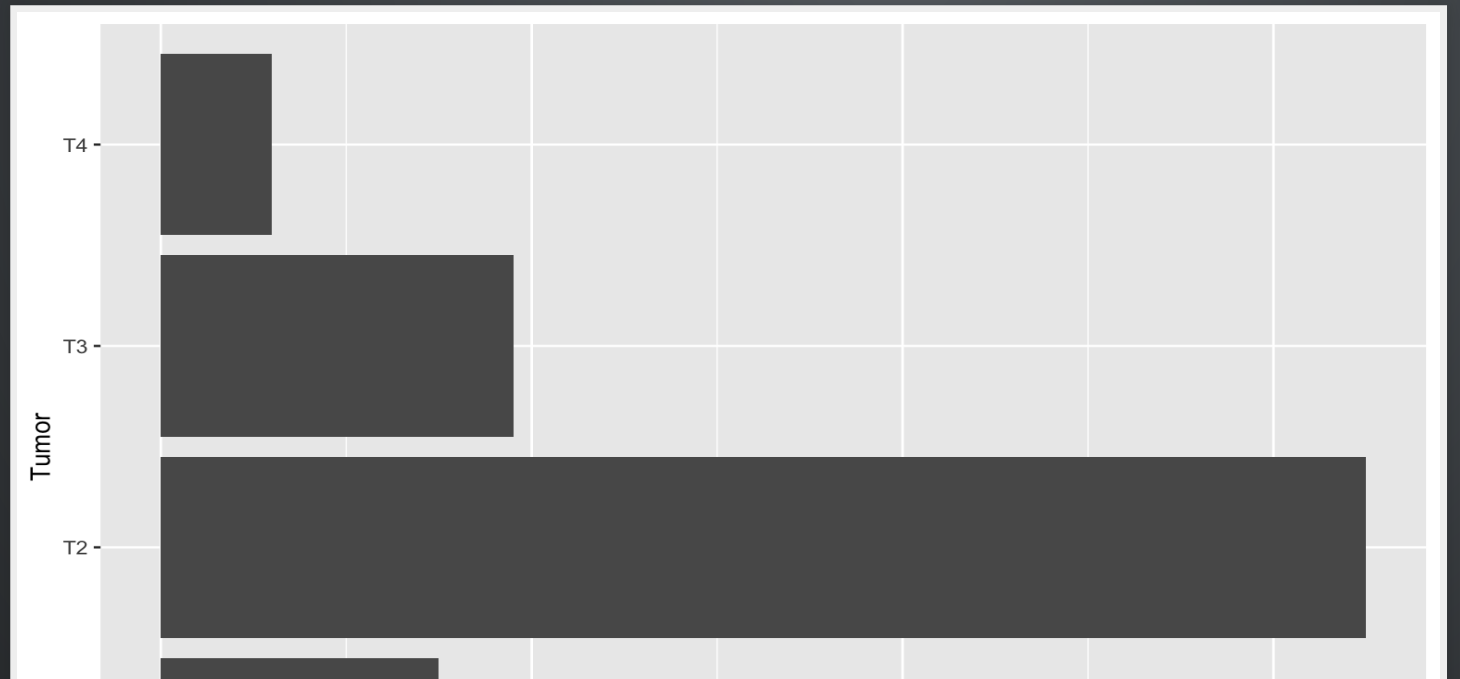
VERTICAL BARS

```
ggplot(data_brca, aes(x = Tumor))+geom_bar()
```



HORIZONTAL BARS

```
ggplot(data_brca, aes(x = Tumor))+geom_bar()+
  coord_flip()
```



RESOURCES

ONLINE RESOURCES

- The ggplot website has many resources to help create visualizations
- The R Graph Gallery
- There are a lot of blogs showing many capabilities of ggplot2
- StackOverflow is the place for Q & A.

OTHER PACKAGES

- The `cowplot` and `ggpubr` packages provide several improvements on ggplot2, including more themes and an easy way to put several graphs together in a panel

GROUP-WISE DESCRIPTIVES AND VISUALIZATIONS

GROUPING

- It is common to look at statistics within subgroups of the data
- The idea is to see if secondary variables affect your primary outcome or relationship

INTRODUCING THE `DPLYR` PACKAGE

`dplyr` is the most lucid package for manipulating and analyzing data organized in a data frame.

- It has a `group_by` function which creates a *grouped data frame*

```
library(dplyr)
grouped_data_spine = data_spine %>% group_by(Class.attribute)
```

Note that you have to group using a discrete valued variable (factor, character, integer)

GROUPED SUMMARIES

```
grouped_data_spine %>%
  summarize(mean(Pelvic.incidence),
            sd(Pelvic.incidence),
            min(Pelvic.incidence),
            max(Pelvic.incidence))
```

Class.attribute	mean(Pelvic.incidence)	sd(Pelvic.incidence)	min(Pelvic.incidence)	max(Pelvic.
Abnormal	64.69256	17.66213	26.14792	1
Normal	51.68524	12.36816	30.74194	

GROUPED SUMMARIES

```
grouped_data_spine %>% summarize(Mean = mean(Pelvic.incidence),
                                 SD = sd(Pelvic.incidence),
                                 Min = min(Pelvic.incidence),
                                 Max = max(Pelvic.incidence))
```

Class.attribute	Mean	SD	Min	Max
Abnormal	64.69256	17.66213	26.14792	129.83404
Normal	51.68524	12.36816	30.74194	89.83468

GROUPED SUMMARIES

```
grouped_data_spine %>% summarize_all(mean)
```

```
#    # A tibble: 2 x 13
#      Class.attribute Pelvic.incidence Pelvic.tilt Lumbar.lordosis…
#      <fct>                      <dbl>       <dbl>             <dbl>
#    1 Abnormal                    64.7        19.8              55.9
#    2 Normal                      51.7        12.8              43.5
#    # ... with 9 more variables: Sacral.slope <dbl>, Pelvic.radius <dbl>,
#    #   Degree.spondylolisthesis <dbl>, Pelvic.slope <dbl>, Direct.tilt <
#    #   Thoracic.slope <dbl>, Cervical.tilt <dbl>, Sacrum.angle <dbl>,
#    #   Scoliosis.slope <dbl>
```

A NOTE ON TIBBLES

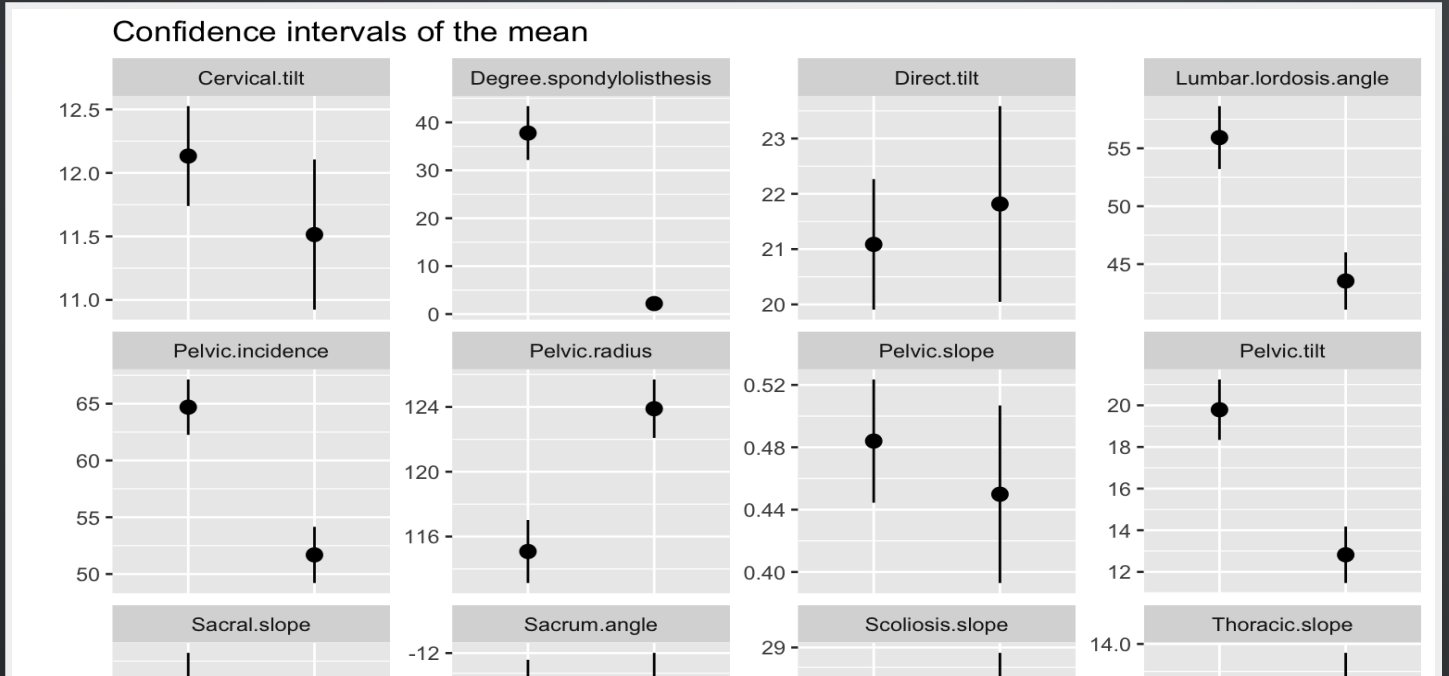
- Tibbles are a new-generation object meant to enhance the `data.frame`.
  - Central to the**tidyverse** packages
- If you want to just get back to a more familiar `data.frame` object, use `as.data.frame`
- A tibble is built on a `data.frame`, so all operations on `data.frame`'s will work.
- To see all columns, set

USING GGPLOT IN A PIPELINE

```
data_spine %>%
  group_by(Class.attribute) %>%
  summarize_all(funs(Mean = mean(., na.rm=T),
                     SEM = sd(., na.rm=T)/sqrt(n()))) %>%
  gather(variable, value, -Class.attribute) %>%
  separate(variable, c('Variable','stat'), sep = '_') %>%
  spread(stat, value) %>%
  mutate(lcb = Mean - 2 * SEM, ucb = Mean + 2 * SEM) %>%
  ggplot(aes(x = Class.attribute, y = Mean, ymin = lcb, ymax = ucb)) +

    geom_pointrange() +
    facet_wrap(~Variable, scales = 'free_y') +
    labs( x = 'Class', y = '') +
    ggtitle('Confidence intervals of the mean')
```

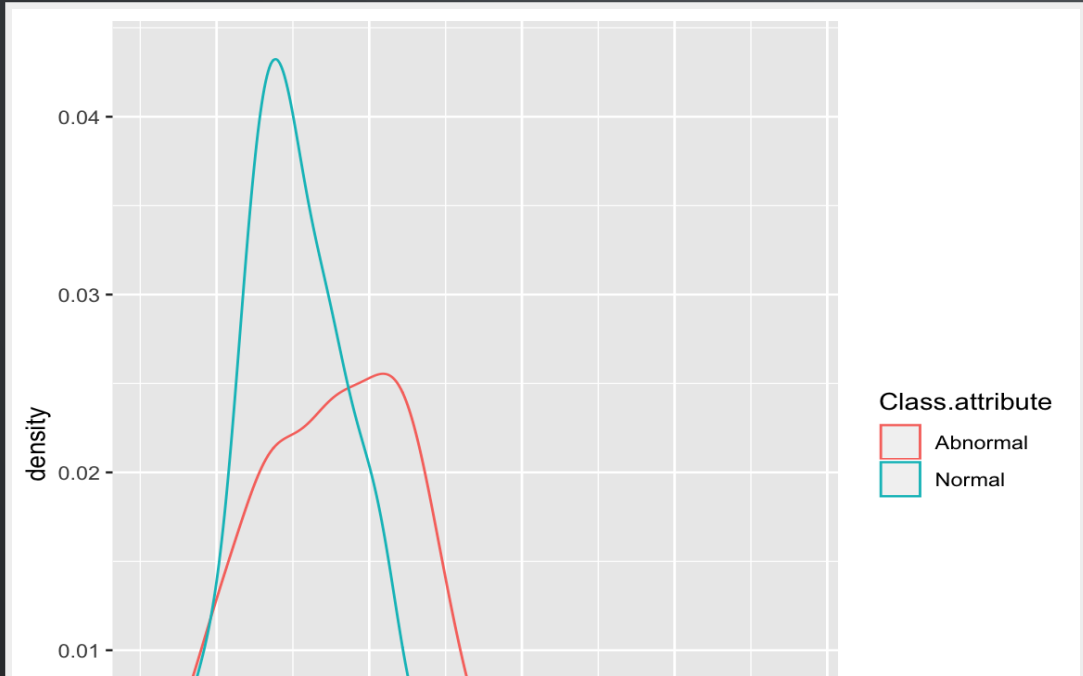
Work through the pipeline yourself to understand what each step does, just like last week


Confidence intervals of the mean
Cervical.tilt
12.5
12.0
11.5
11.0
Degree.spondylolisthesis
40
30
20
10
0
Direct.tilt
23
22
21
20
Lumbar.lordosis.angle
55
50
45
Pelvic.incidence
65
60
55
50
Pelvic.radius
124
120
116
Pelvic.slope
0.52
0.48
0.44
0.40
Pelvic.tilt
20
18
16
14
12
Sacral.slope
Sacrum.angle
-12
Scoliosis.slope
29
Thoracic.slope
14.0

GROUPED VISUALIZATION

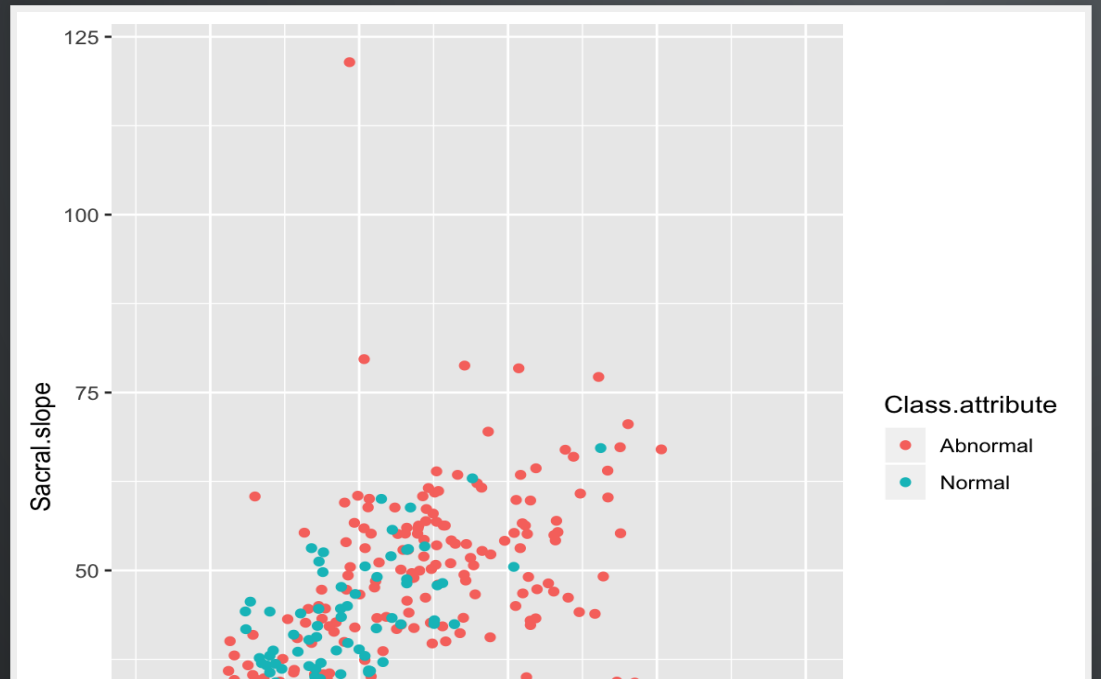
DENSITY PLOT

```
ggplot(data_spine, aes(x = Sacral.slope, group = Class.attribute,
                       color=Class.attribute))+
  geom_density()
```



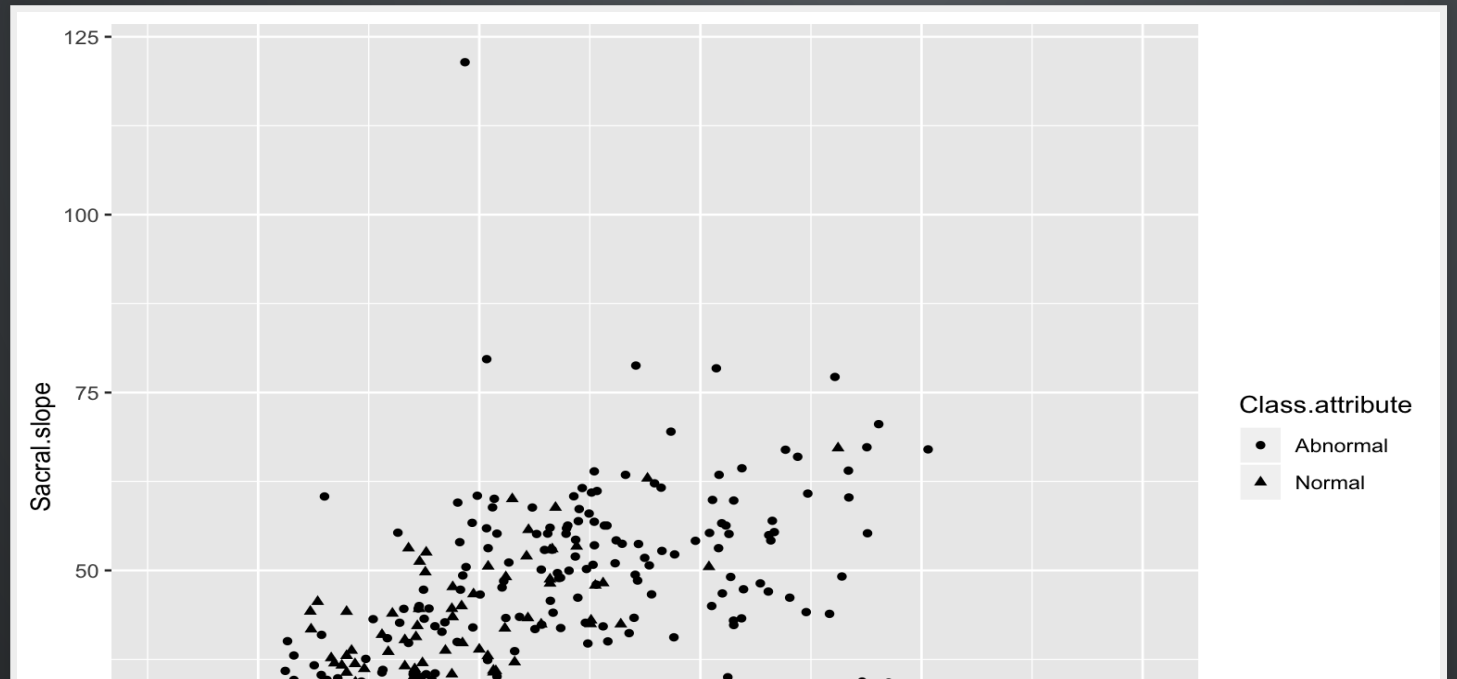
SCATTER PLOT

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope,
                       group = Class.attribute, color = Class.attribute))
  geom_point()
```



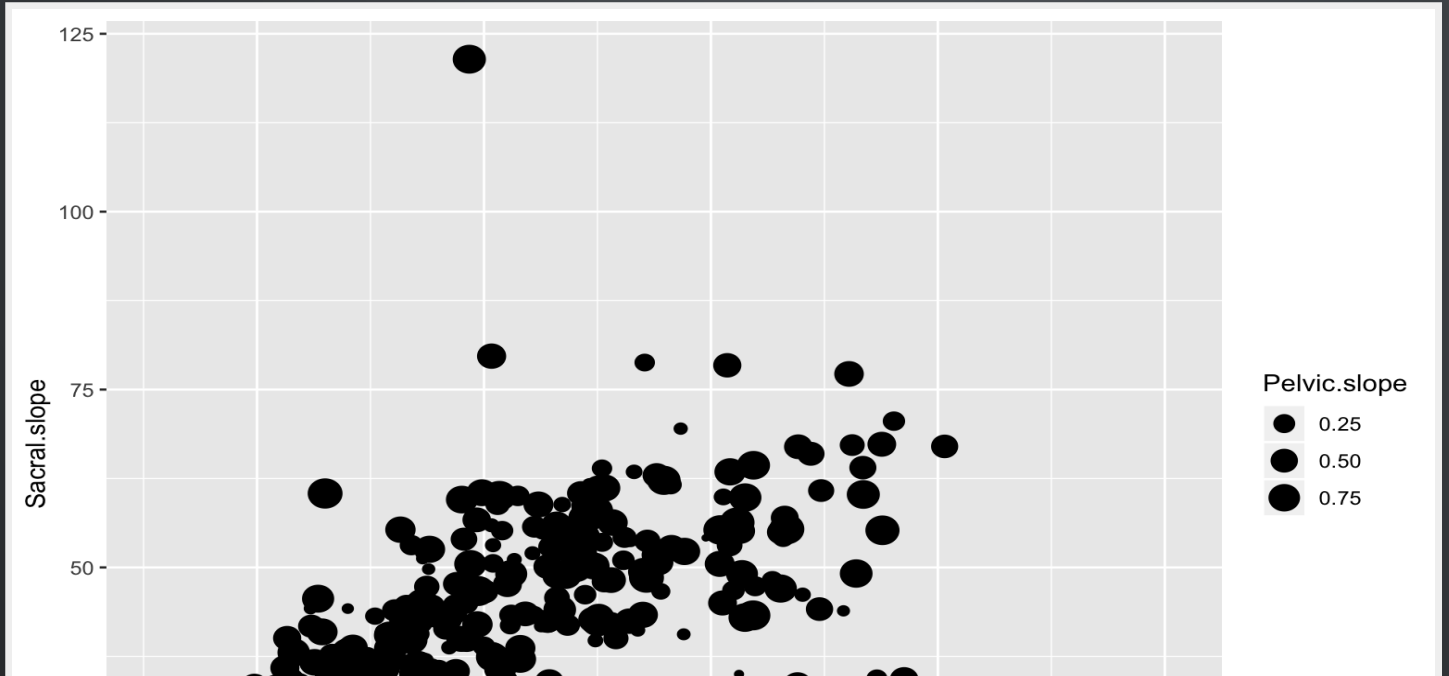
SCATTER PLOT (BLACK AND WHITE)

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope,
                       group = Class.attribute, shape = Class.attribute))
  geom_point()
```



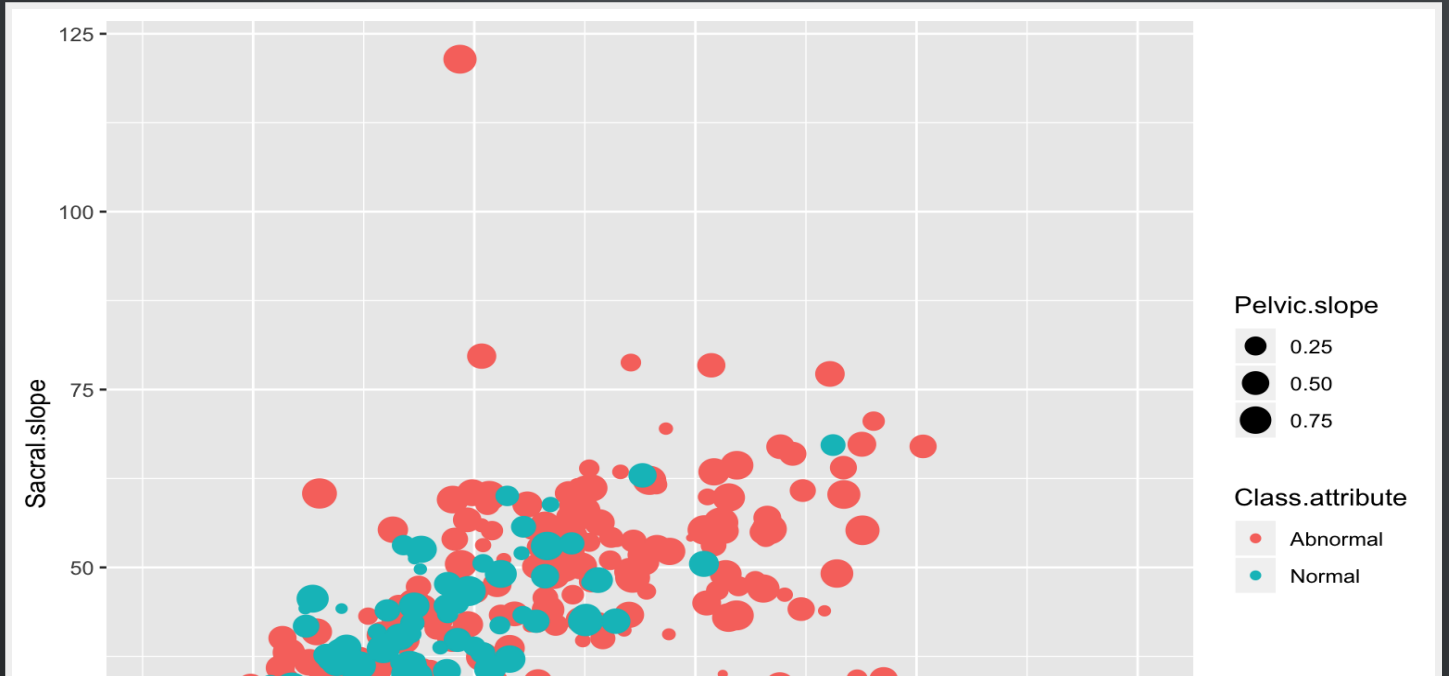
SCATTER PLOT WITH SIZE REPRESENTING THIRD VARIABLE

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope))+
  geom_point(aes(size = Pelvic.slope))
```



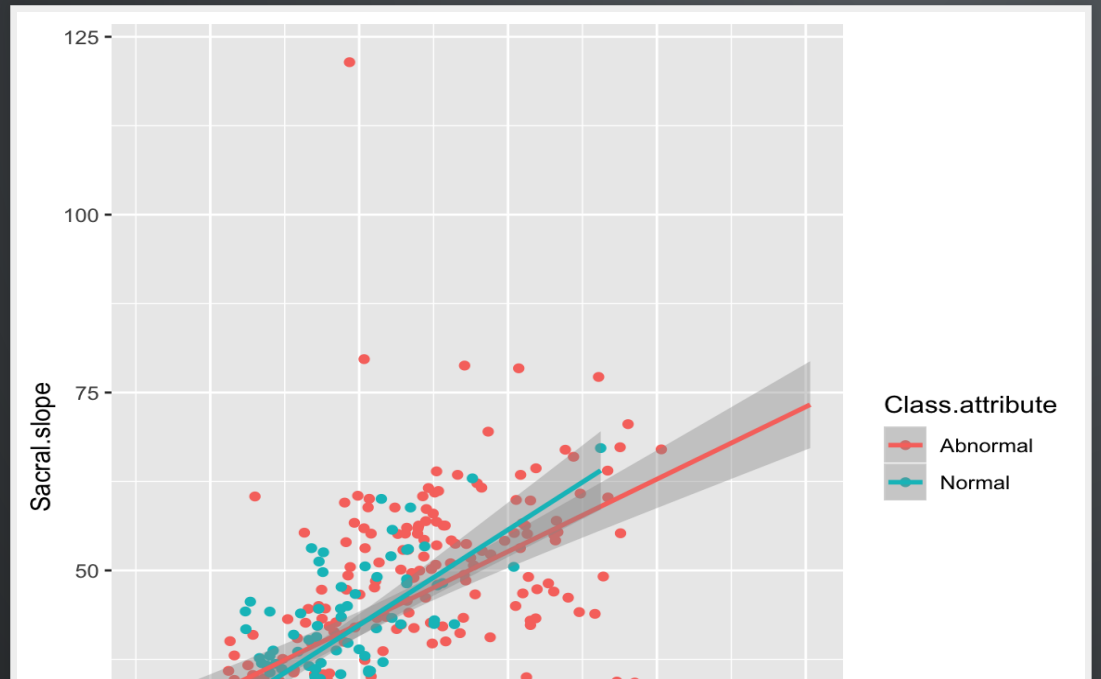
SCATTER PLOT COMBINATIONS

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope,
                       group = Class.attribute, color = Class.attribute))
  geom_point(aes(size = Pelvic.slope))
```



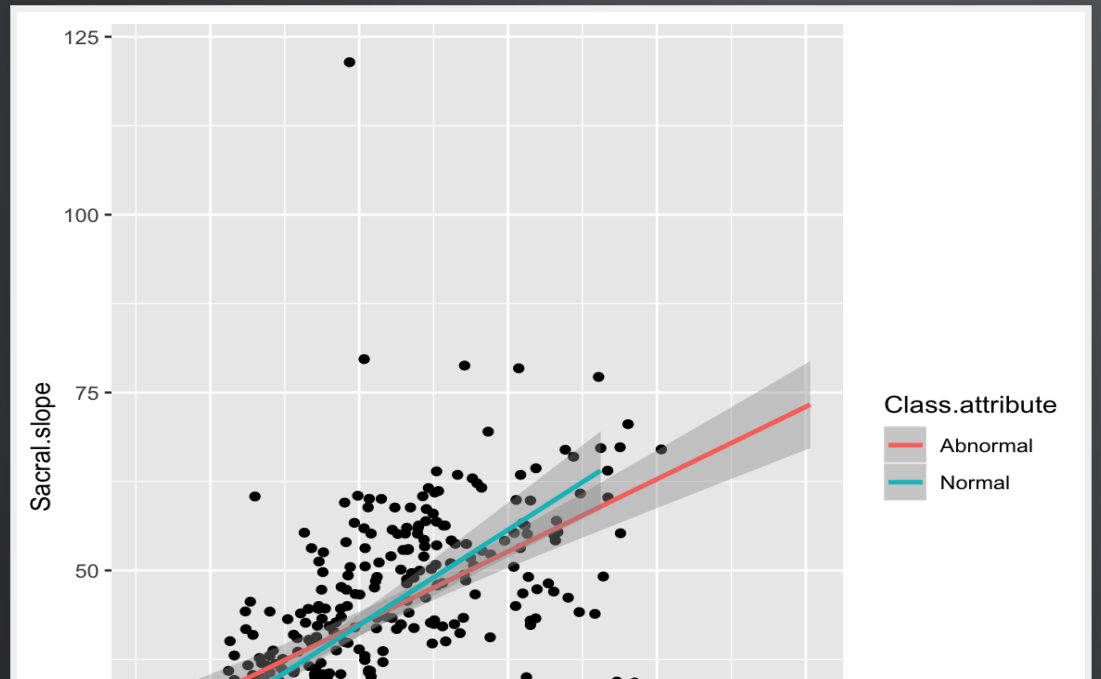
SCATTER PLOT WITH LINES

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope,
                       group = Class.attribute, color=Class.attribute))+
  geom_point()+
  geom_smooth(method='lm')
```



SCATTER PLOT WITH LINES

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope))+
  geom_point()+
  geom_smooth(aes(color = Class.attribute), method='lm')
```



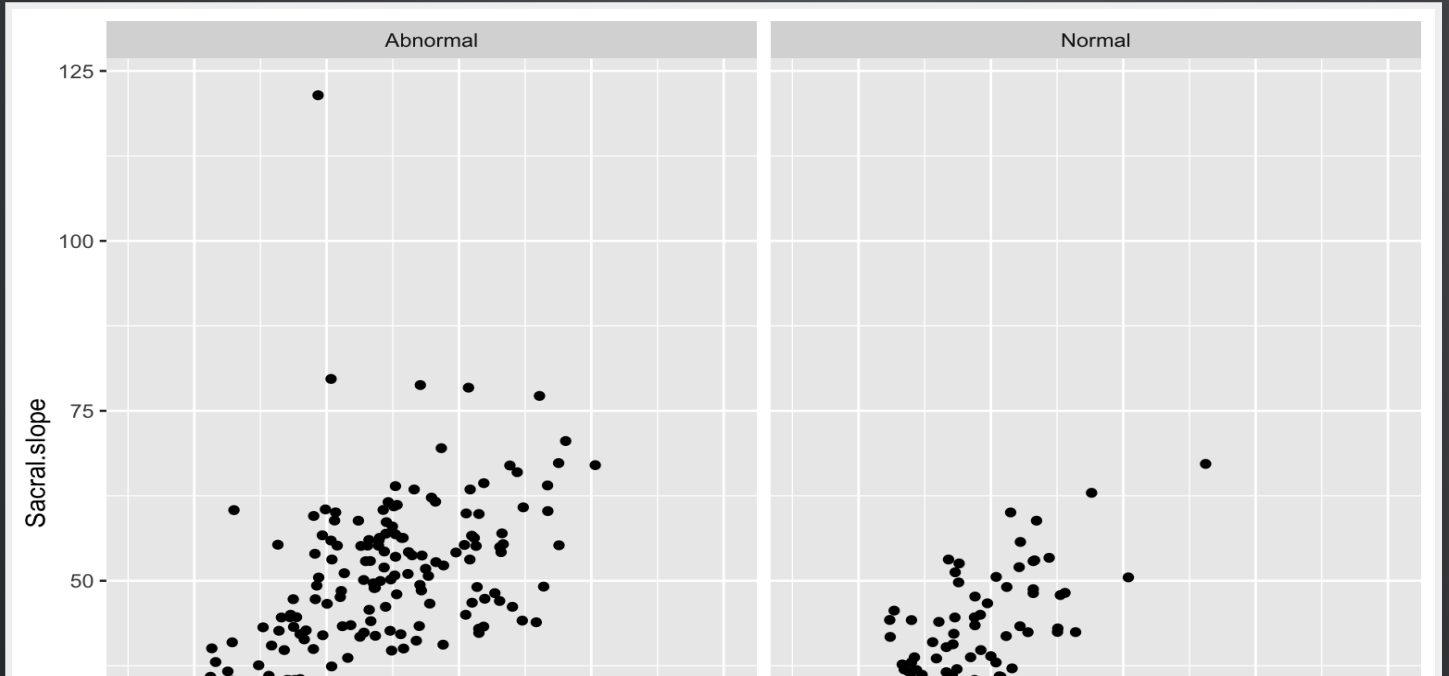
FACETTING

FACETTING

Facetted graphs are a panel of graphs, each of which corresponds to a particular subgroup of the data.

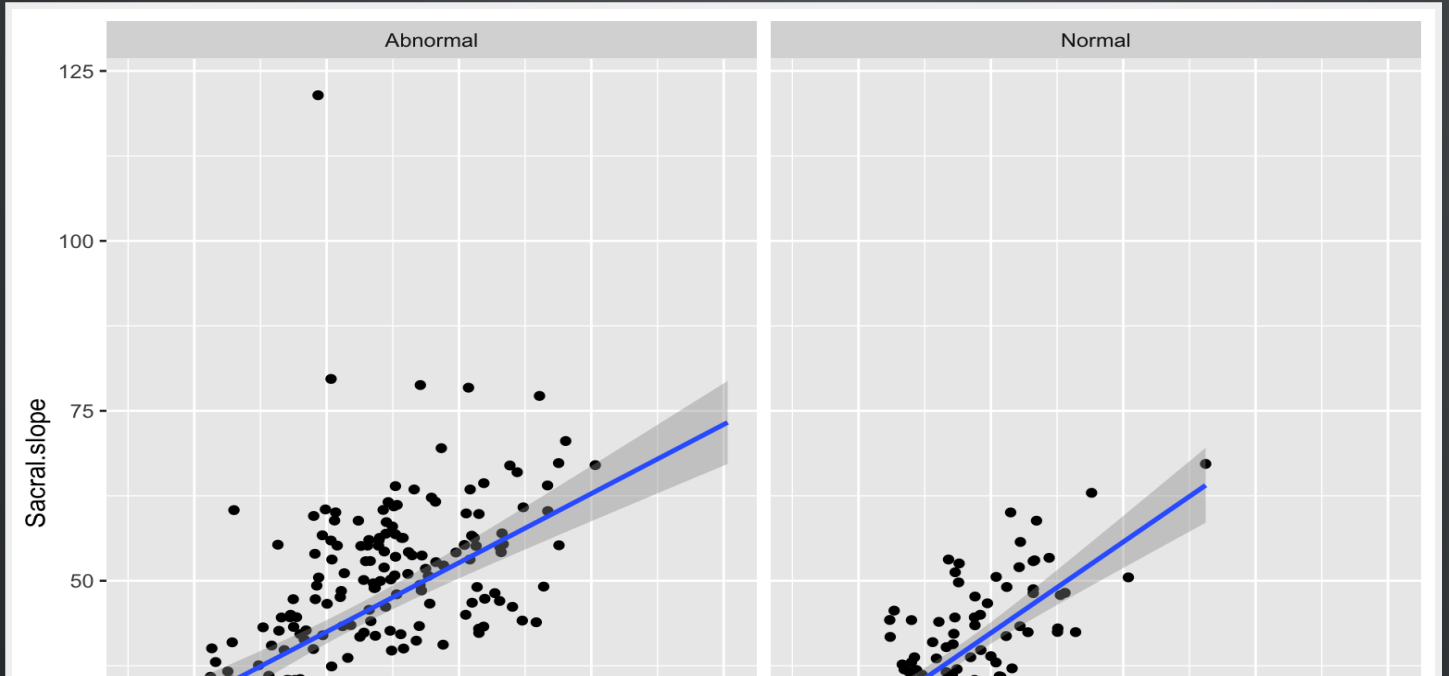
FACETTED SCATTER PLOT

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope))+
  geom_point()+
  facet_wrap( ~ Class.attribute, nrow=1)
```



FACETTED SCATTER PLOT WITH LINES

```
ggplot(data_spine, aes(x = Lumbar.lordosis.angle, y = Sacral.slope))+
  geom_point()+ geom_smooth(method='lm')+
  facet_wrap( ~ Class.attribute, nrow=1)
```



MANHATTAN PLOT

MANHATTAN PLOT

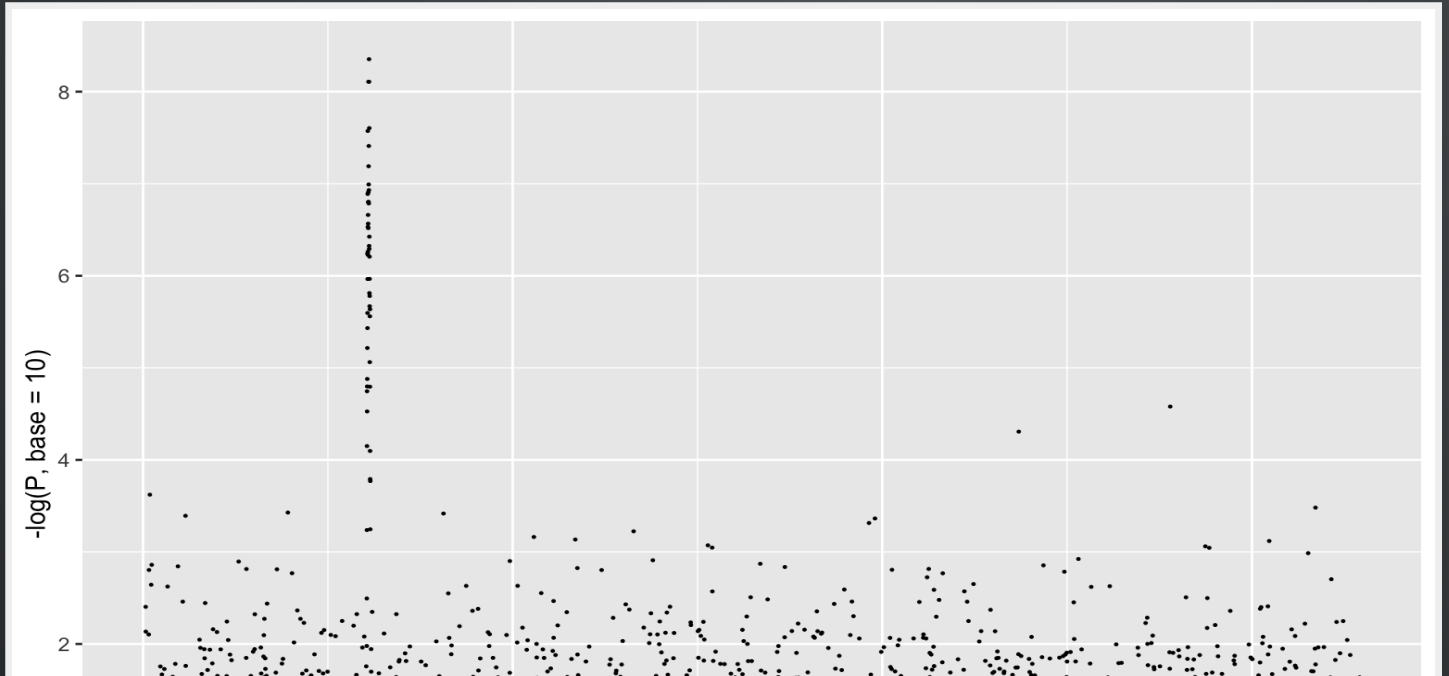
```r
library(qqman)
data(gwasResults)
head(gwasResults)
```

```
#      SNP CHR BP         P
#    1 rs1   1  1 0.9148060
#    2 rs2   1  2 0.9370754
#    3 rs3   1  3 0.2861395
#    4 rs4   1  4 0.8304476
#    5 rs5   1  5 0.6417455
#    6 rs6   1  6 0.5190959
```

```r
gwasResults = transform(gwasResults, x_position = 1:nrow(gwasResults))
```

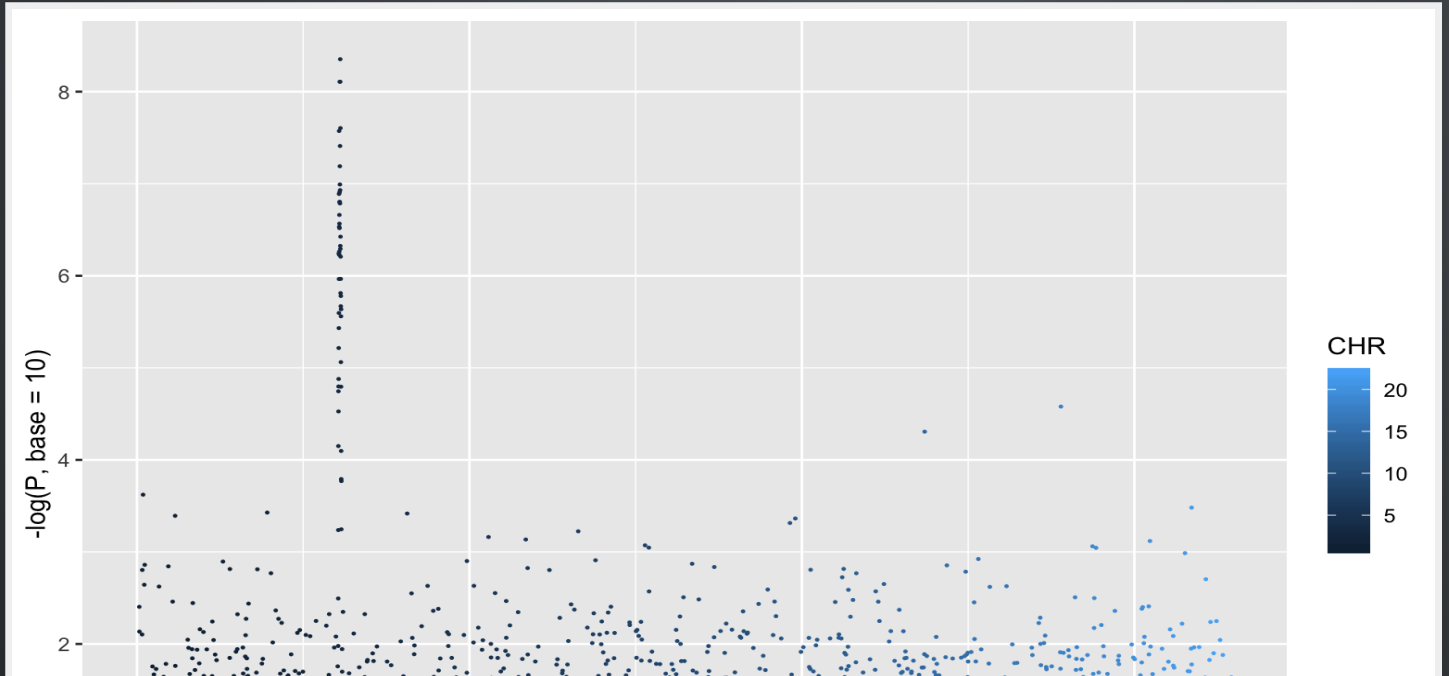
MANHATTAN PLOT

```
ggplot(gwasResults, aes(x = x_position, y = -log(P, base=10)))+
  geom_point(size=0.2)
```



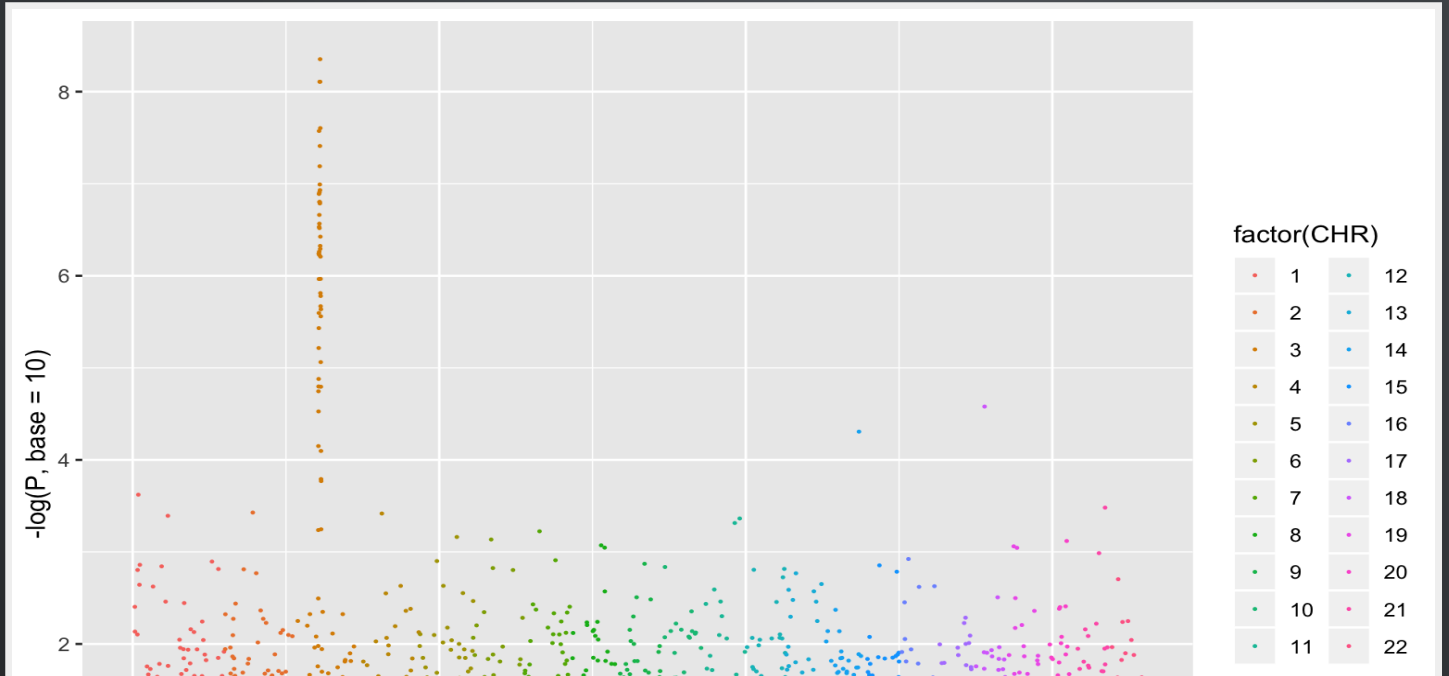
MANHATTAN PLOT

```
ggplot(gwasResults, aes(x = x_position, y = -log(P, base=10),
                        group=CHR, color=CHR))+
  geom_point(size=0.2)
```



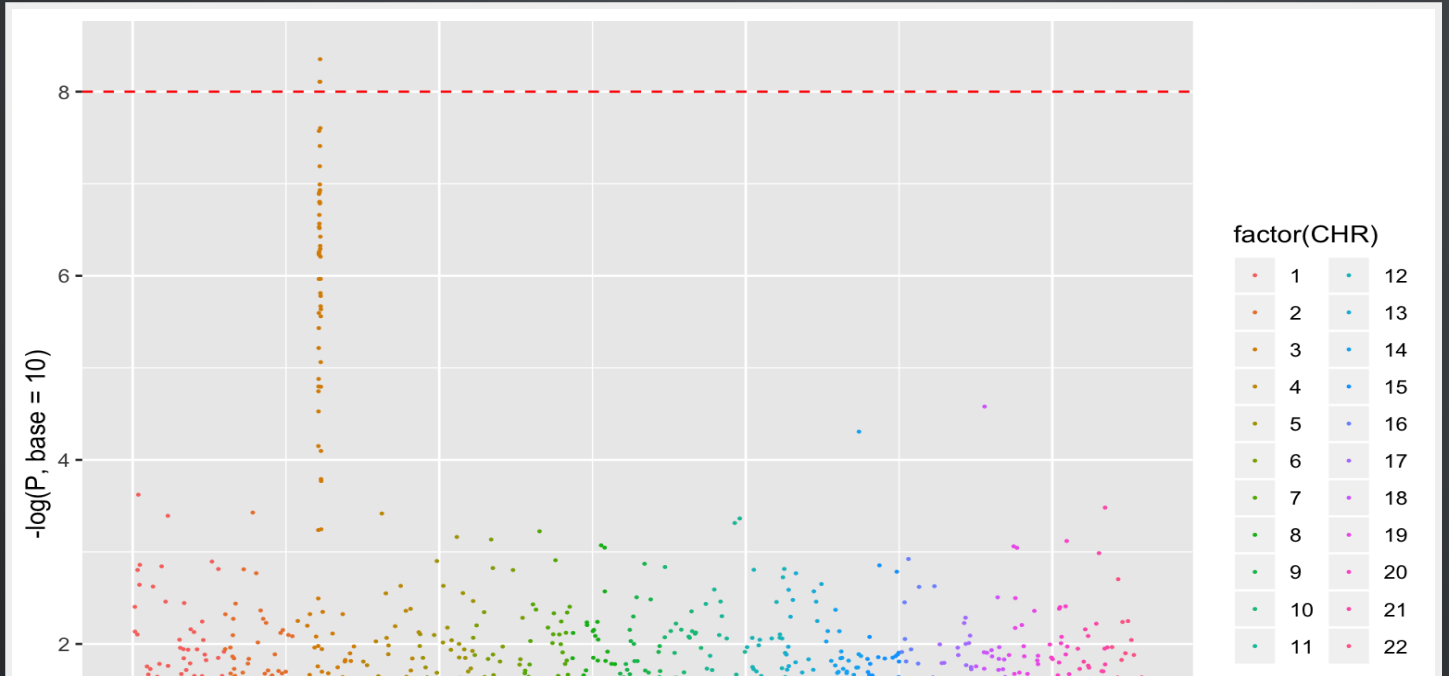
MANHATTAN PLOT

```
ggplot(gwasResults, aes(x = x_position, y = -log(P, base=10),
                        group=factor(CHR), color=factor(CHR)))+
  geom_point(size=0.2)
```



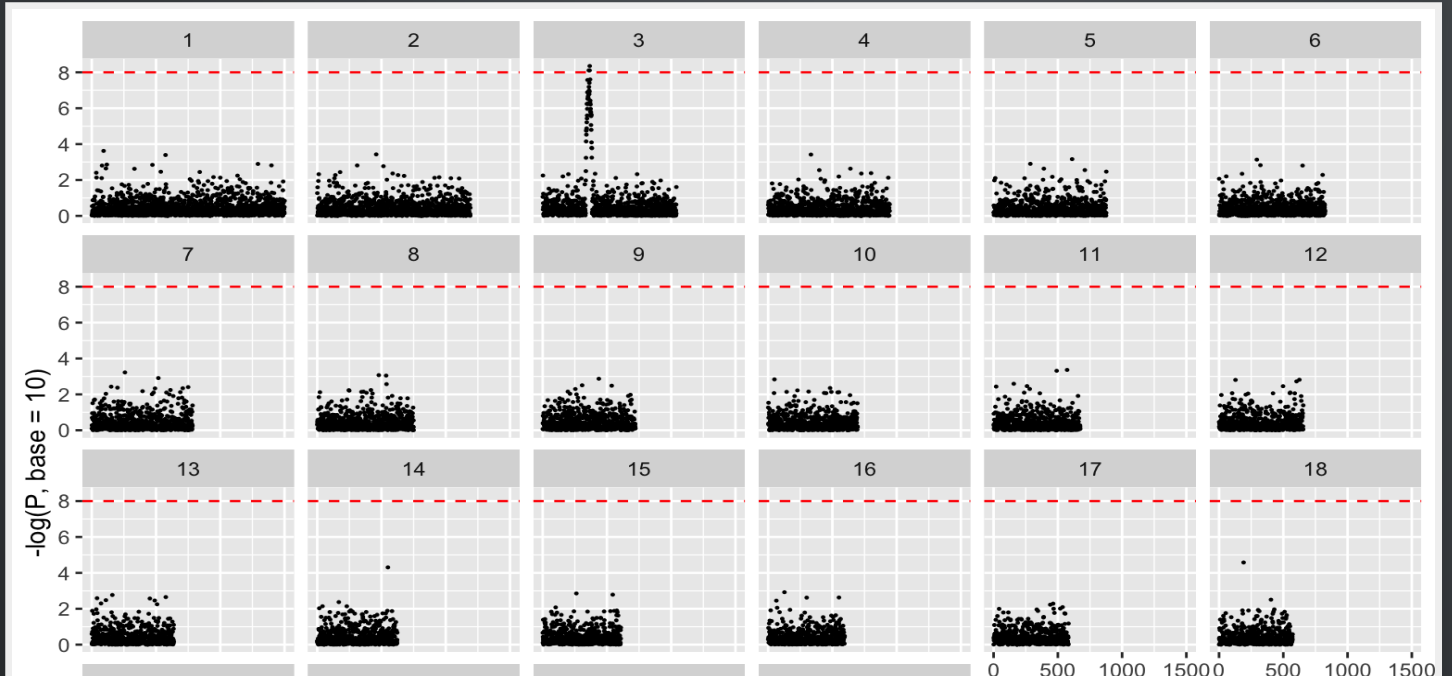
MANHATTAN PLOT

```
ggplot(gwasResults, aes(x = x_position, y = -log(P, base=10),
                        group=factor(CHR), color=factor(CHR)))+
  geom_point(size=0.2)+
  geom_hline(yintercept = 8, color='red', linetype=2)
```



MANHATTAN PLOT, EXPLODED

```
ggplot(gwasResults, aes(x = BP, y = -log(P, base=10)))+
  geom_point(size=0.2)+
  facet_wrap(~ CHR, nrow=4)+
  geom_hline(yintercept = 8, color='red', linetype=2)
```



MANHATTAN PLOT, EXPLODED