Exploring R Studio Cloud

Week 1

Introduction to Today

R Studio Cloud will allow you to use the full functionality of R & R Studio without having to download any software.

Additionally, we (me + your GSIs) have access to your scripts and can easily hop-on to troubleshoot with you.

The purpose of this **script** (what we call a file with R code in it) is to introduce you to some R lingo and show off some of the cool things R can do. The purpose *is not* for you to understand 100% of what is happening in R right now. We will slowly get into that as the lectures go on. Today I just wanted us to dive in and explore.

The type of **script** that this is, is called an *R Markdown* file. The extension is **.rmd**. This is different than a "normal" **.r** script. A **.r** script only lets you type and run code with comments hashtagged throughout. An **R Markdown** (.rmd) file allows you to create beautifully formatted research reports. It's great practice to start learning how to use R markown files now.

There are 3 main components to an R Markdown File:

- Headers
- Regular Text
- Code chunks

Headers

Headers are used for organizing your code. Think of them as headers in a book or a research paper. You have your header and your sub-headers. The biggest header is designated with one "#" and sub-headers add on "##"

This is a level 1 Header

This is a level 2 Header

This is a level 3 Header

Regular Text

Regular Text is what you are reading right now. Regular meaning non-code.

Code Chunks

A **code chunk** is what you see below. R will *only* run code that is inside of a code chunk.

For example:

2 + 2

The above is treated as regular text and will *not* be run as an equation.

```
#This is a code chunk.
#If you want to comment your code, you must precede it with a "#" or R will give you an error

#We have put some code below (super simple)
#R will run this and then show you the output below

2 + 2
```

```
## [1] 4
```

Plat around in R Studio Cloud/Hover to find the keyboard shortcuts.

For me (windows), "Ctrl" + "Enter" will run a line of code and "Shift" + "Ctrl" + "Enter" will run the entire code chunk.

```
#Try running just one line of code
a <- 5

#Try running the whole chunk
b <- 10
a  +  b
```

```
## [1] 15
```

Graphs

Now I want to show you some cool graphs R can make!

First, we have to load our "libraries". A **library** in R is an open-source package created by a very kind individual that contains functions (short cuts) to get things done in R.

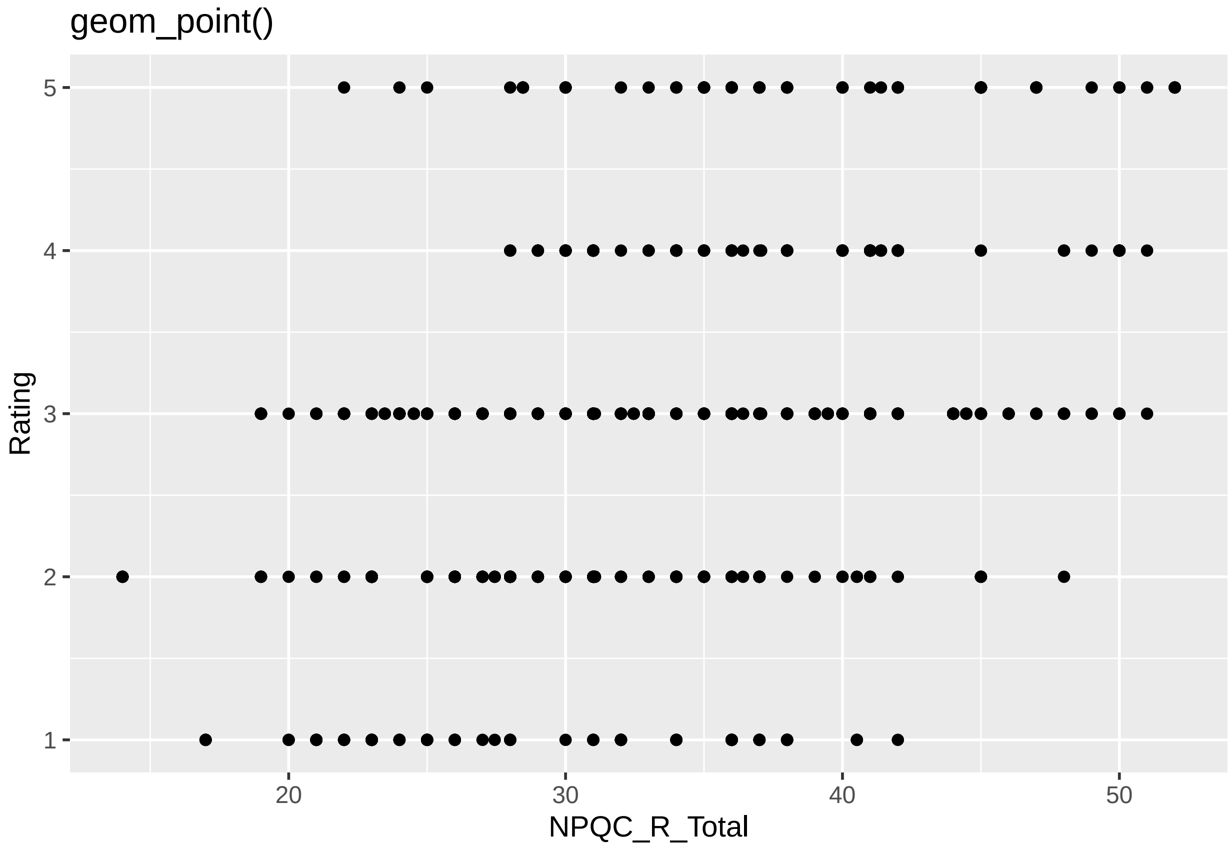
```
library(ggplot2)
library(reshape)
library(plyr)
```

Facebook Data on Narcissism

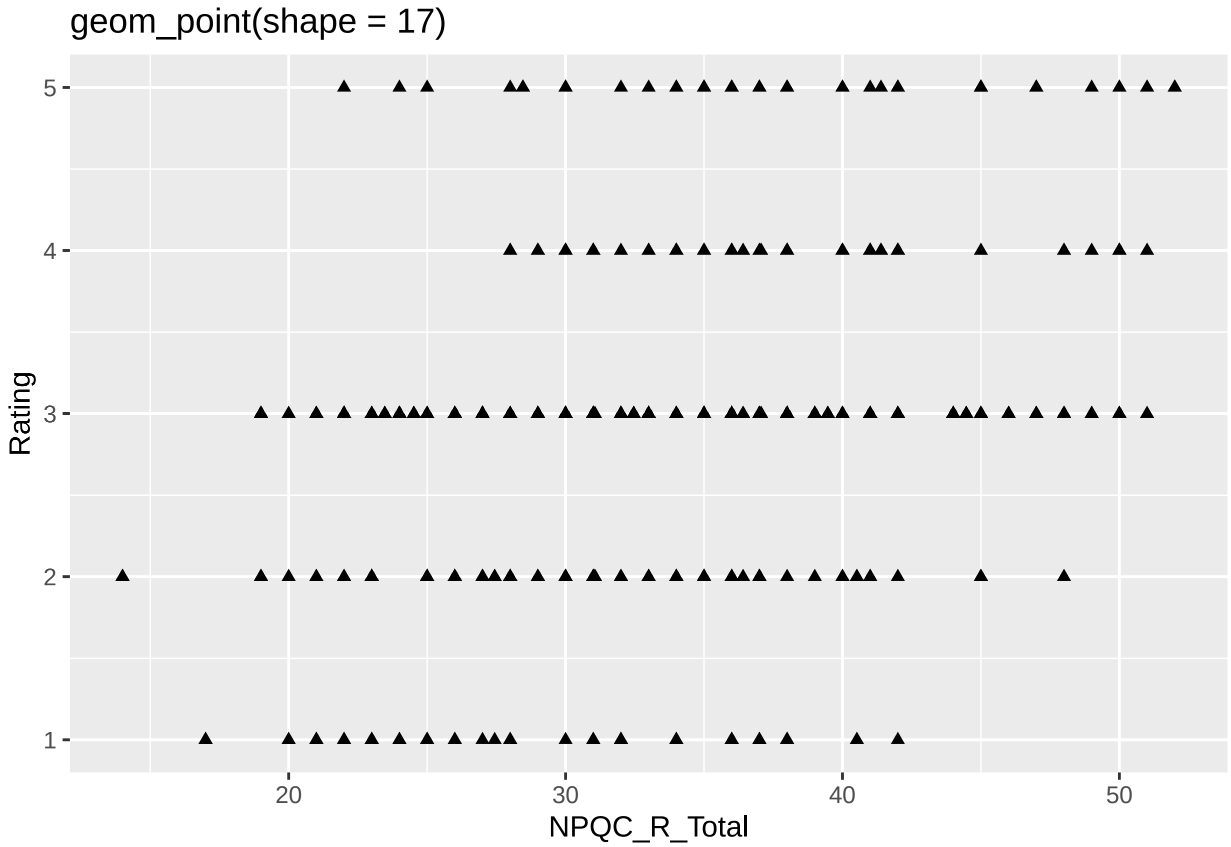
Next We will load in some data! I have uploaded the data files for you into the cloud, so you can just run the below code.

```
facebookData <- read.delim("FacebookNarcissism.dat",  header = TRUE)

#creating our basic plot called "graph"
graph <- ggplot(facebookData, aes(NPQC_R_Total, Rating))
```

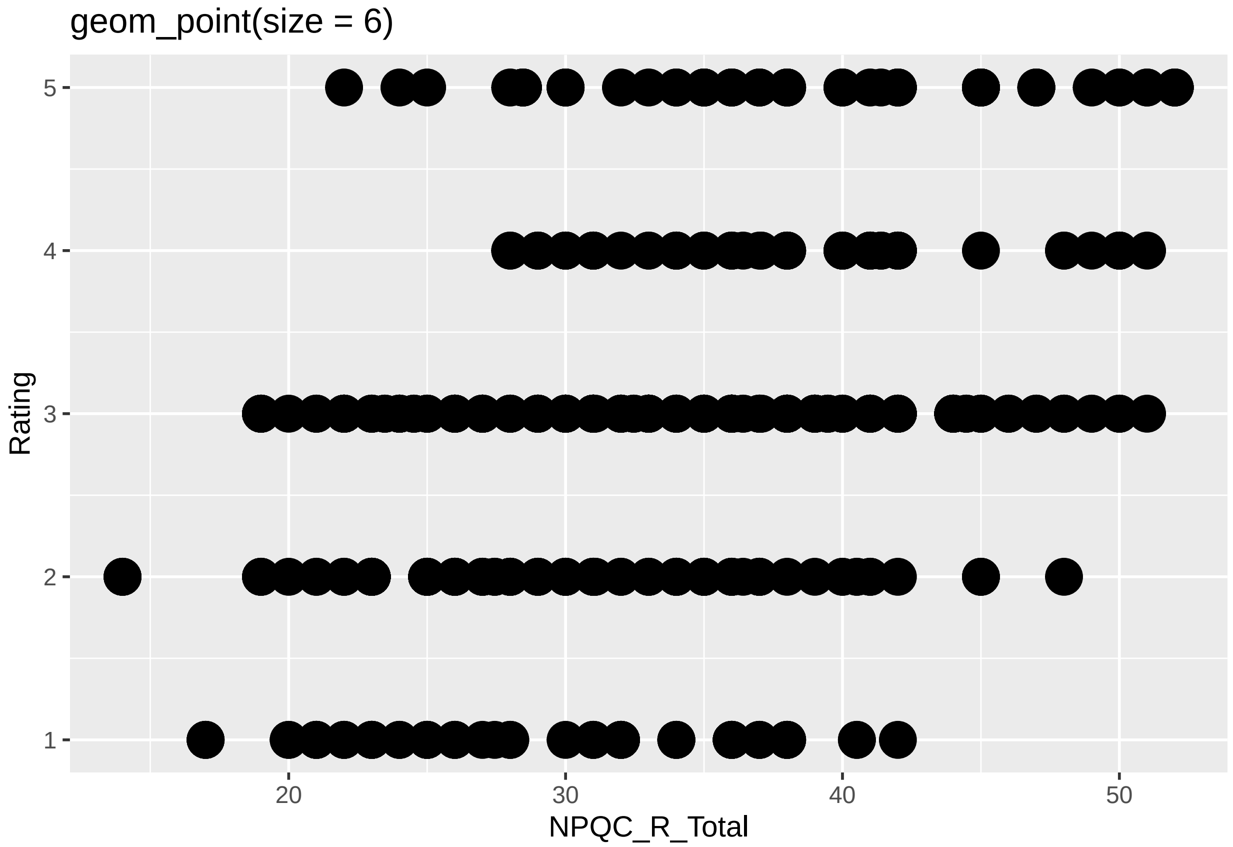
```
#graph with points
graph + geom_point() + labs(title = "geom_point()")
```



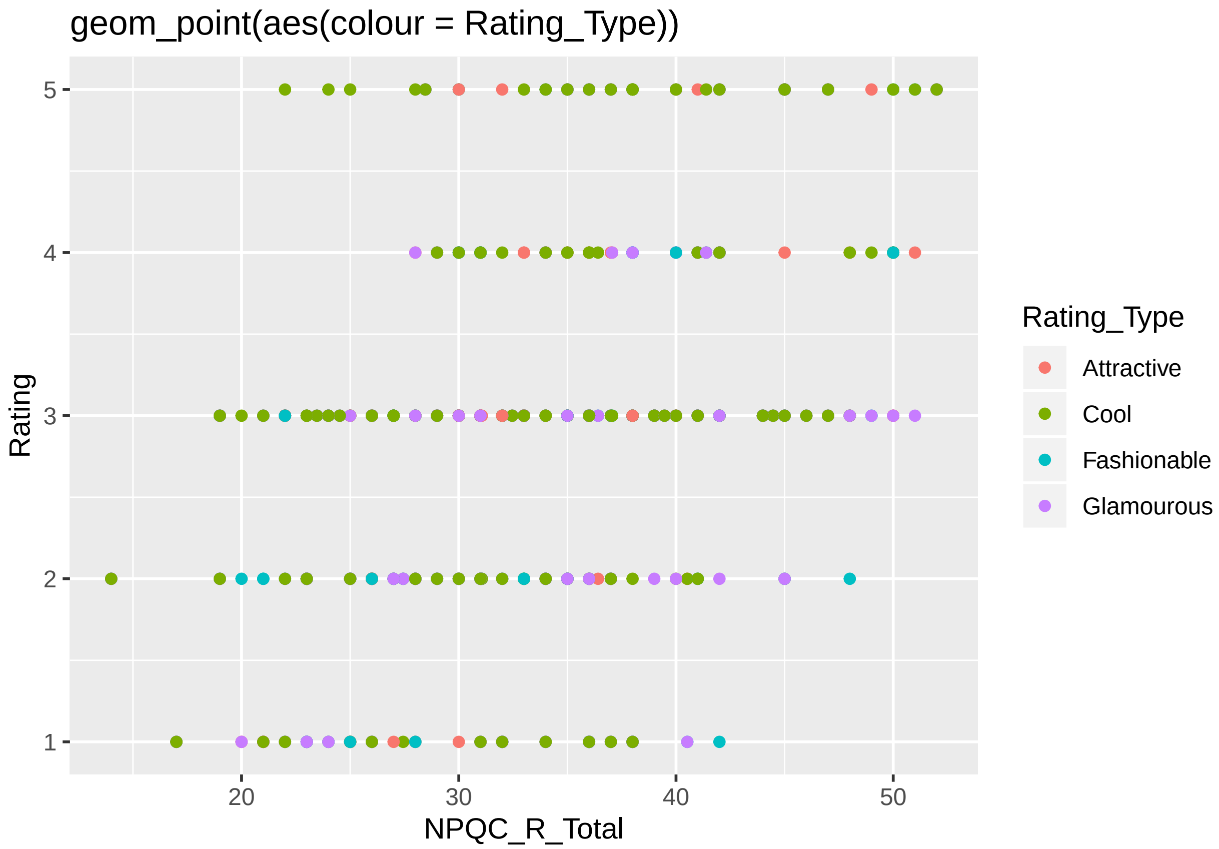
```
#graph with triangles
graph + geom_point(shape = 17) + labs(title = "geom_point(shape = 17)")
```



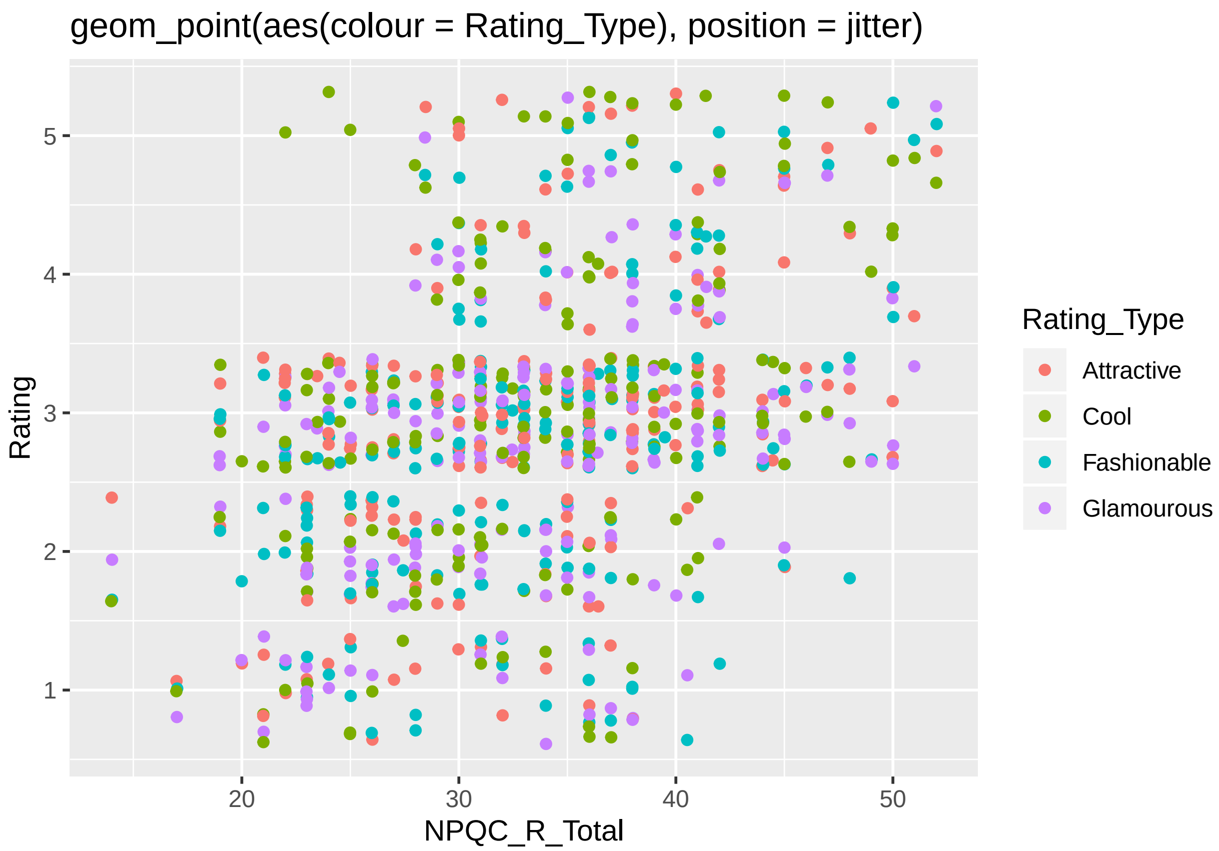
```
#changing the size
graph + geom_point(size = 6) + labs(title = "geom_point(size = 6)")
```



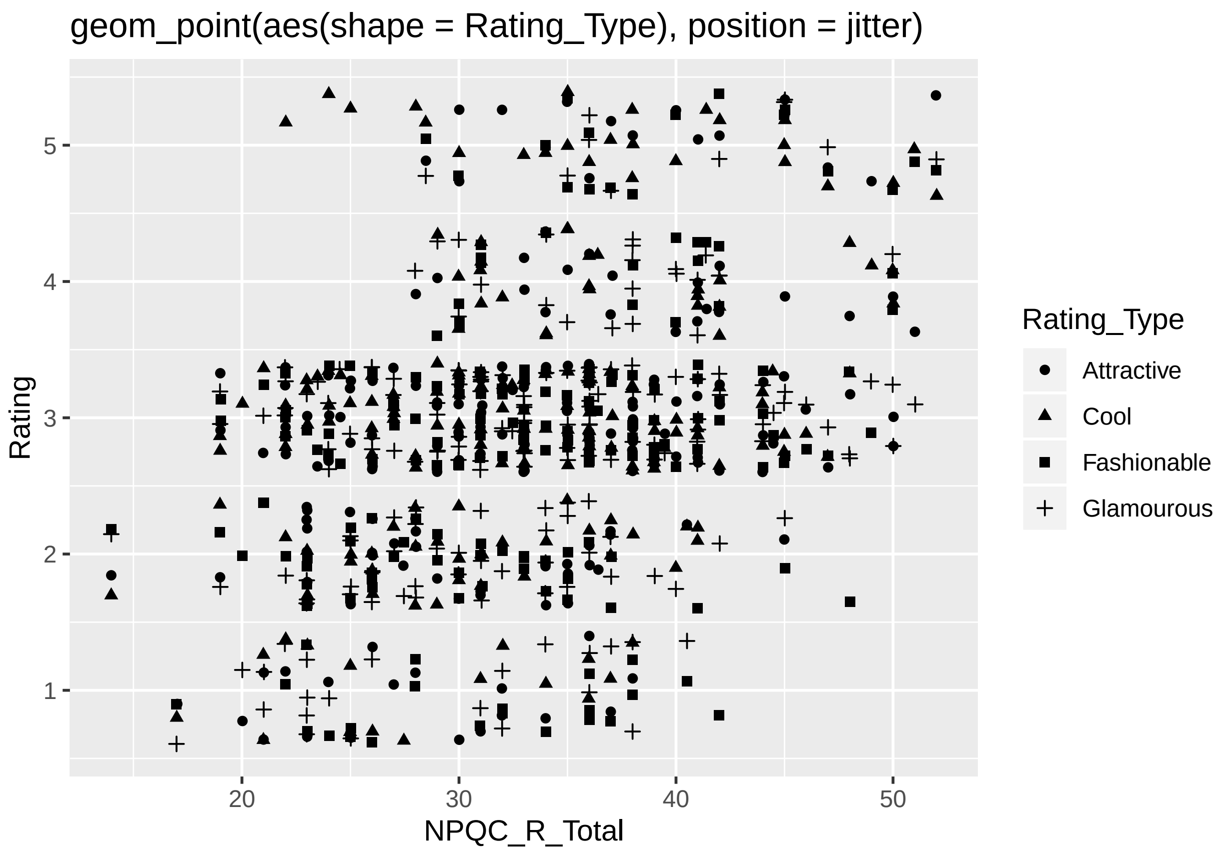
```
# coloring
graph + geom_point(aes(colour = Rating_Type)) + labs(title = "geom_point(aes(colour = Rating_Type))")
```



```
# color + jittering
graph + geom_point(aes(colour = Rating_Type), position = "jitter") + labs(title = "geom_point(aes(colou
```



```
# shape + jitter
graph + geom_point(aes(shape = Rating_Type), position = "jitter") + labs(title = "geom_point(aes(shape
```

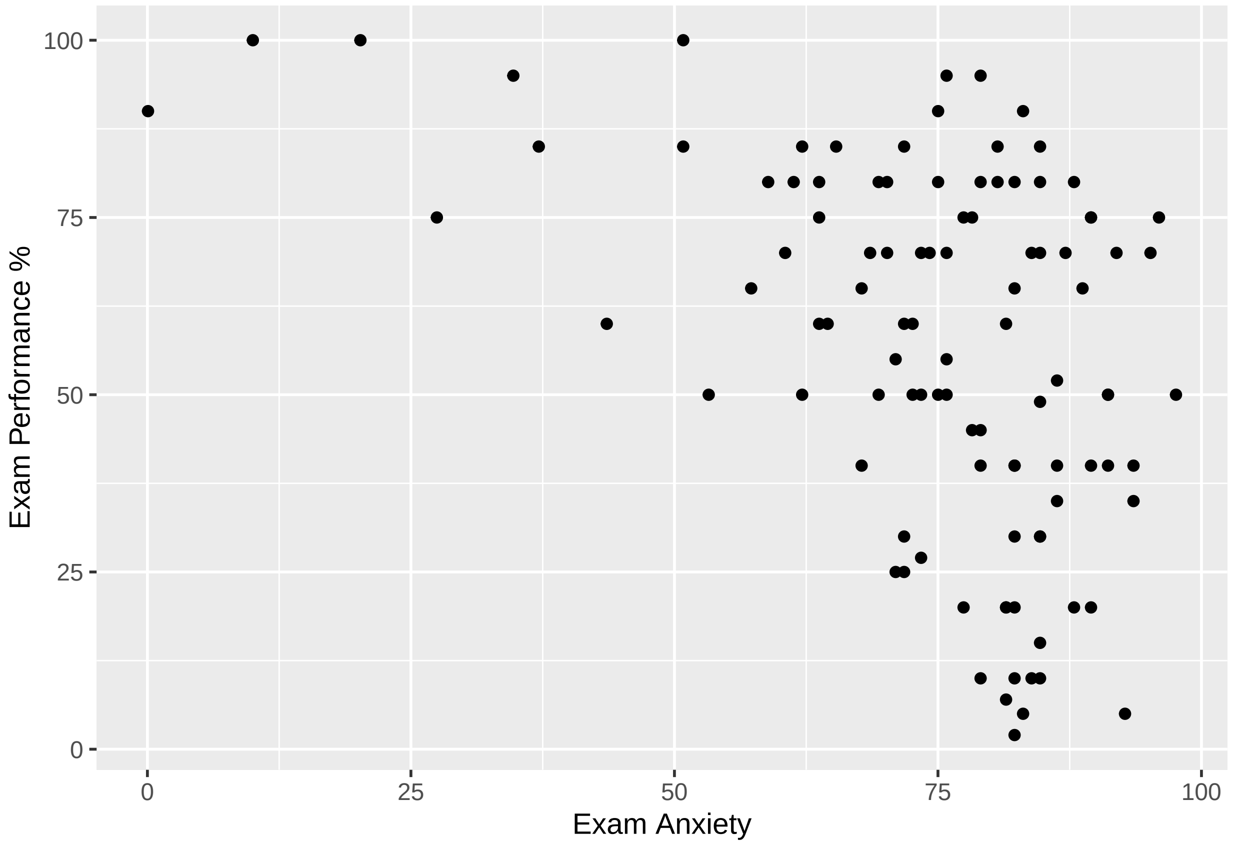


Scatterplots

```
#reading in data
examData <- read.delim("Exam Anxiety.dat",  header = TRUE)

#seeing variable names
names(examData)
```

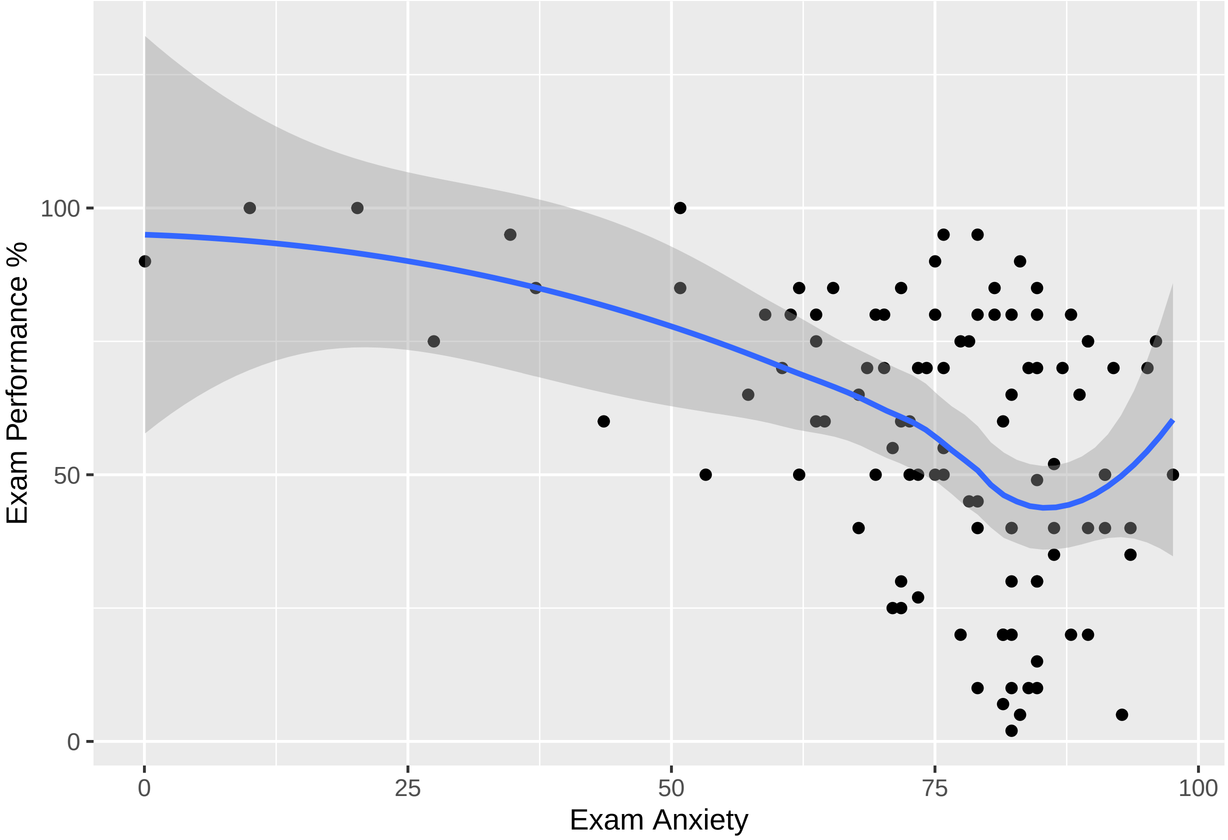
```
## [1] "Code"     "Revise"   "Exam"     "Anxiety" "Gender"
```

```r
#Simple scatterplot
scatter <- ggplot(examData, aes(Anxiety, Exam))
scatter + geom_point() +
  labs(x = "Exam Anxiety", y = "Exam Performance %")
```

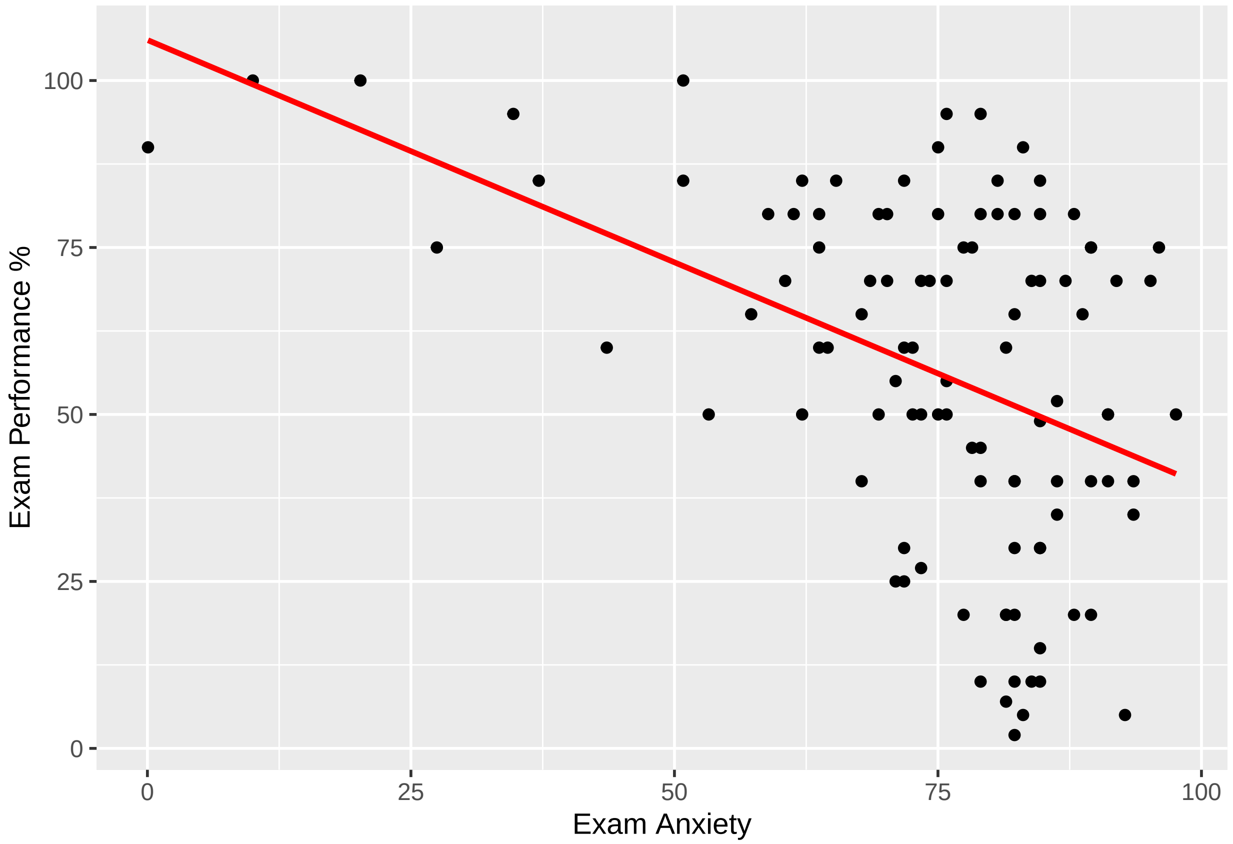


```
#Simple scatter with smooth
scatter <- ggplot(examData, aes(Anxiety, Exam))
scatter + geom_point() + geom_smooth() + labs(x = "Exam Anxiety", y = "Exam Performance %")
```

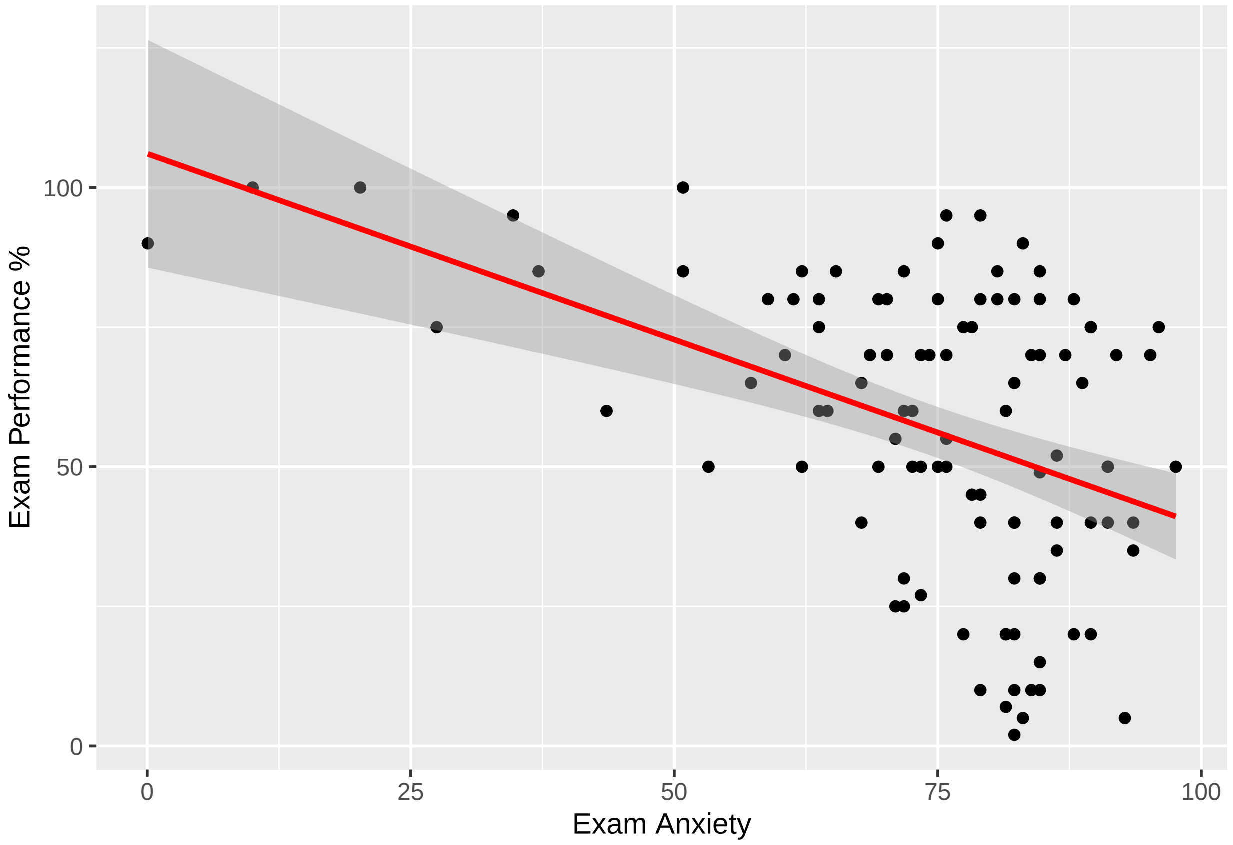
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



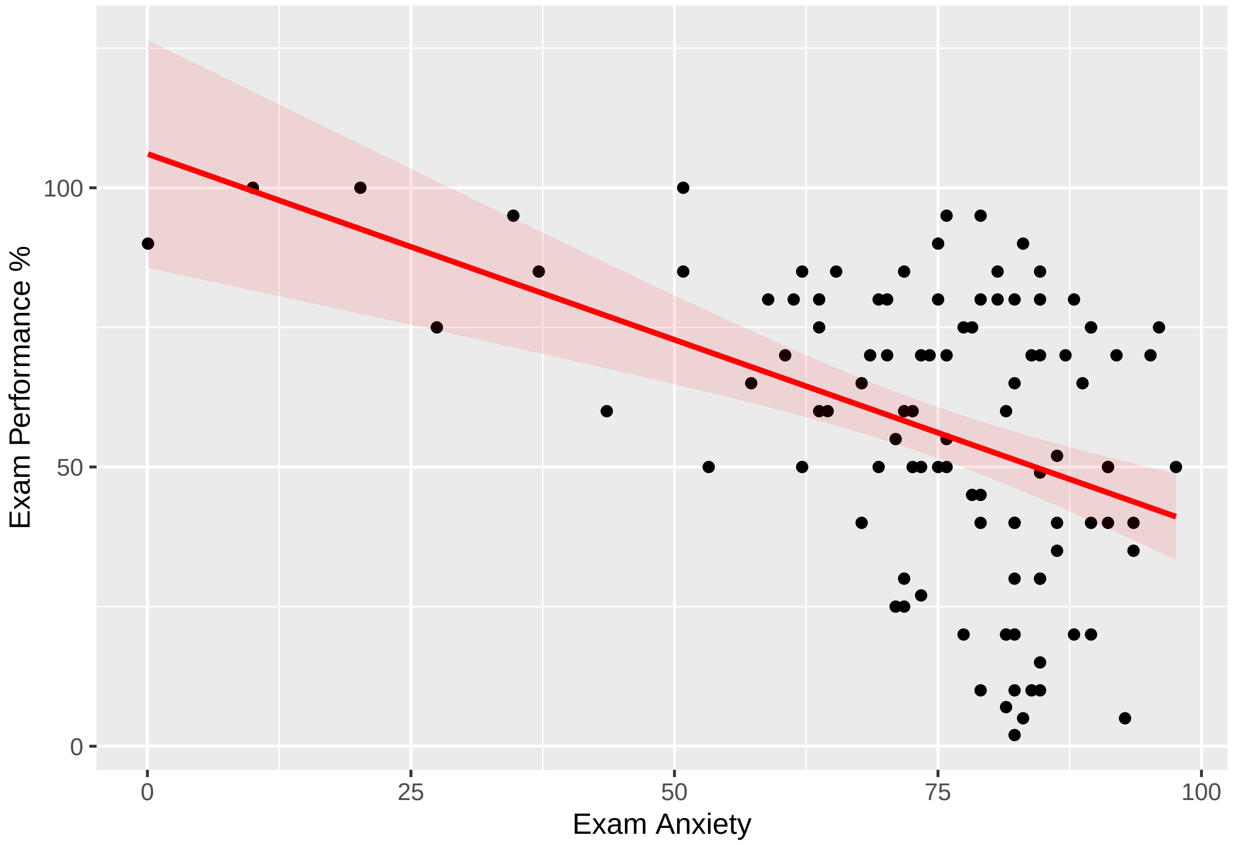
```r
#Simple scatter with regression line
scatter <- ggplot(examData, aes(Anxiety, Exam))
scatter + geom_point() + geom_smooth(method = "lm", colour = "Red", se = F) +
  labs(x = "Exam Anxiety", y = "Exam Performance %")
```



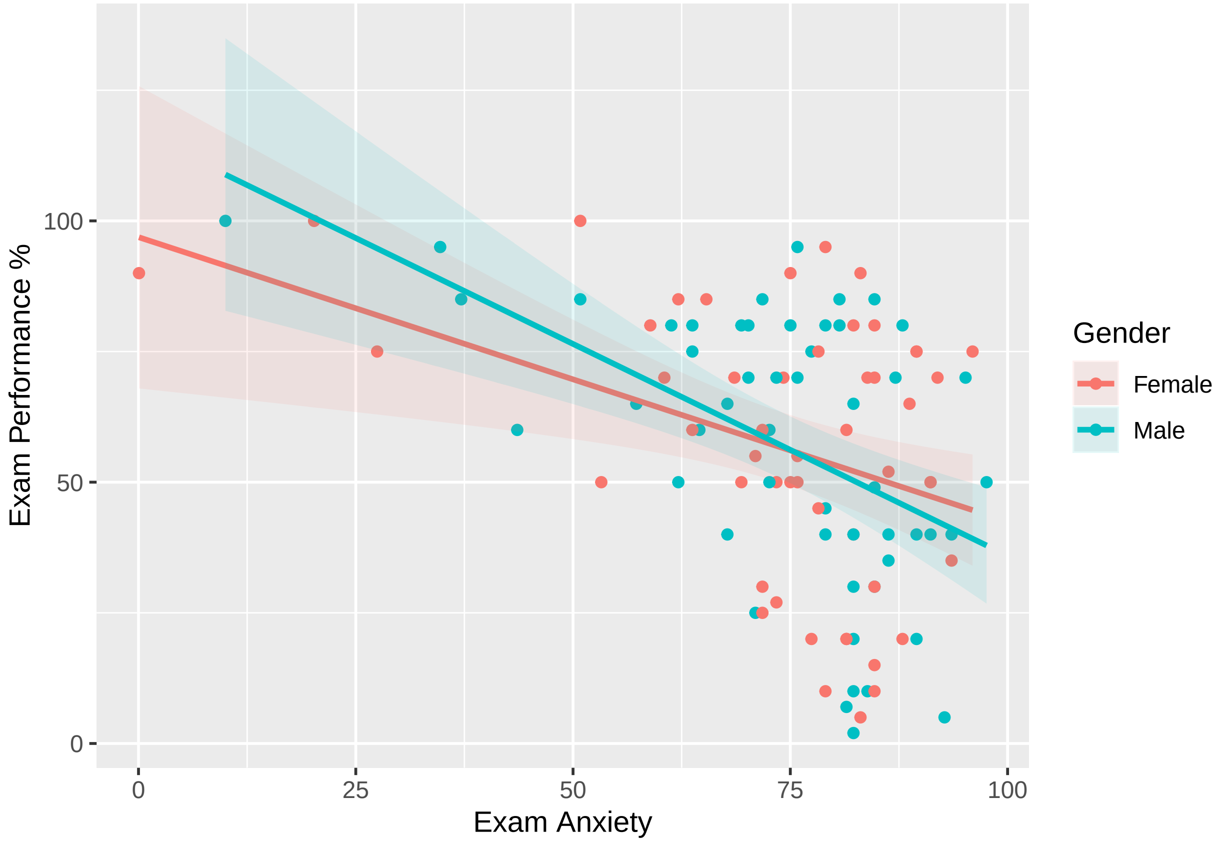
```
#Simple scatter with regression line + CI
scatter <- ggplot(examData, aes(Anxiety, Exam))
scatter + geom_point() + geom_smooth(method = "lm", colour = "Red")+
  labs(x = "Exam Anxiety", y = "Exam Performance %")
```



```
#Simple scatter with regression line + coloured CI
scatter <- ggplot(examData, aes(Anxiety, Exam))
scatter + geom_point() + geom_smooth(method = "lm", colour = "Red", alpha = 0.1, fill = "Red") +
  labs(x = "Exam Anxiety", y = "Exam Performance %")
```

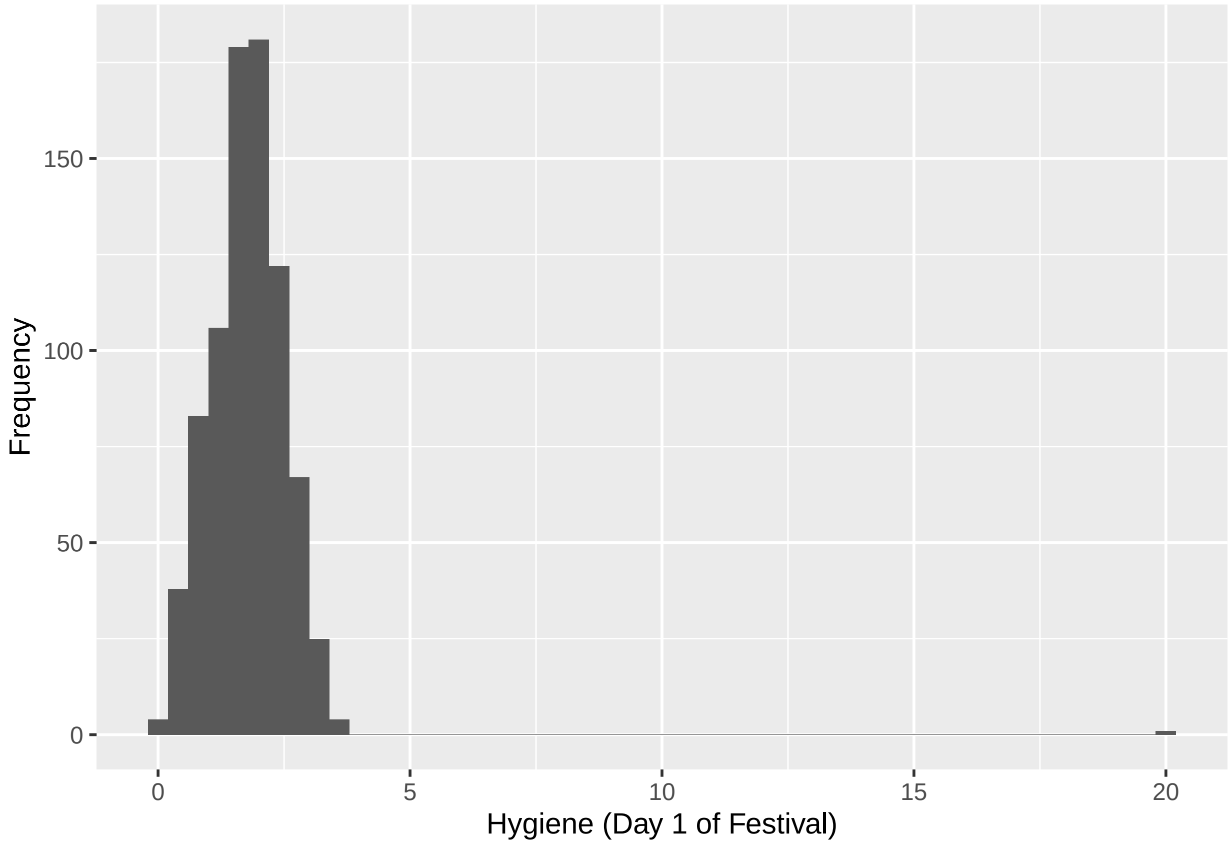


```
#Grouped scatter with regression line + CI
scatter <- ggplot(examData, aes(Anxiety, Exam, colour = Gender))
scatter + geom_point() + geom_smooth(method = "lm", aes(fill = Gender), alpha = 0.1) +
  labs(x = "Exam Anxiety",  y = "Exam Performance %", colour = "Gender")
```



Histograms

```
#Load the data file
festivalData <- read.delim("DownloadFestival.dat",  header = TRUE)

#creating our basic plot
festivalHistogram <- ggplot(festivalData, aes(day1)) + theme(legend.position='none')
festivalHistogram + geom_histogram(binwidth = 0.4) +
  labs(x = "Hygiene (Day 1 of Festival)", y = "Frequency")
```



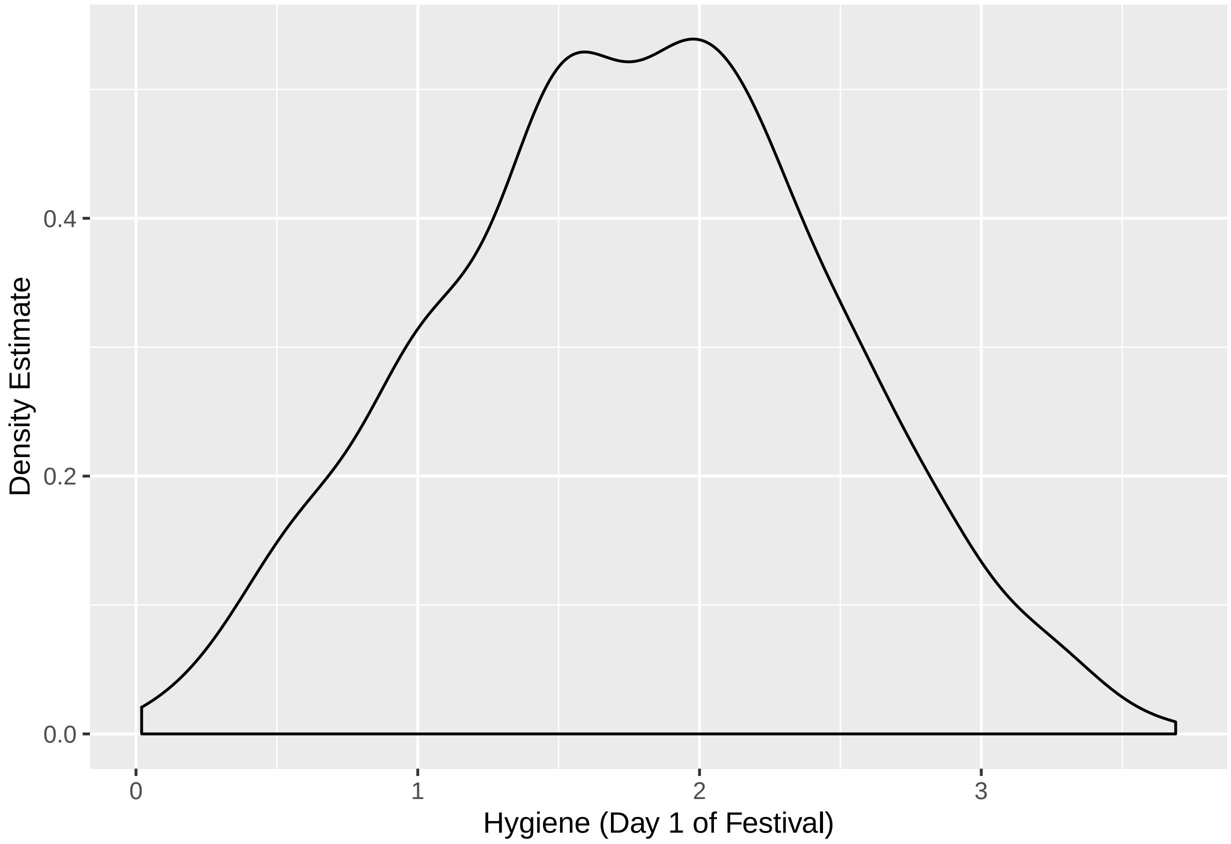
```
#Locate outlier
festivalData<-festivalData[order(festivalData$day1),]
head(festivalData) #first 6
```

```
##     ticknumb gender day1 day2 day3
## 595     4107 Female 0.02   NA   NA
## 366     3540 Female 0.05   NA   NA
## 41      2662 Female 0.11   NA   NA
## 177     3030   Male 0.11 0.29   NA
## 354     3511 Female 0.23 0.14   NA
## 569     4011 Female 0.23 0.84   NA
```

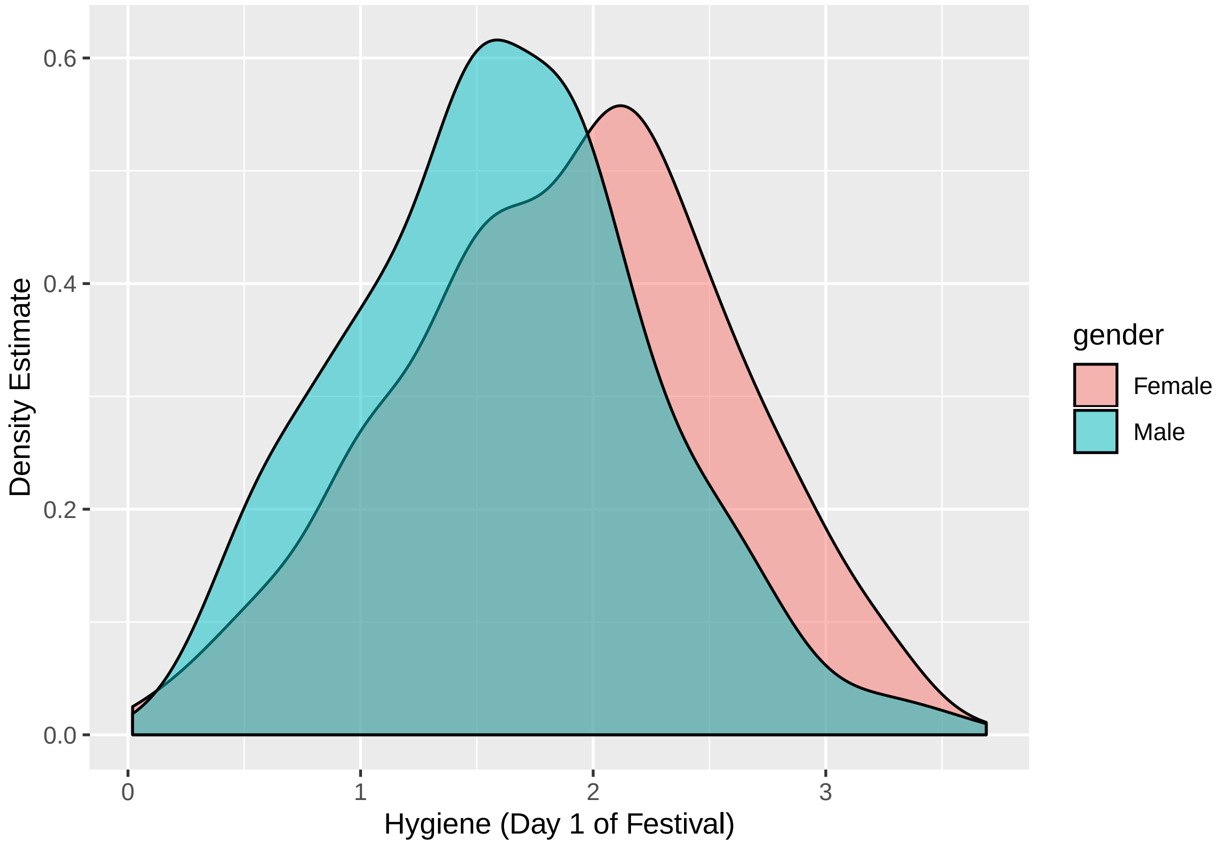
```
tail(festivalData) #last 6
```

```
##     ticknumb gender  day1 day2 day3
## 774     4564 Female  3.38 3.44 3.41
## 300     3371 Female  3.41   NA   NA
## 657     4264   Male  3.44   NA   NA
## 303     3374   Male  3.58 3.35   NA
## 574     4016 Female  3.69   NA   NA
## 611     4158 Female 20.02 2.44   NA
```

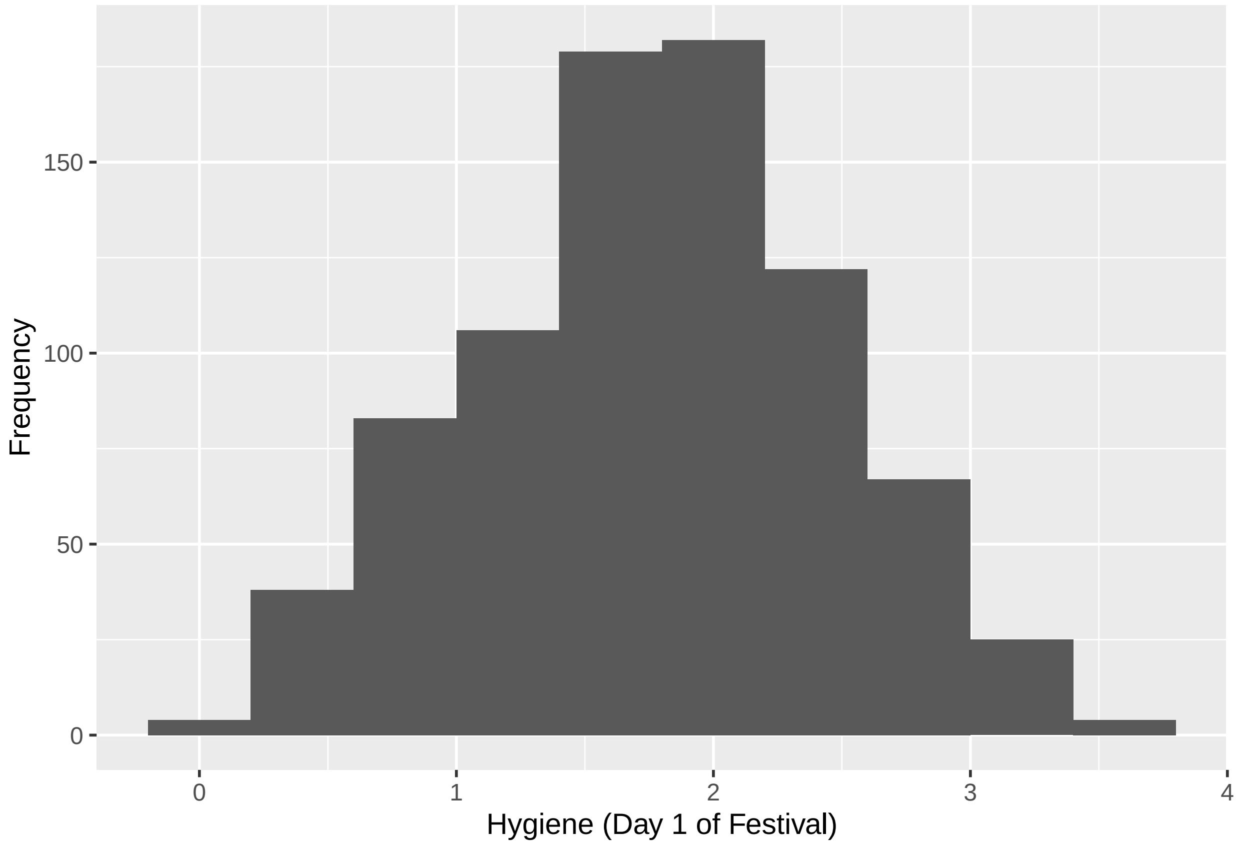
```
#Density without outlier
festivalData2 = read.delim("DownloadFestival(No Outlier).dat",  header = TRUE)
festivalDensity <- ggplot(festivalData2, aes(day1))
festivalDensity + geom_density() +
  labs(x = "Hygiene (Day 1 of Festival)", y = "Density Estimate")
```



```r
festivalDensity + geom_density(aes(fill = gender), alpha = 0.5) +
  labs(x = "Hygiene (Day 1 of Festival)", y = "Density Estimate")
```

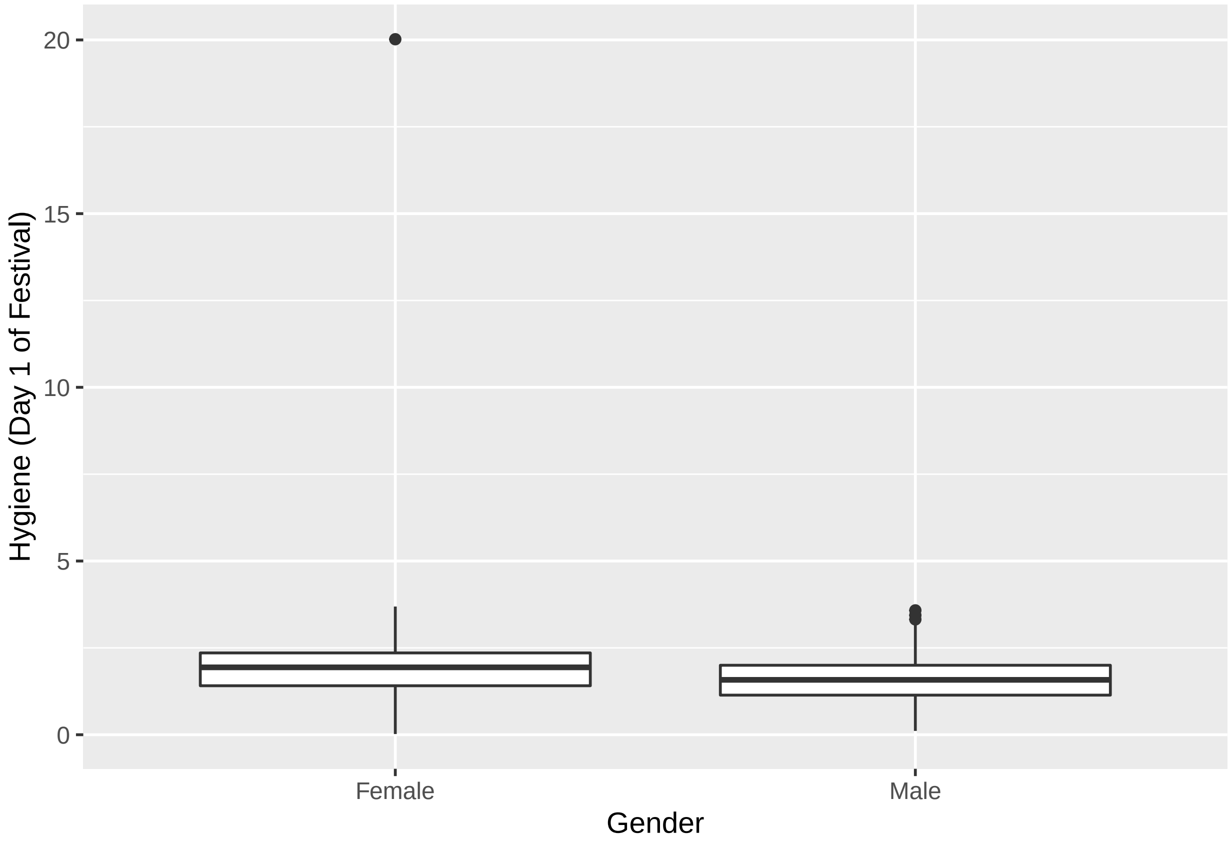


```
festivalDensity + geom_histogram(binwidth = 0.4) +
  labs(x = "Hygiene (Day 1 of Festival)", y = "Frequency") +
  theme(legend.position="none")
```

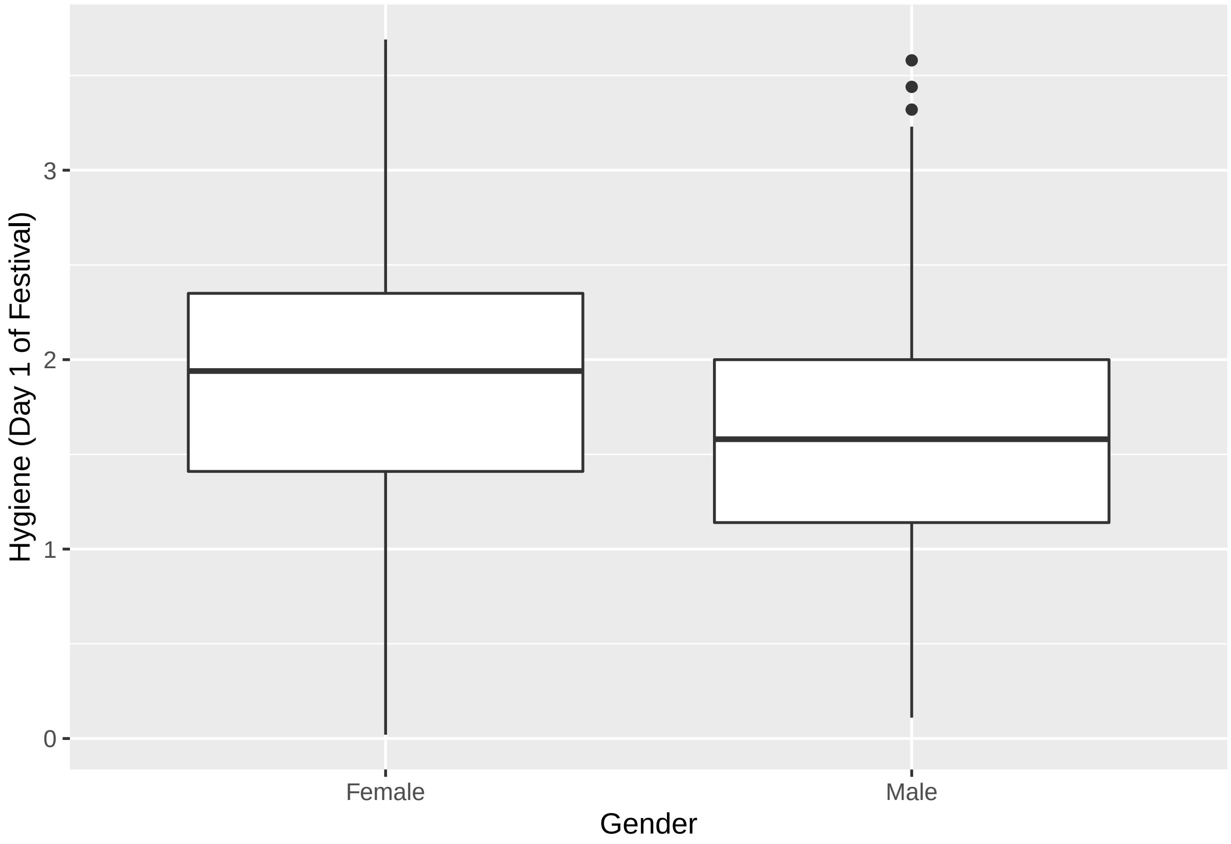


Boxplots

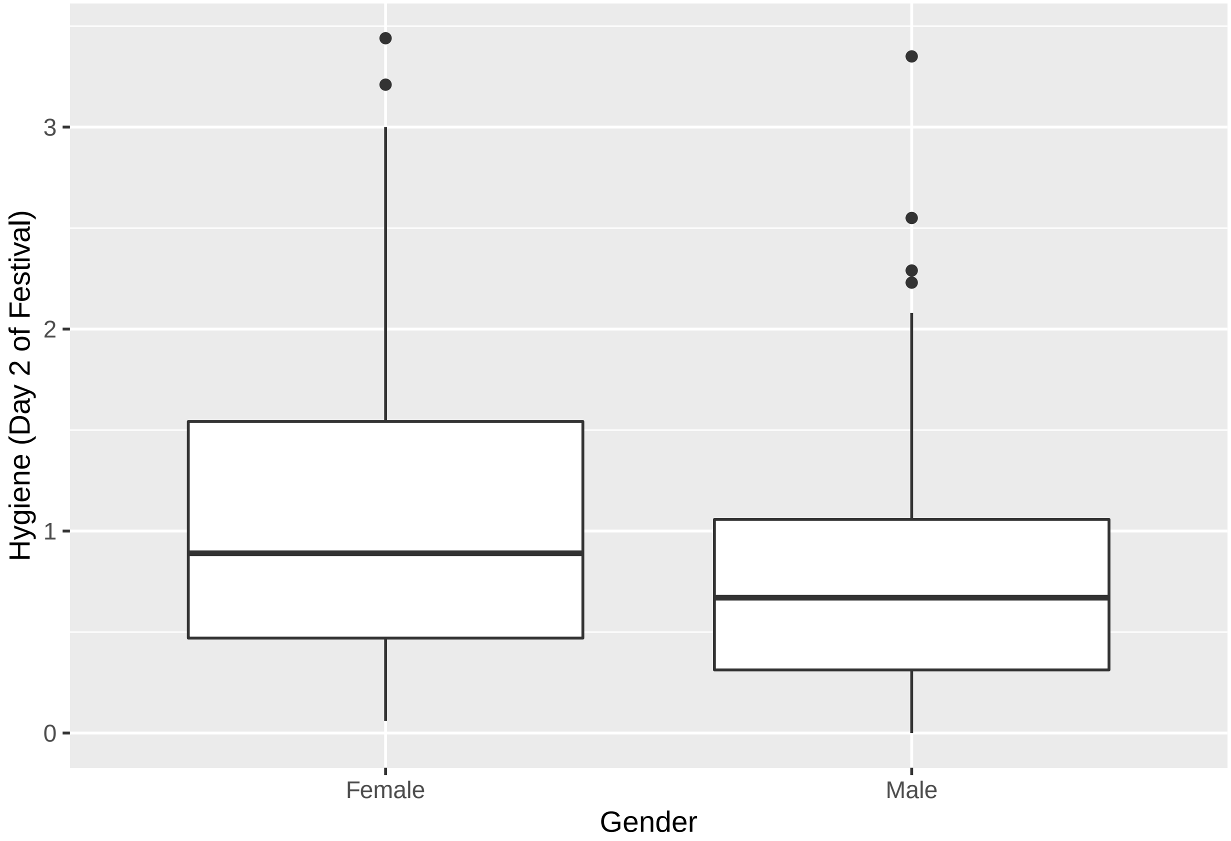
```
festivalBoxplot <- ggplot(festivalData, aes(gender, day1))
festivalBoxplot + geom_boxplot() + labs(x = "Gender", y = "Hygiene (Day 1 of Festival)")
```



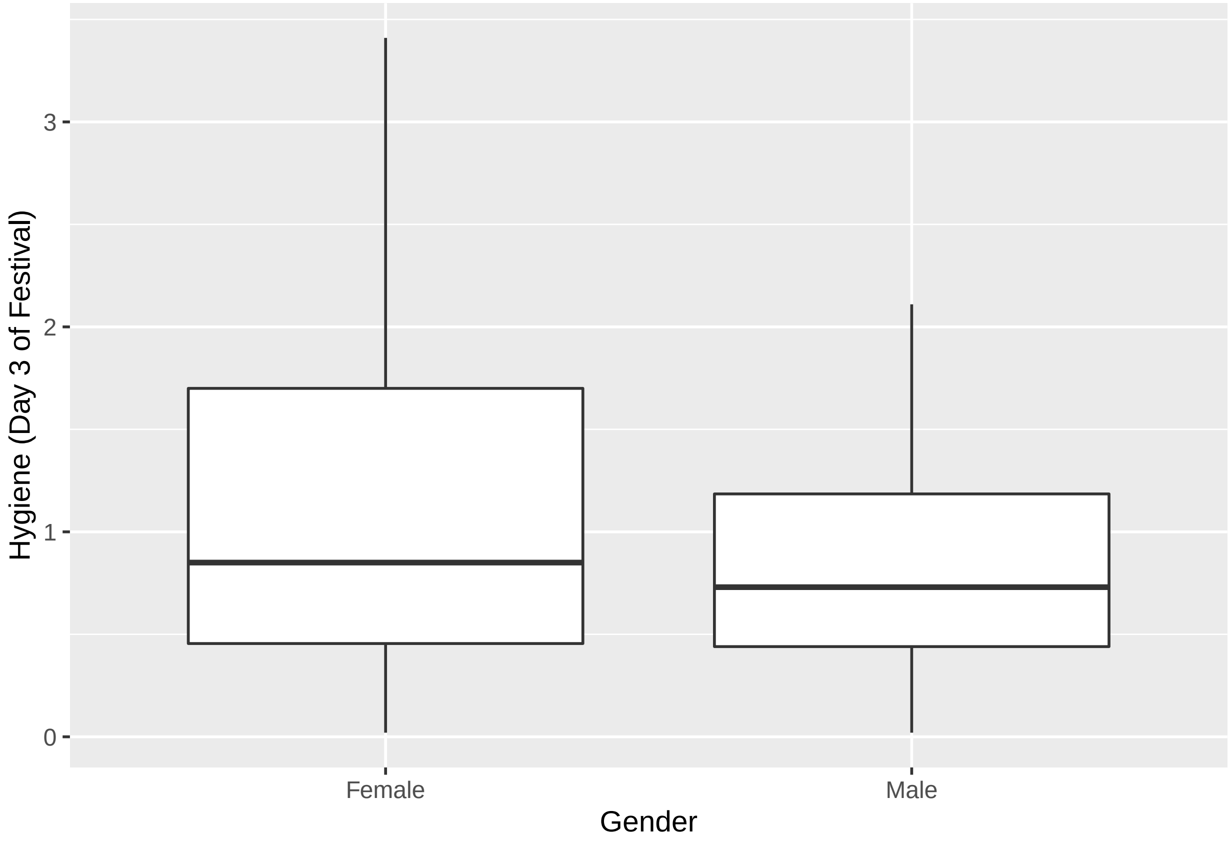
```
#with outlier removed
festivalData2 = read.delim("DownloadFestival(No Outlier).dat",  header = TRUE)

festivalBoxplot2 <- ggplot(festivalData2, aes(gender, day1))
festivalBoxplot2 + geom_boxplot() +
  labs(x = "Gender", y = "Hygiene (Day 1 of Festival)")
```



```r
#days 2 and 3
festivalBoxplot <- ggplot(festivalData, aes(gender, day2))
festivalBoxplot + geom_boxplot() +
  labs(x = "Gender", y = "Hygiene (Day 2 of Festival)")
```

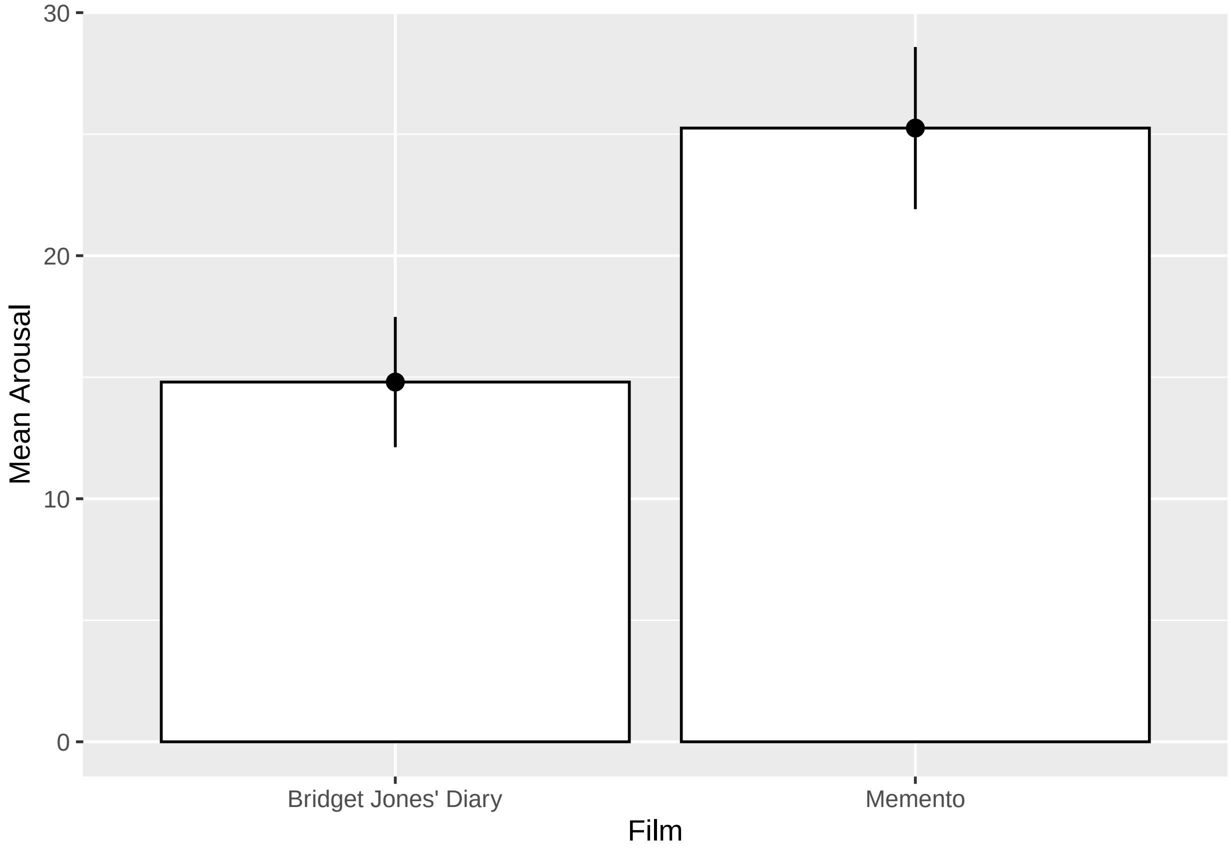


```
festivalBoxplot <- ggplot(festivalData, aes(gender, day3))
festivalBoxplot + geom_boxplot() +
  labs(x = "Gender", y = "Hygiene (Day 3 of Festival)")
```

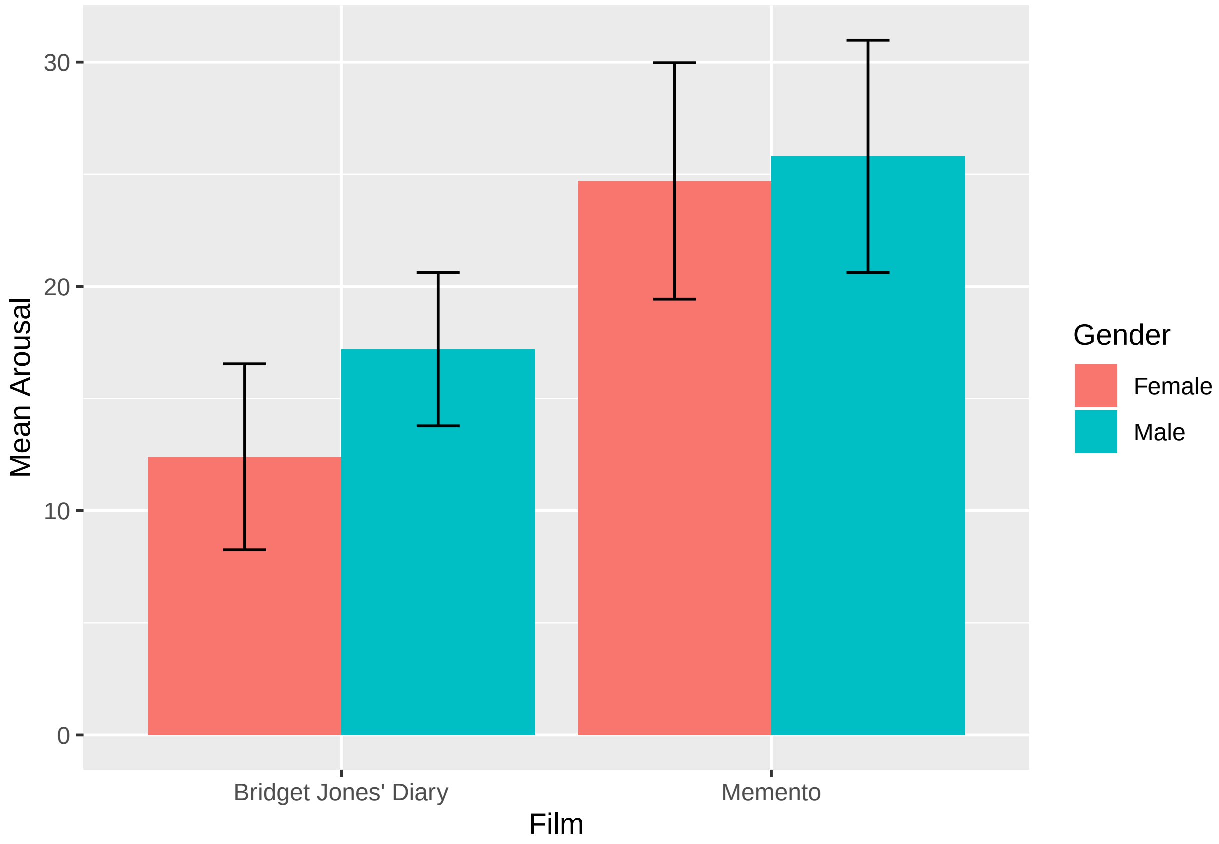


Bar Charts

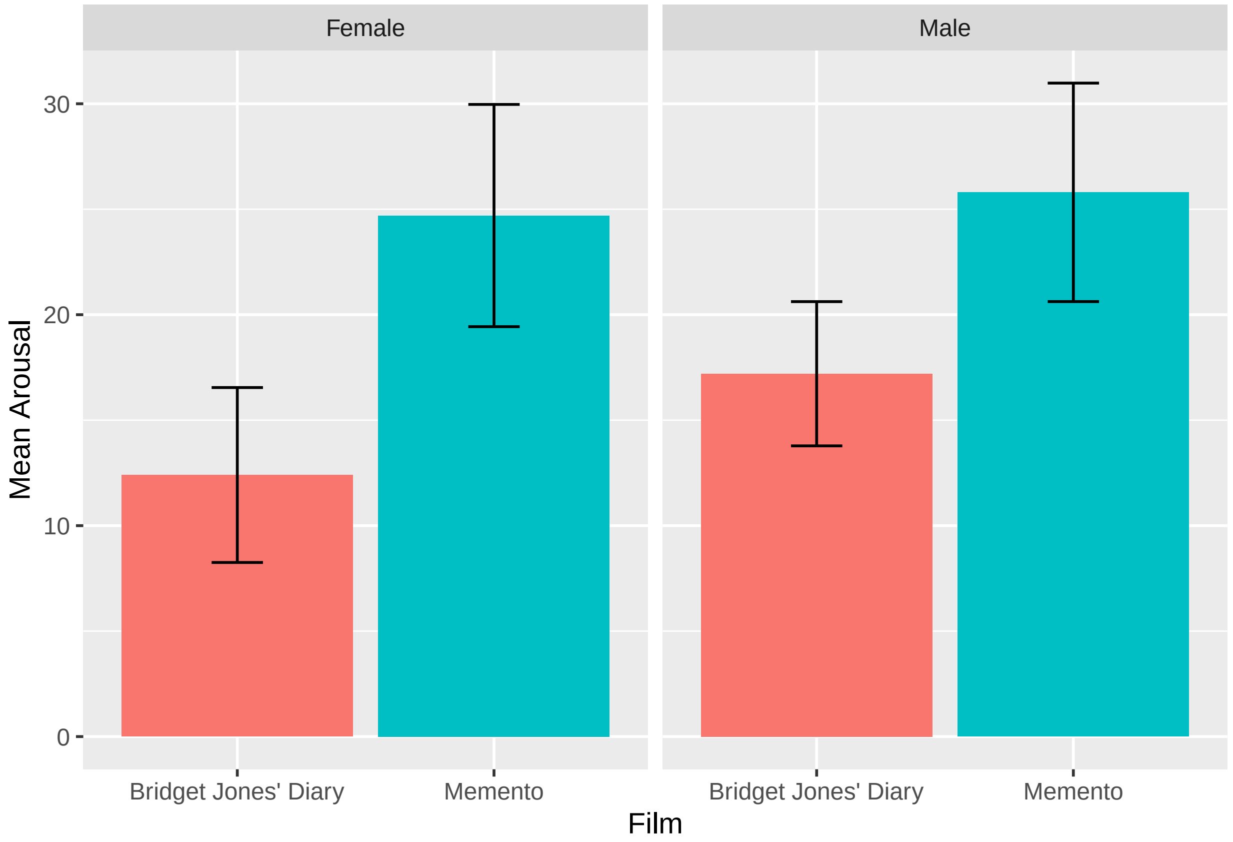
```
#loading in our data
chickFlick = read.delim("ChickFlick.dat",  header = TRUE)
#creating base plot
bar <- ggplot(chickFlick, aes(film, arousal))
bar +
  stat_summary(fun.y = mean, geom = "bar", fill = "White", colour = "Black") +
  stat_summary(fun.data = mean_cl_normal, geom = "pointrange") +
  labs(x = "Film", y = "Mean Arousal")
```



```r
colours = c(Female = "Red", Male = "Green")
bar <- ggplot(chickFlick)
bar +
  stat_summary(aes(film, arousal, fill = gender ),  fun.y = mean, geom = "bar", position="dodge") +
  stat_summary(aes(film, arousal, fill = gender ), fun.data = mean_cl_normal, geom = "errorbar",
               position=position_dodge(width=0.90), width = 0.2) +
  labs(x = "Film", y = "Mean Arousal", fill = "Gender")
```

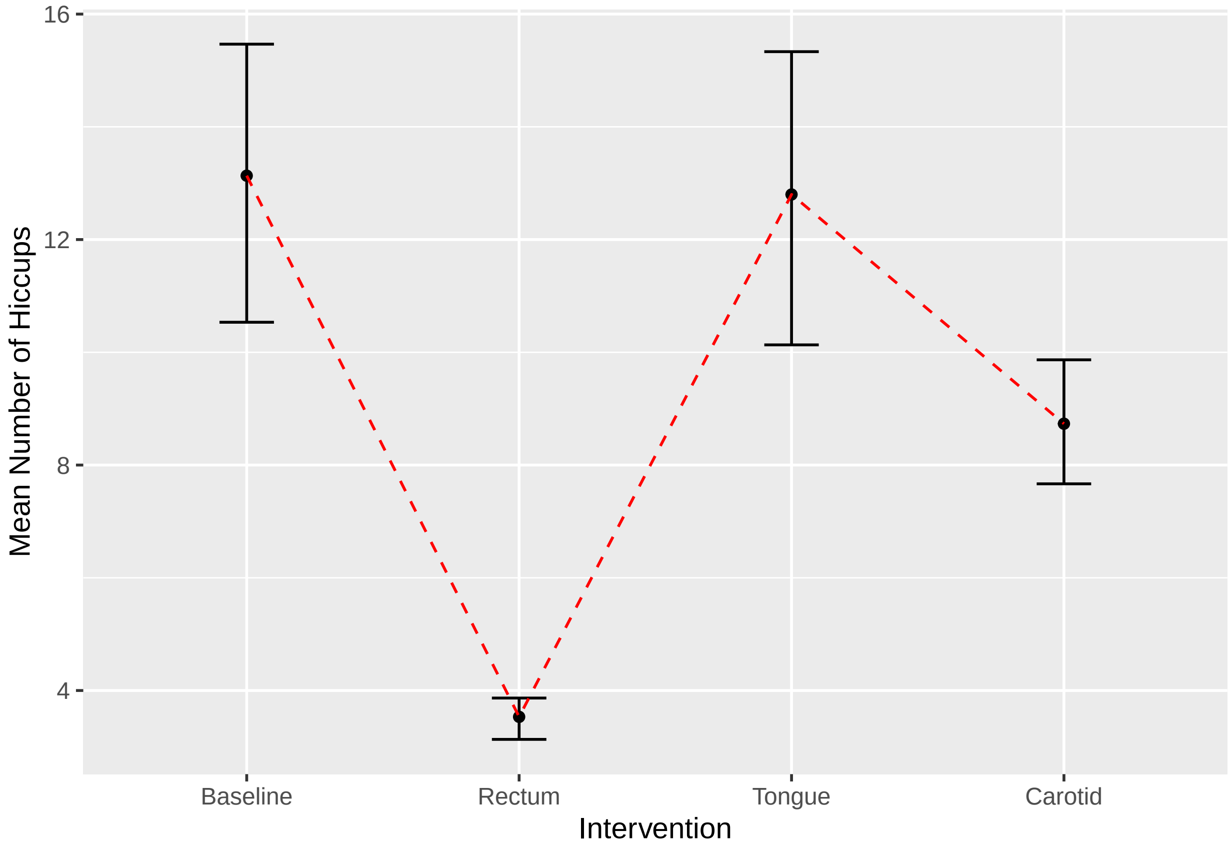


```r
bar <- ggplot(chickFlick, aes(film, arousal, fill = film))
bar + stat_summary(fun.y = mean, geom = "bar") +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar", width = 0.2) + facet_wrap(~gender) +
  labs(x = "Film", y = "Mean Arousal") + theme(legend.position="none")
```



Line Charts

```
hiccupsData <- read.delim("Hiccups.dat",  header = TRUE)
hiccups<-stack(hiccupsData)
names(hiccups)<-c("Hiccups","Intervention")
hiccups$Intervention_Factor<-factor(hiccups$Intervention, levels(hiccups$Intervention)[c(1, 4, 2, 3)])

line <- ggplot(hiccups,  aes(Intervention_Factor, Hiccups))
line + stat_summary(fun.y = mean, geom = "point") + stat_summary(fun.data = mean_cl_boot, geom = "errorb
```



Summary/Descriptive Tables

```
library(knitr)
library(apaTables)
```

```
## Warning: package 'apaTables' was built under R version 3.6.3
```

```
objects(chickFlick)
```

```
## [1] "arousal" "film"    "gender"
```

```r
names(chickFlick)
```

```
## [1] "gender"  "film"    "arousal"
```

```r
names(chickFlick)[1] <- "Gender"
names(chickFlick)[2] <- "Film"
names(chickFlick)[3] <- "Arousal"
names(chickFlick)
```

```
## [1] "Gender"  "Film"    "Arousal"
```

```r
apa.2way.table(Gender, Film, Arousal, data = chickFlick, table.number = 1,
  show.conf.interval = FALSE, show.marginal.means = FALSE,
  landscape = TRUE, filename = "flick.doc")
```

```
##
##
## Table 1
##
## Means and standard deviations for Arousal as a function of a 2(Gender) X 2(Film) design
##
##                          Film
##         Bridget Jones' Diary      Memento
##   Gender                     M    SD        M    SD
##   Female                12.40 5.80   24.70 7.36
##     Male                17.20 4.78   25.80 7.24
##
## Note. M and SD represent mean and standard deviation, respectively.
```