Solutions to ggplot2 Exercises

Based on *R Graphics Cookbook* by Winston Chang

Colonel Smoothie

27.12.2013

Introduction

Just leaving some notes here. I've been having some trouble with the tidy so I'm explicitly setting the option here:

```
opts_chunk$set(tidy = FALSE)
```

Things to do

I don't want this to be too similar to Chang's book or else there's not really much point in making this document, other than having it in question format so you can use active recall to test yourself. For the most part, I'll be creating exercises using the `MASS` package in addition to what's already in the book, they will be tailored to match the chapters but will be slightly different so you'll have something new.

I've also added a Miscellanous chapter for features of ggplot2 that are not covered in the textbook as well as an Actuarial Graphics chapter that includes actuarial applications of ggplot2.

Chapter 1

R Basics

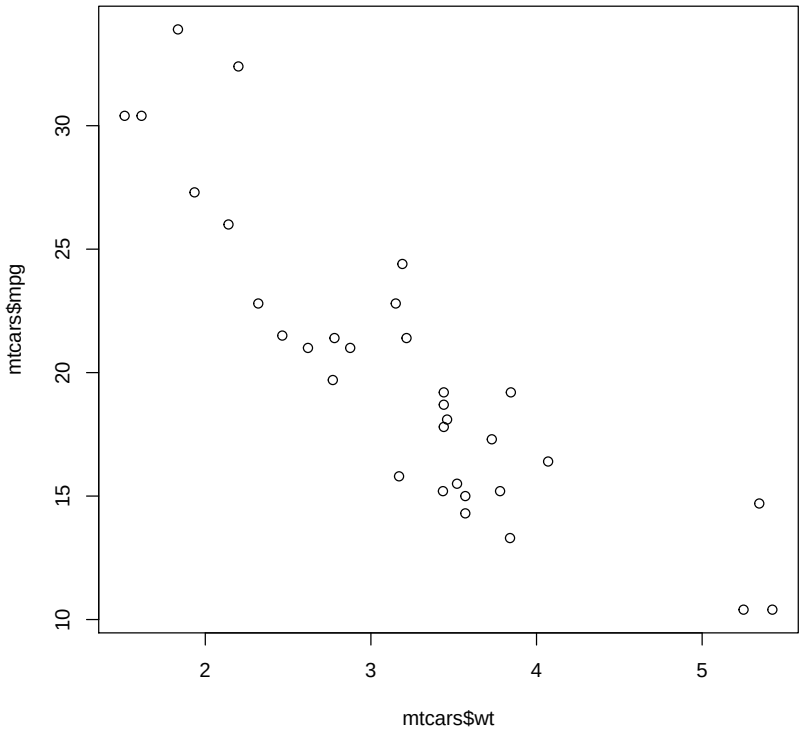
There's not really much to add for this chapter. Move on to the next one.

Chapter 2

Quickly Exploring Data

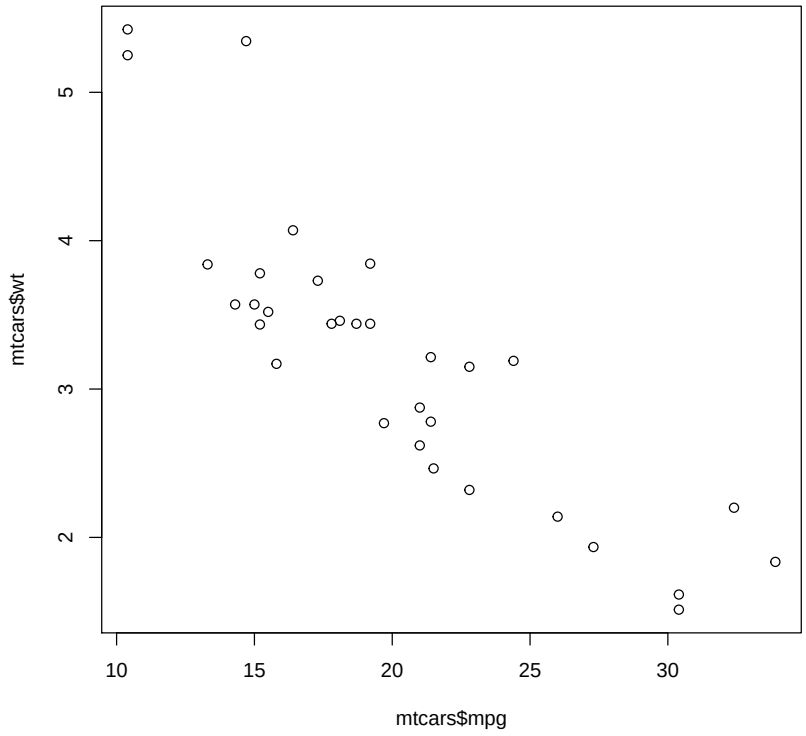
1. Produce the following plot with the `mtcars` dataset. It's built into R so you do not need to load any packages:

```
plot(mtcars$wt, mtcars$mpg)
```



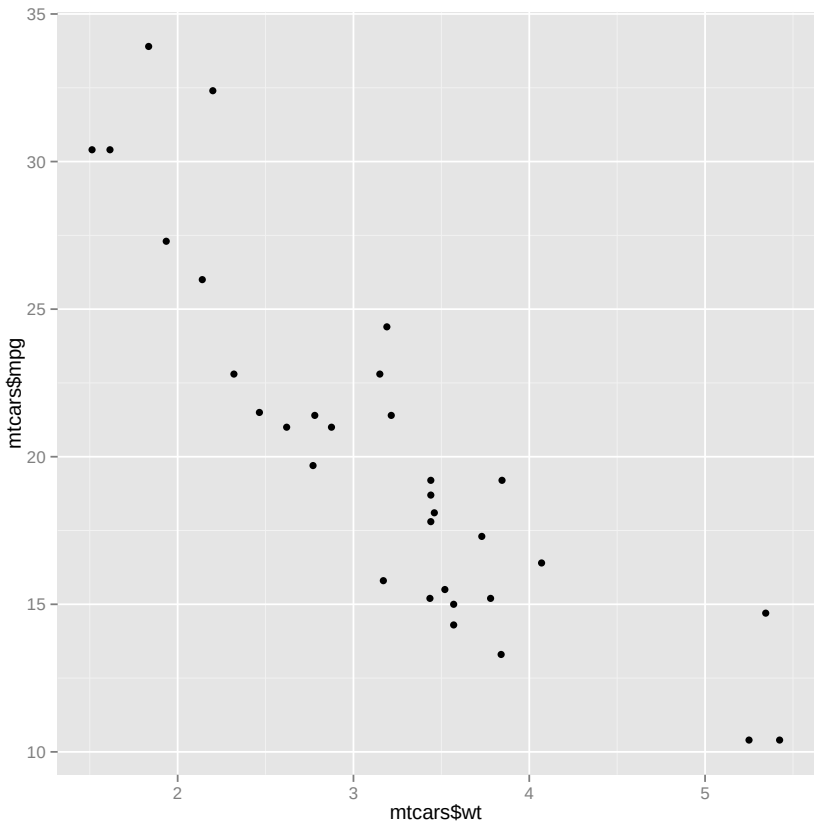
2. Produce the following plot with the `mtcars` dataset. It's built into R so you do not need to load any packages:

```
plot(mtcars$mpg, mtcars$wt)
```



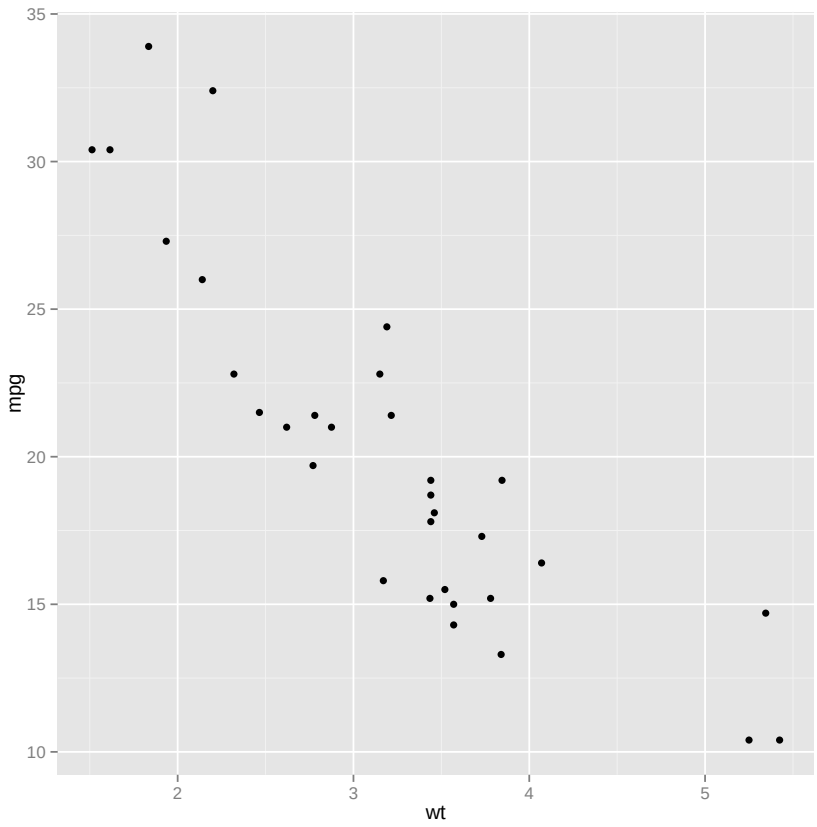
3. Load the ggplot2 package and produce the following plot with the mtcars dataset:

```
library(ggplot2)
qplot(mtcars$wt, mtcars$mpg)
```



4. Load the ggplot2 package and produce the following plot with the mtcars dataset:

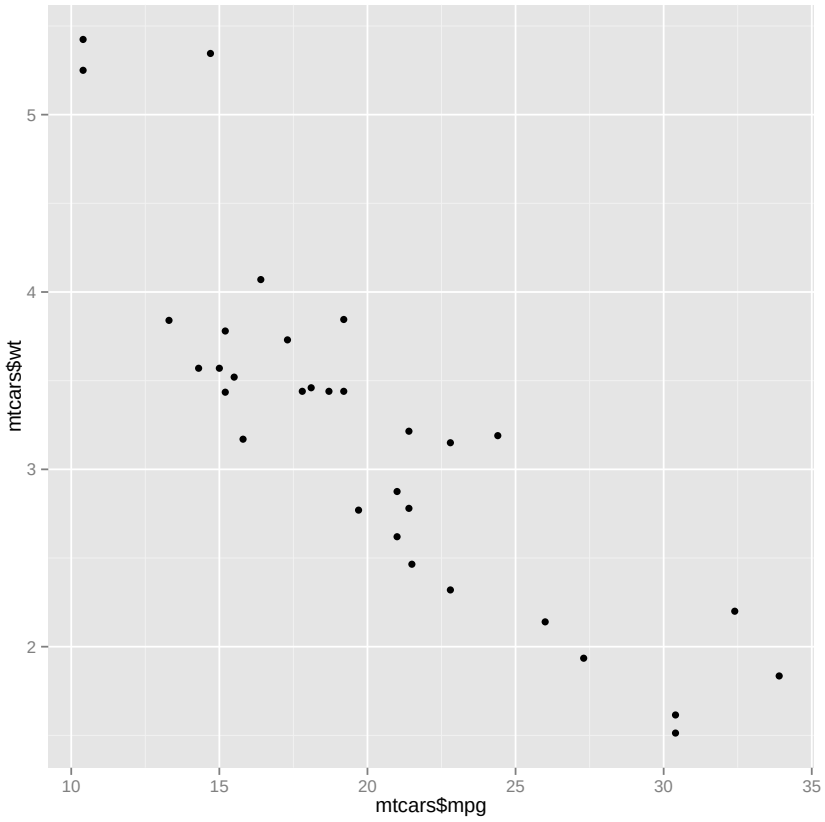
```
library(ggplot2)
qplot(wt, mpg, data=mtcars)
```




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

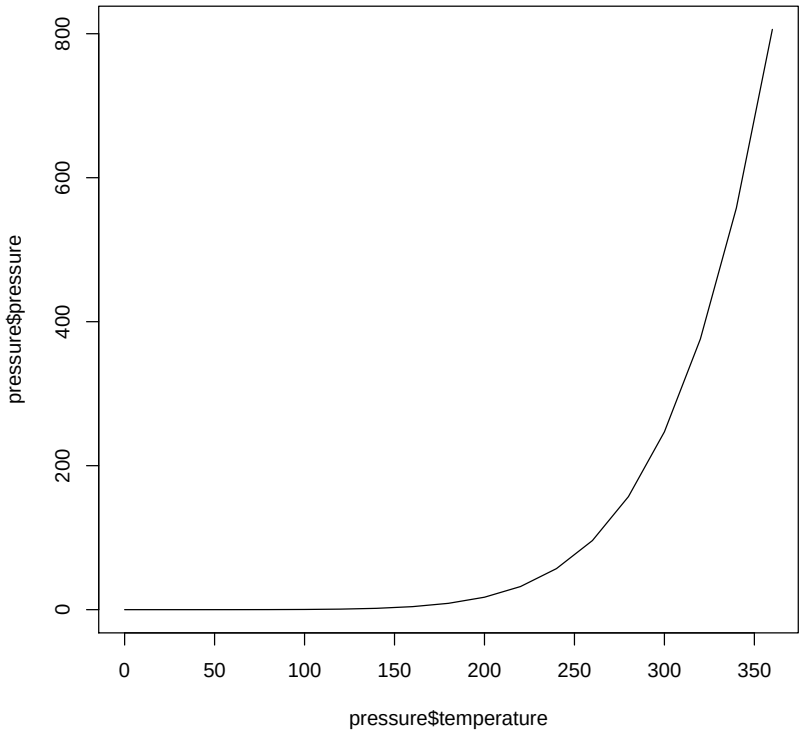
5. Load the ggplot2 package and produce the following plot with the mtcars dataset:

```
library(ggplot2)
qplot(mtcars$mpg, mtcars$wt)
```



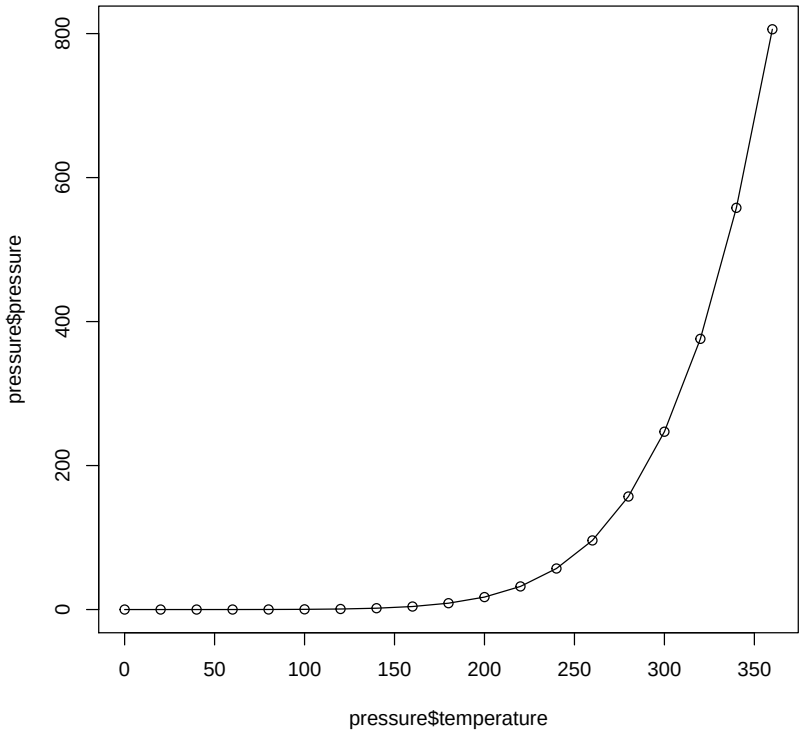
6. Produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages:

```
plot(pressure$temperature, pressure$pressure, type = "l")
```



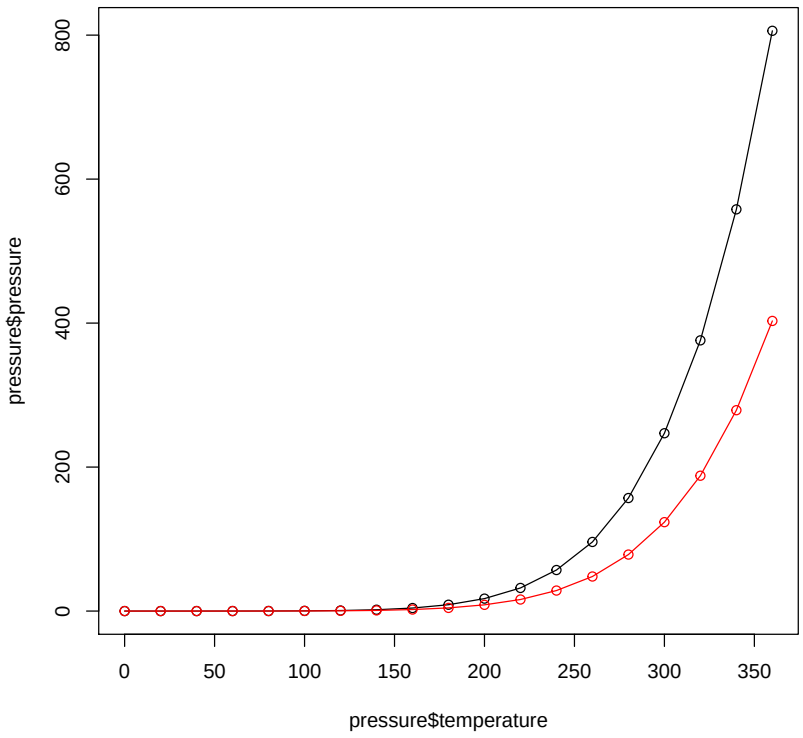
7. Produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages:

```
plot(pressure$temperature, pressure$pressure, type = "l")
points(pressure$temperature, pressure$pressure)
```



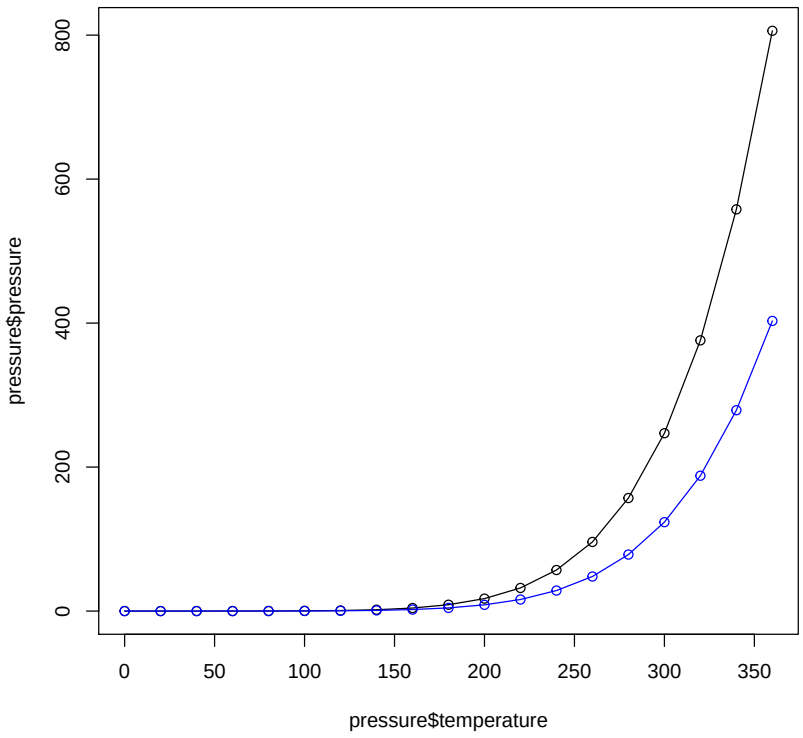
8. Produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages. The height of the red line is one-half of the height of the black line at all points:

```
plot(pressure$temperature, pressure$pressure, type = "l")
points(pressure$temperature, pressure$pressure)

lines(pressure$temperature, pressure$pressure/2, col="red")
points(pressure$temperature, pressure$pressure/2, col="red")
```



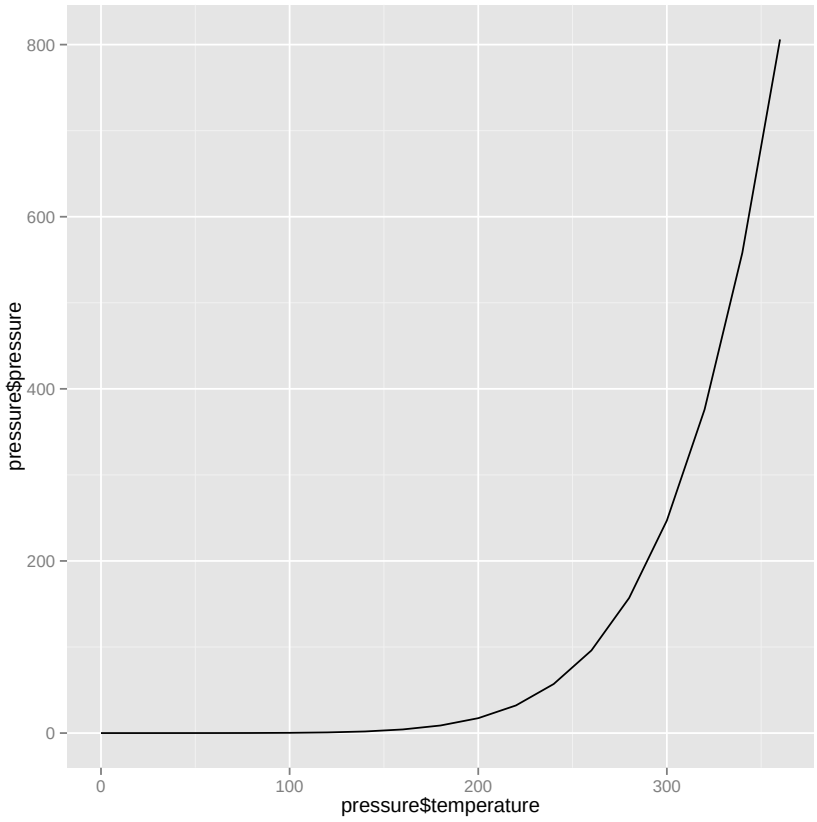
9. Produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages. The height of the blue line is one-half of the height of the black line at all points:

```
plot(pressure$temperature, pressure$pressure, type = "l")
points(pressure$temperature, pressure$pressure)

lines(pressure$temperature, pressure$pressure/2, col="blue")
points(pressure$temperature, pressure$pressure/2, col="blue")
```



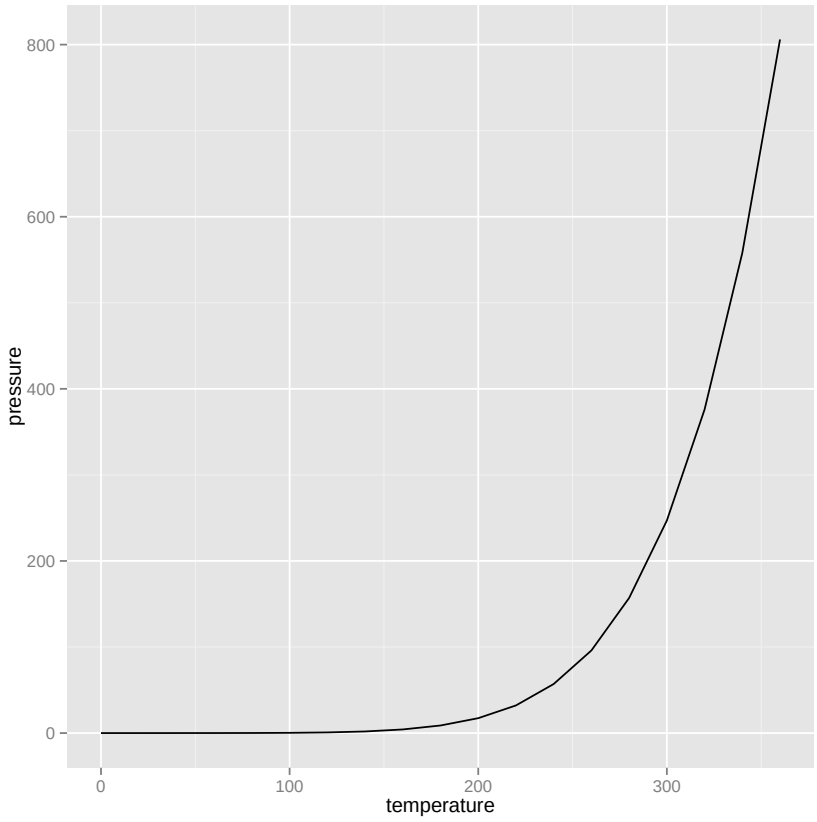
10. Load the ggplot2 package and produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(pressure$temperature, pressure$pressure, geom="line")
```



11. Load the ggplot2 package and produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages:

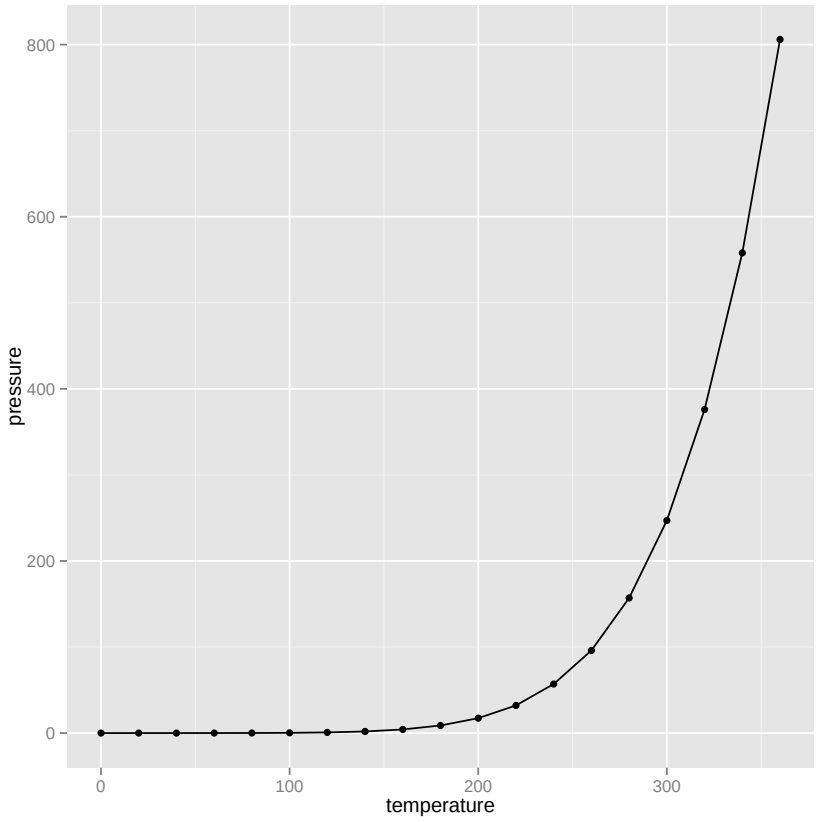
```
library(ggplot2)
qplot(temperature, pressure, data=pressure, geom="line")
```




```
#Alternative Solution
library(ggplot2)
ggplot(pressure, aes(x=temperature, y=pressure))+geom_line()
```

12. Load the ggplot2 package and produce the following plot with the pressure dataset. It's built into R so you do not need to load any packages:

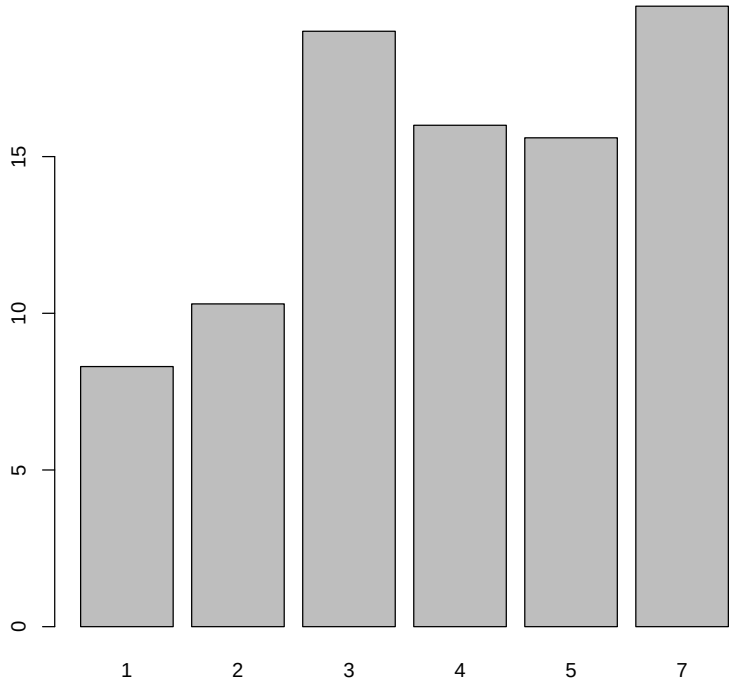
```
library(ggplot2)
qplot(temperature, pressure, data=pressure, geom=c("line", "point"))
```




```
library(ggplot2)
ggplot(pressure, aes(x=temperature, y=pressure)) + geom_line() + geom_point()
```

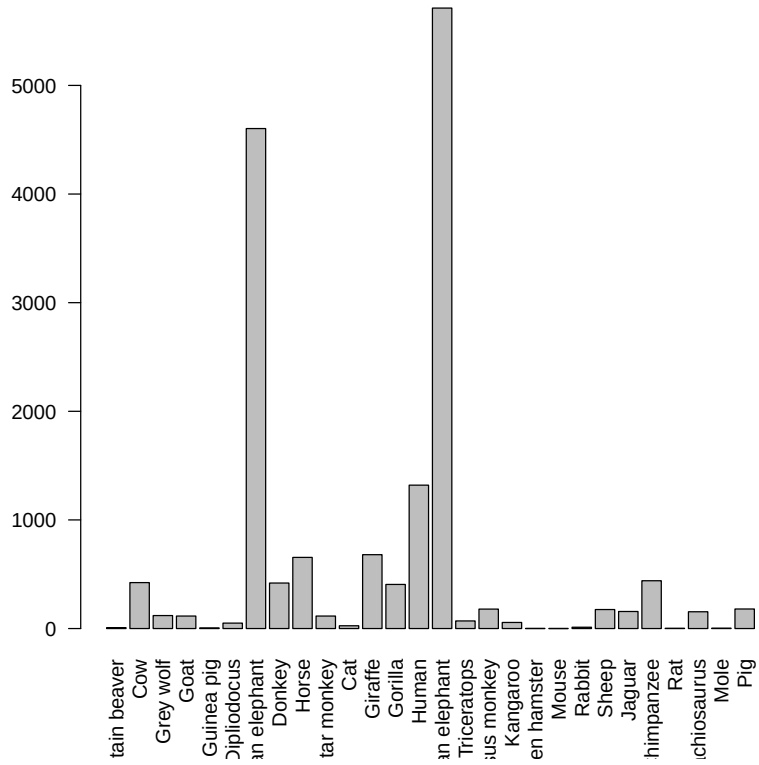
13. Produce the following plot with the BOD dataset. It's built into R so you do not need to load any packages:

```
barplot(BOD$demand, names.arg=BOD$Time)
```



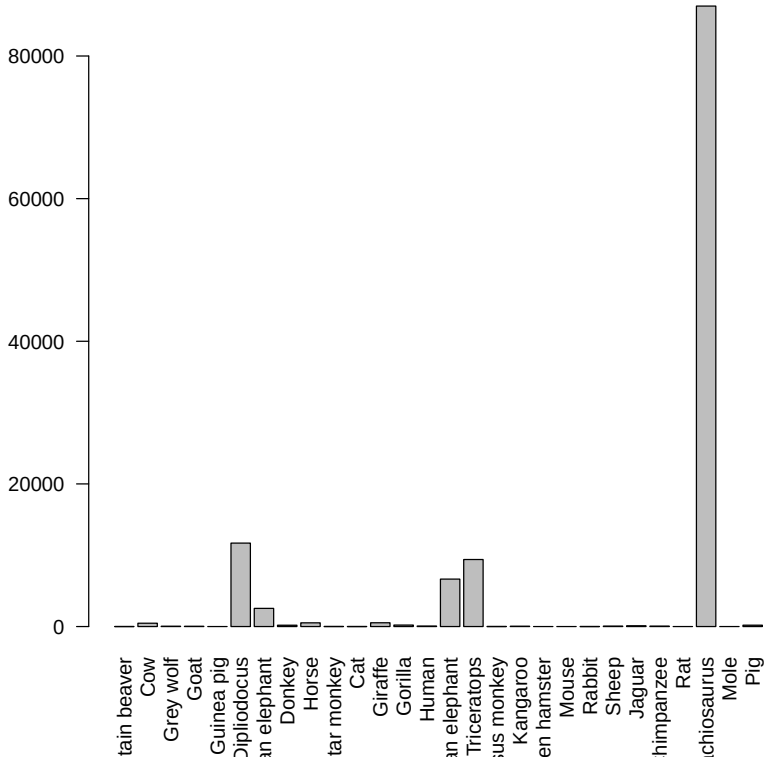
14. Load the MASS package and produce the following plot with the Animals dataset:

```
library(MASS)
barplot(Animals$brain, names.arg=row.names(Animals), las=2)
```



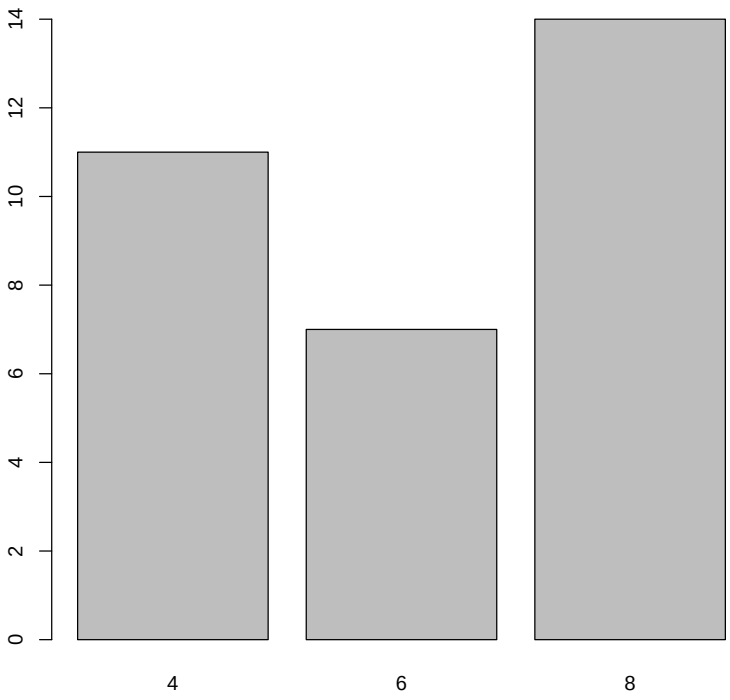
15. Load the MASS package and produce the following plot with the Animals dataset:

```
library(MASS)
barplot(Animals$body, names.arg=row.names(Animals), las=2)
```



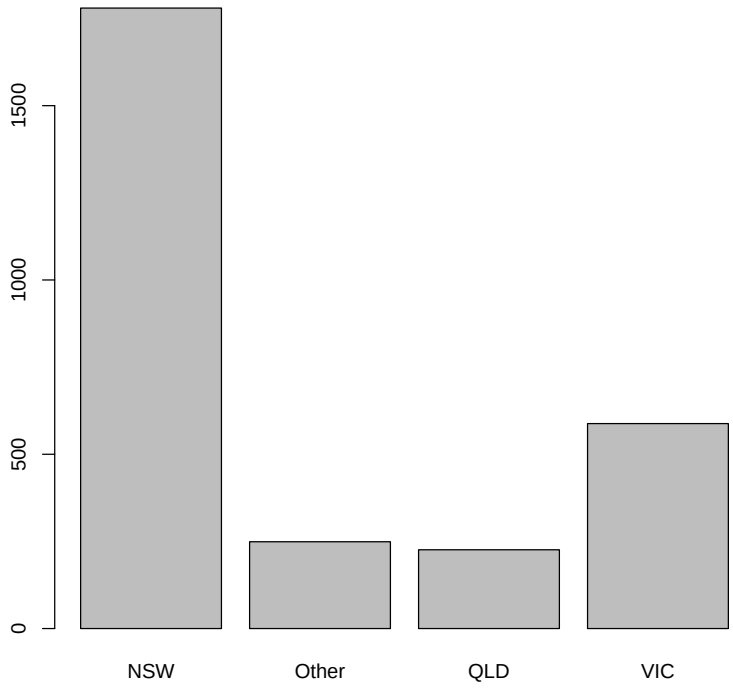
16. Produce the following plot with the `mtcars` dataset. It's built into R so you do not need to load any packages:

```
barplot(table(mtcars$cyl))
```



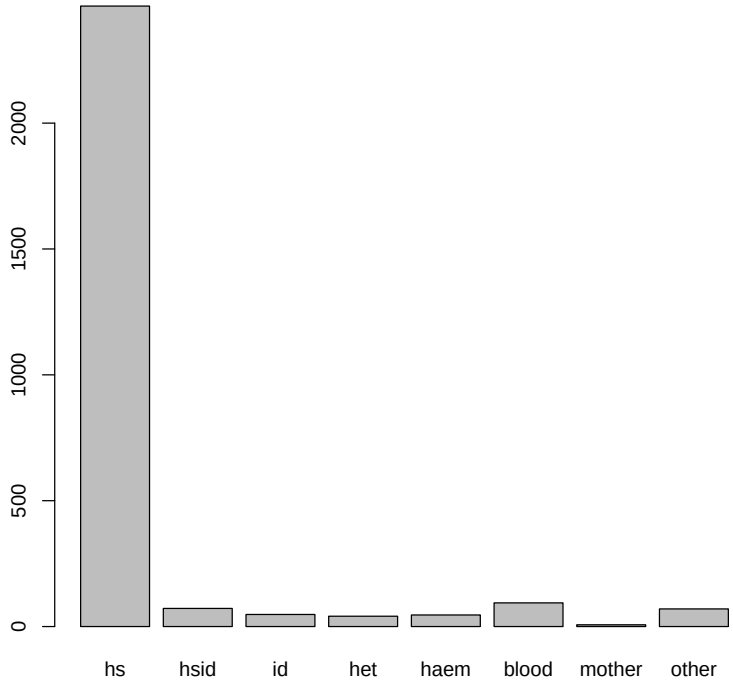
17. Load the MASS package and produce the following plot with the Aids2 dataset:

```
library(MASS)
barplot(table(Aids2$state))
```



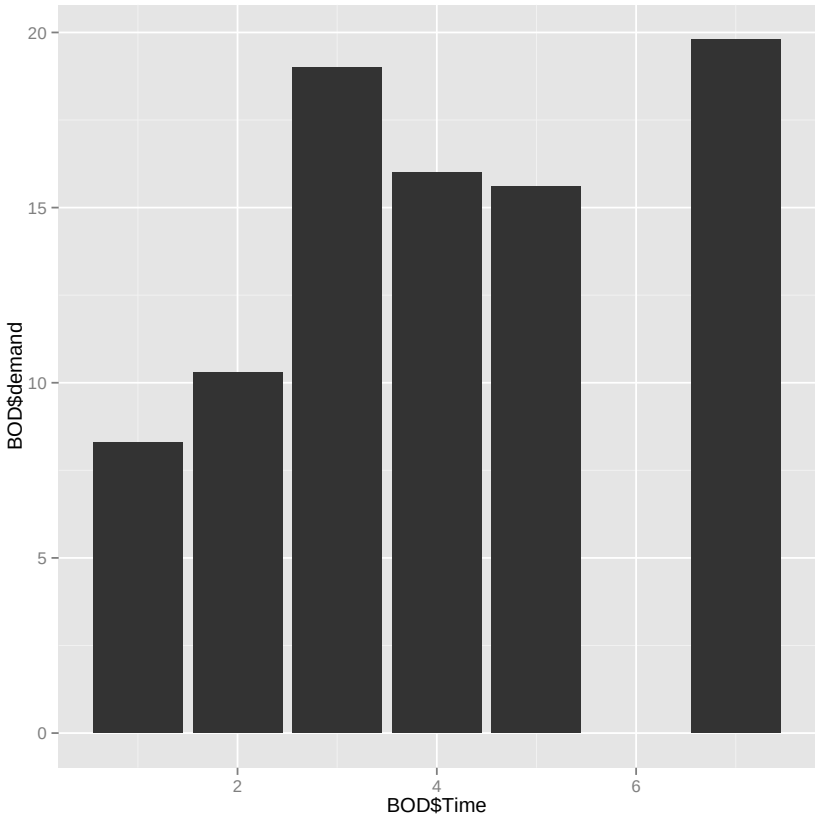
18. Load the MASS package and produce the following plot with the Aids2 dataset:

```
library(MASS)
barplot(table(Aids2$T.categ))
```



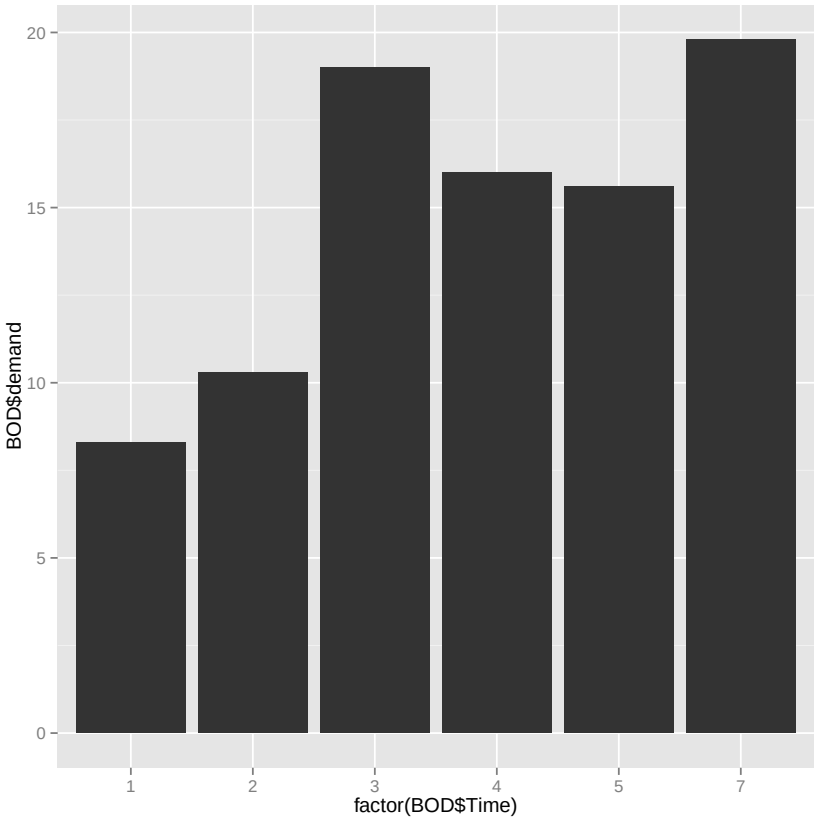
19. Load the ggplot2 package and produce the following plot with the BOD dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(BOD$Time, BOD$demand, geom="bar", stat="identity")
```



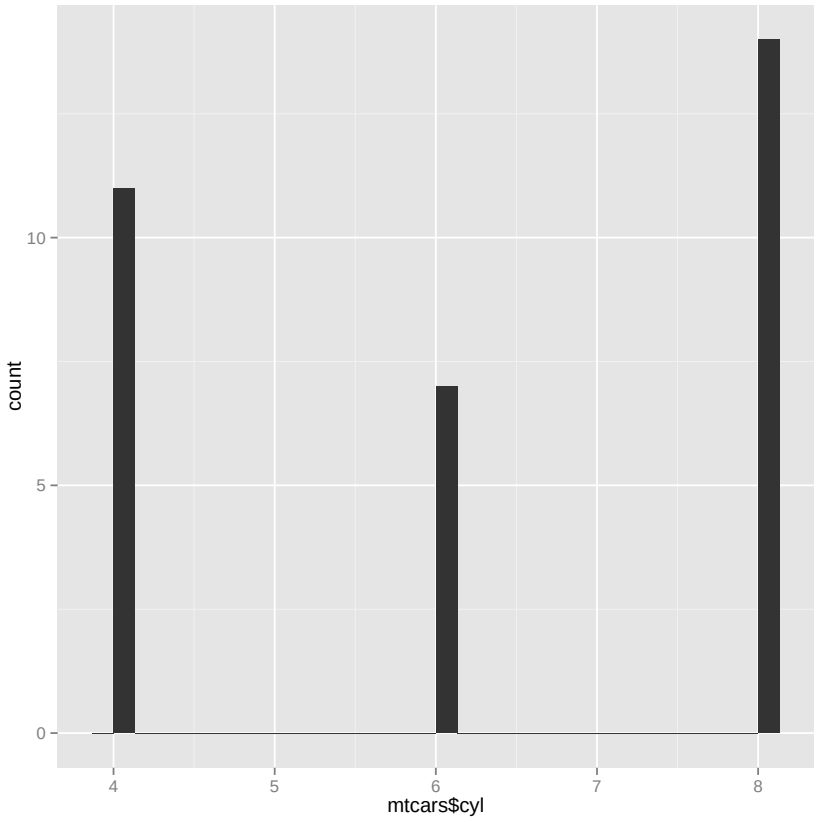
20. Load the ggplot2 package and produce the following plot with the BOD dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(factor(BOD$Time), BOD$demand, geom="bar", stat="identity")
```



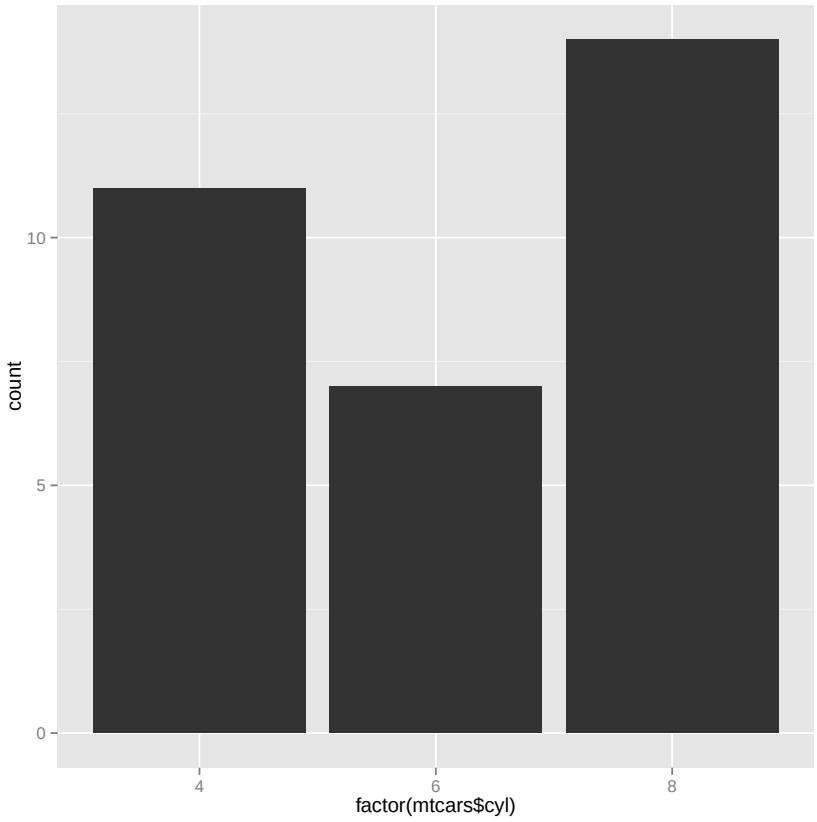
21. Load the ggplot2 package and produce the following plot with the mtcars dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(mtcars$cyl)
```



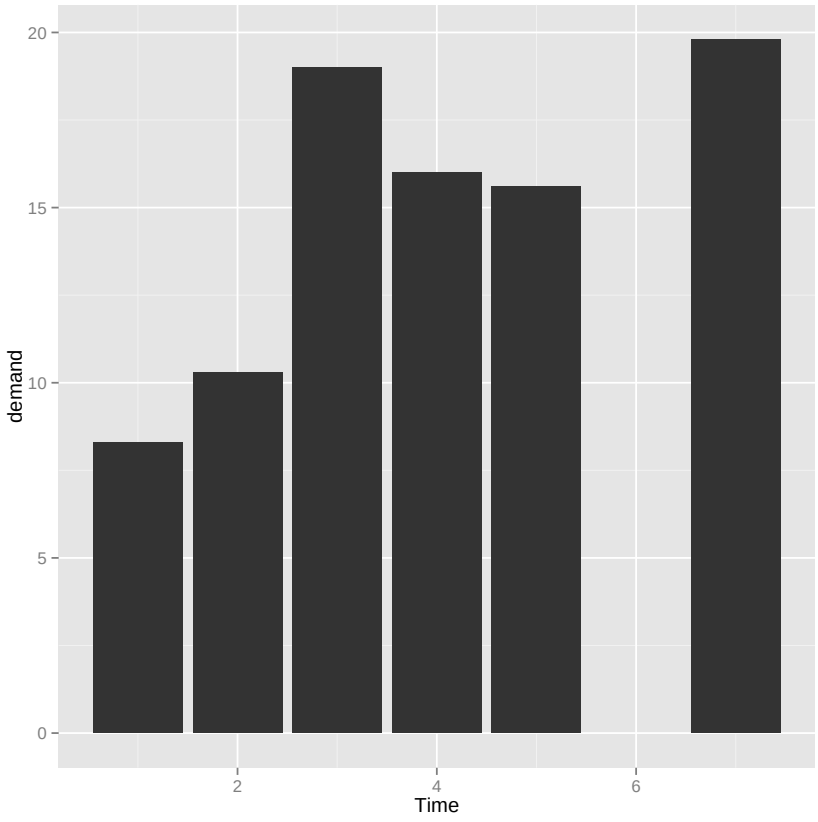
22. Load the ggplot2 package and produce the following plot with the mtcars dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(factor(mtcars$cyl))
```



23. Load the ggplot2 package and produce the following plot with the BOD dataset. It's built into R so you do not need to load any packages:

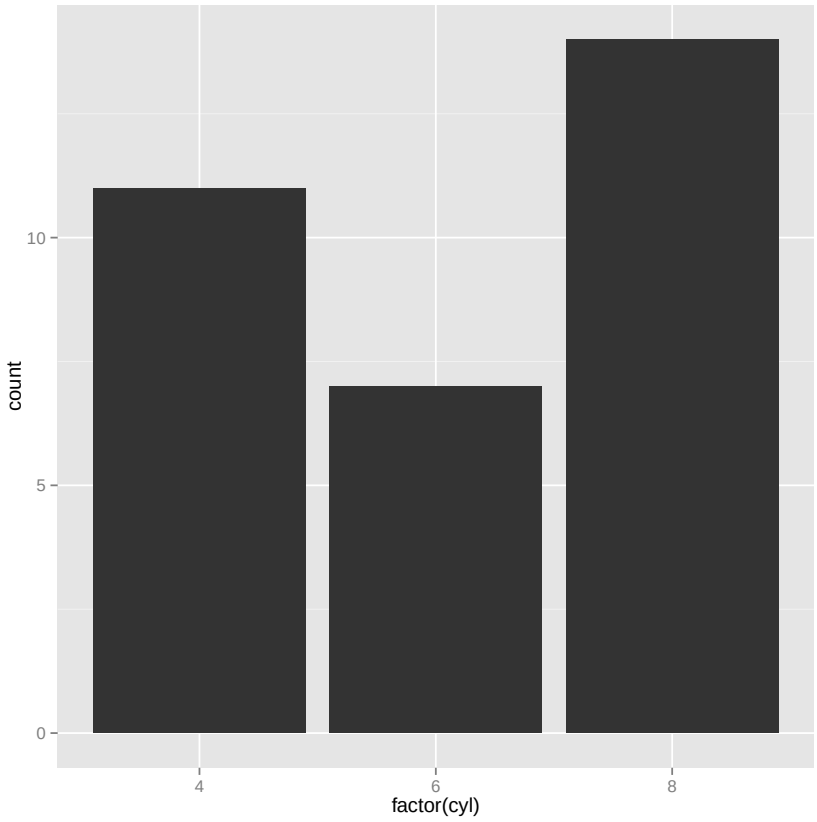
```
library(ggplot2)
qplot(Time, demand, data=BOD, geom="bar", stat="identity")
```




```
#Alternate Solution
library(ggplot2)
ggplot(BOD, aes(x=Time, y=demand)) + geom_bar(stat="identity")
```

24. Load the ggplot2 package and produce the following plot with the mtcars dataset. It's built into R so you do not need to load any packages:

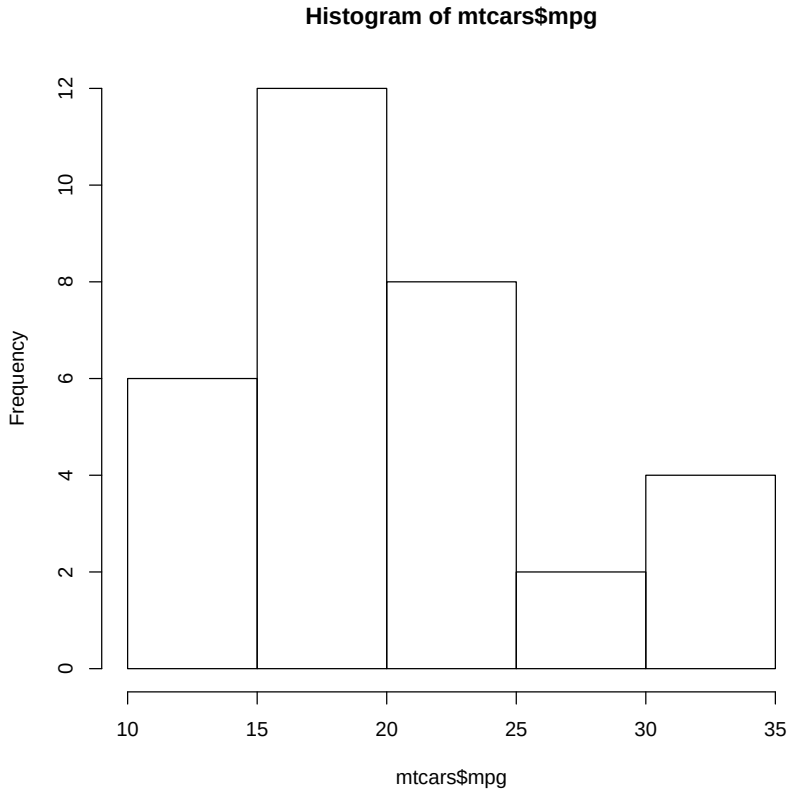
```
library(ggplot2)
qplot(factor(cyl), data=mtcars)
```




```
#Alternate Solution
library(ggplot2)
ggplot(mtcars, aes(x=factor(cyl))) + geom_bar()
```

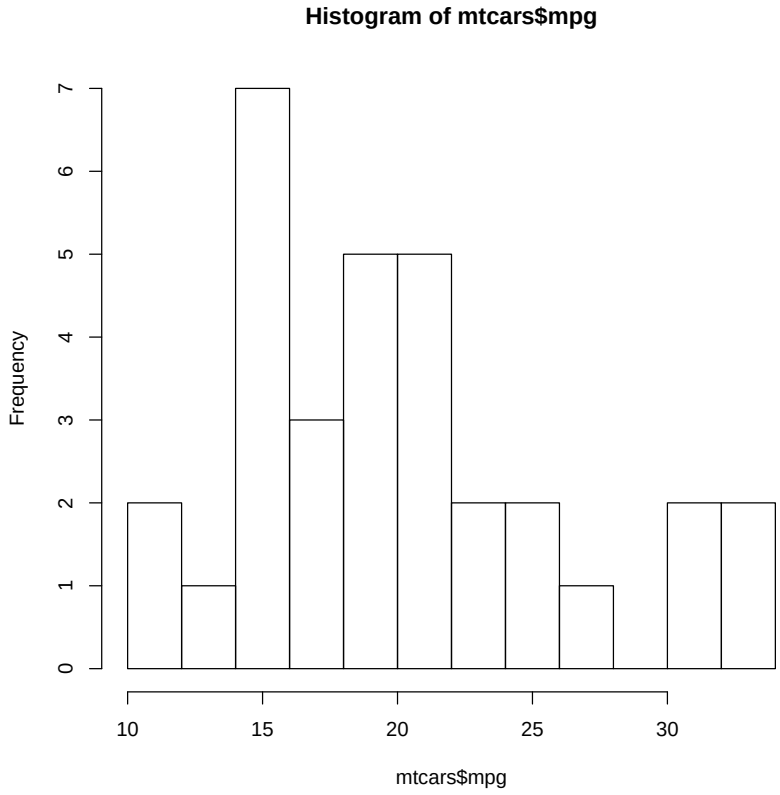
25. Produce the following plot using the `mtcars` dataset. It's built into R so you do not need to load any packages:

```
hist(mtcars$mpg)
```



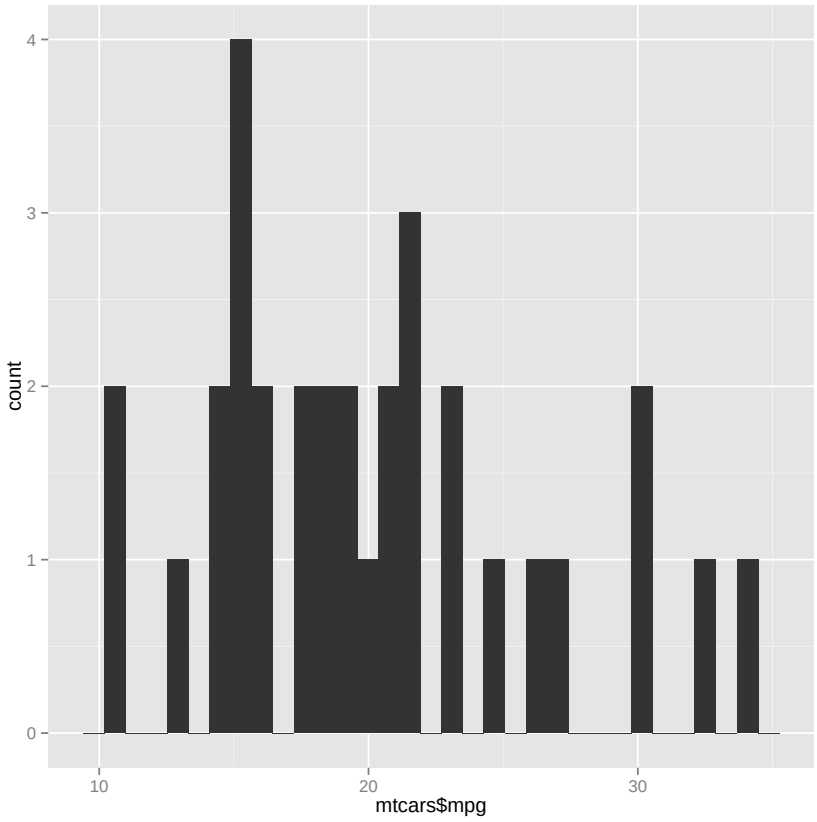
26. Produce the following plot using the `mtcars` dataset. It's built into R so you do not need to load any packages. This time, use 10 bins in the plot:

```
hist(mtcars$mpg, breaks=10)
```



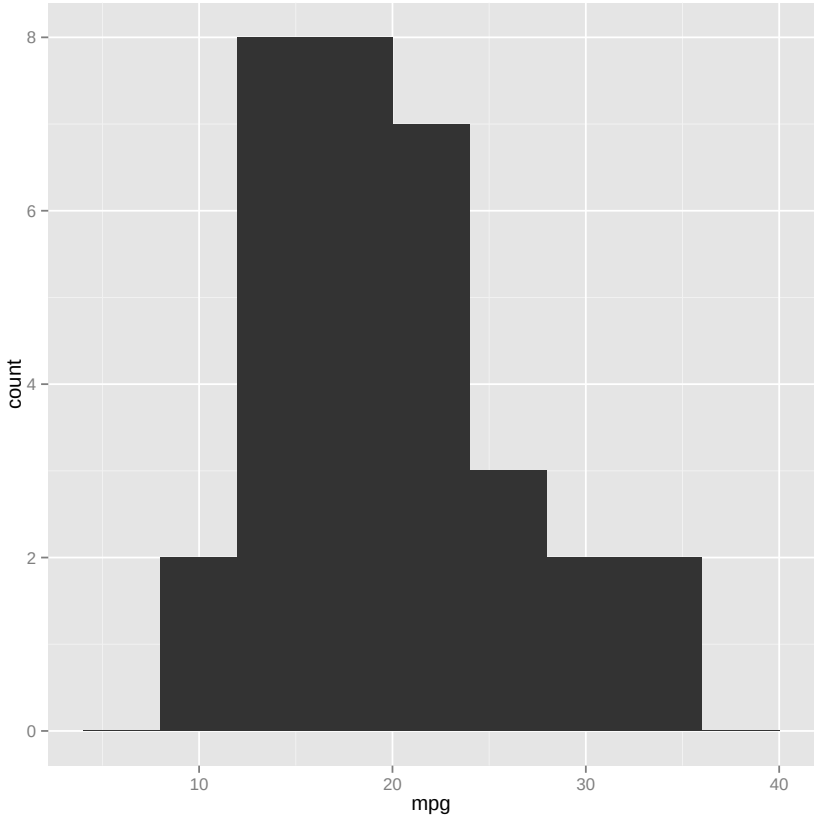
27. Load the ggplot2 package and produce the following plot with the mtcars dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(mtcars$mpg)
```



28. Load the texttt ggplot2 package and produce the following plot with the mtcars dataset. It's built into R so you do not need to load any packages. Set the width of the bins to 4:

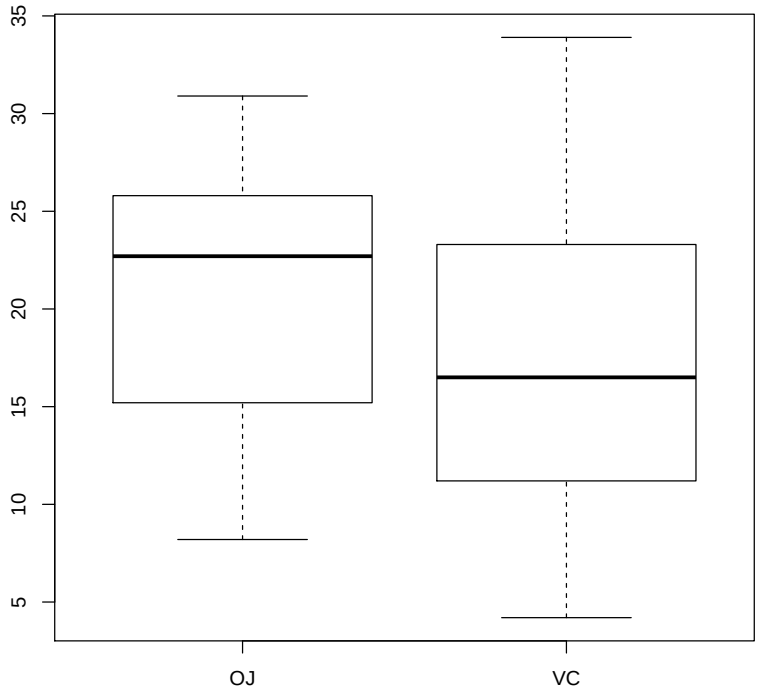
```
library(ggplot2)
qplot(mpg, data=mtcars, binwidth=4)
```




```
#Alternate Solution
library(ggplot2)
ggplot(mtcars, aes(x=mpg)) + geom_histogram(binwidth=4)
```

29. Produce the follwing plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

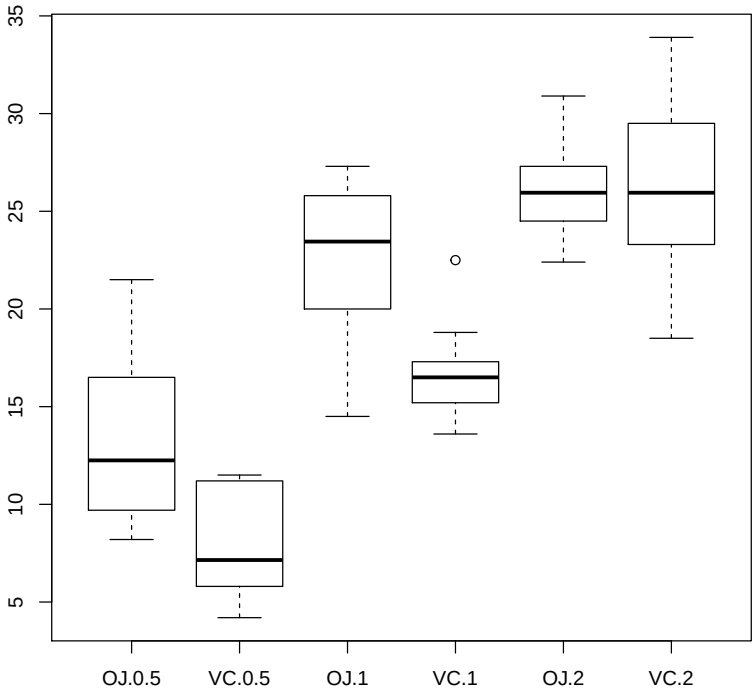
```
plot(ToothGrowth$supp, ToothGrowth$len)
```




```
#Alternate Solution
boxplot(len ~ supp, data = ToothGrowth)
```

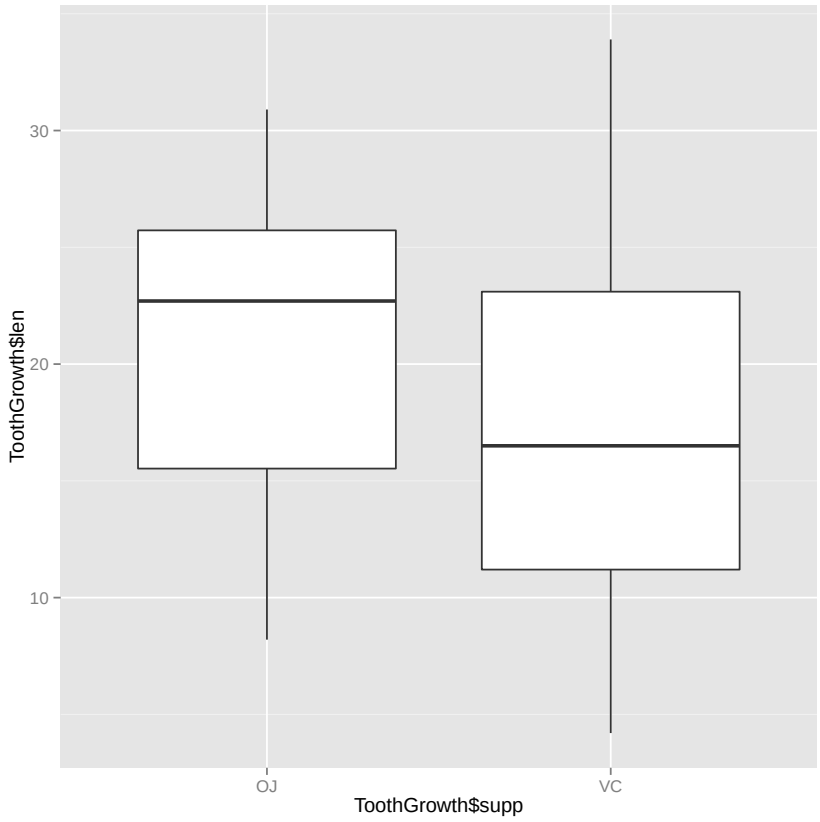
30. Produce the following plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

```
boxplot(len ~ supp + dose, data = ToothGrowth)
```



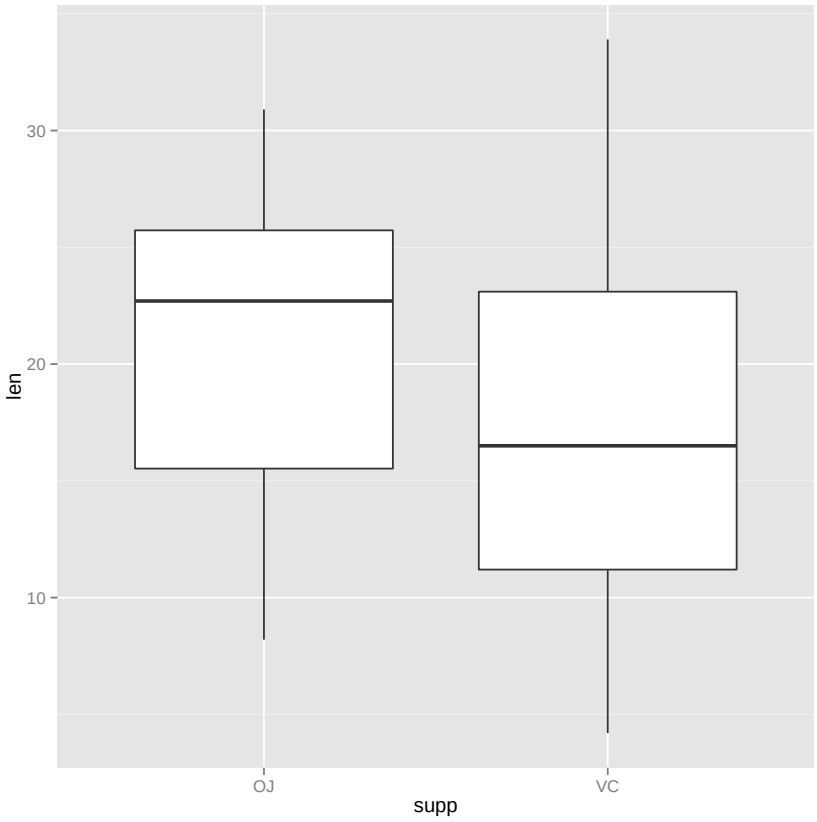
31. Load the ggplot2 package and produce the following plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

```
library(ggplot2)
qplot(ToothGrowth$supp, ToothGrowth$len, geom="boxplot")
```



32. Load the ggplot2 package and produce the following plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

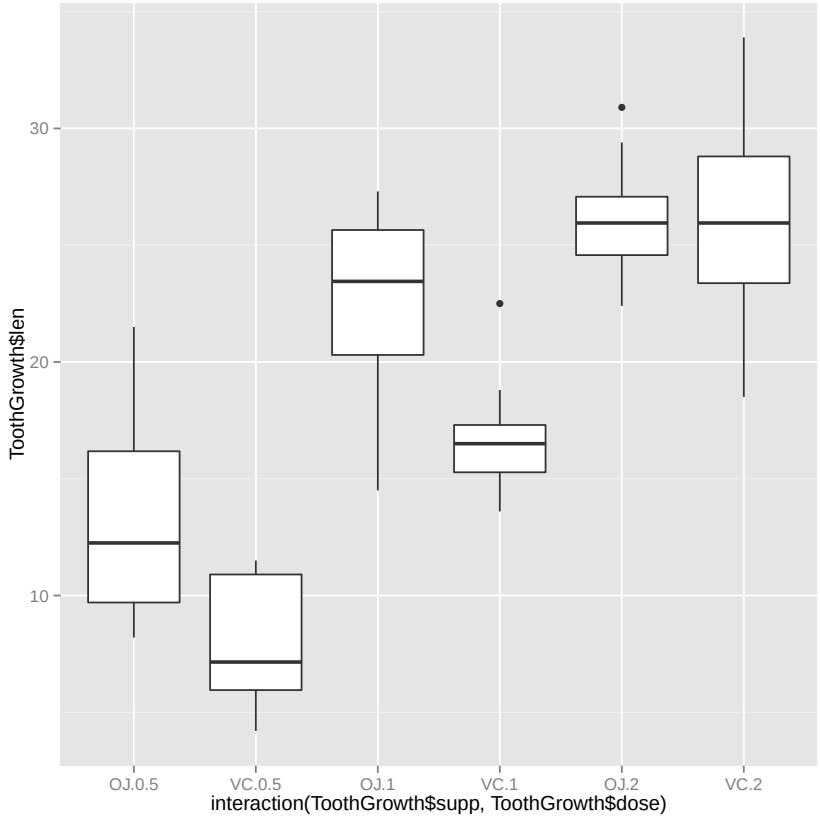
```
library(ggplot2)
qplot(supp, len, data=ToothGrowth, geom="boxplot")
```




```
#Alternative Solution
library(ggplot2)
ggplot(ToothGrowth, aes(x=supp, y=len)) + geom_boxplot()
```

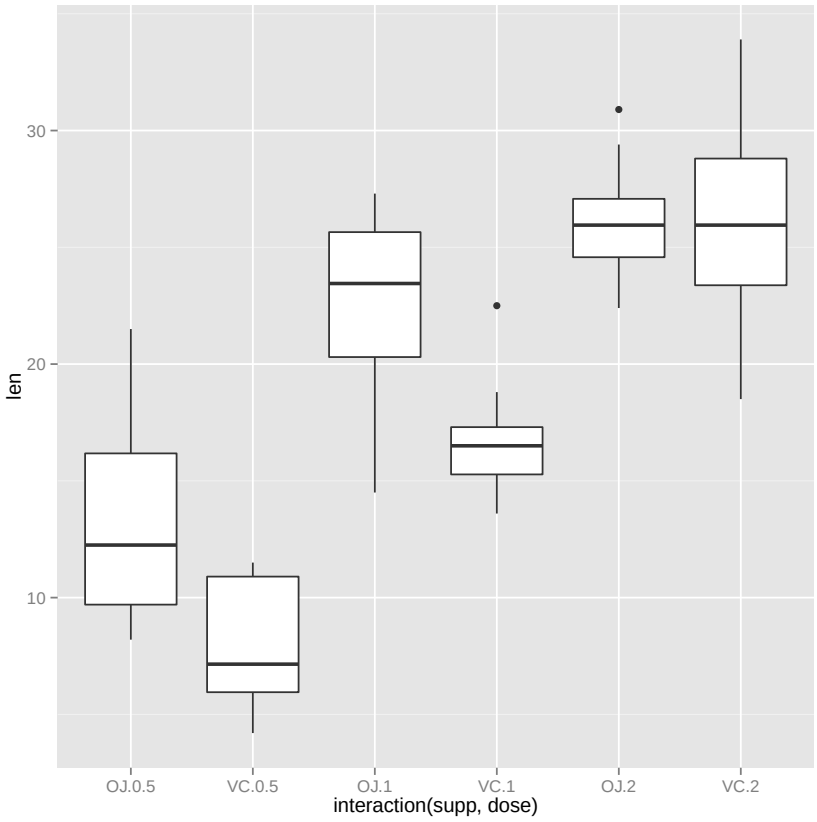
33. Load the ggplot2 package and produce the following plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

```
  library(ggplot2)
  qplot(interaction(ToothGrowth$supp, ToothGrowth$dose), ToothGrowth$len,
geom="boxplot")
```



34. Load the ggplot2 package and produce the following plot with the ToothGrowth dataset. It's built into R so you do not need to load any packages:

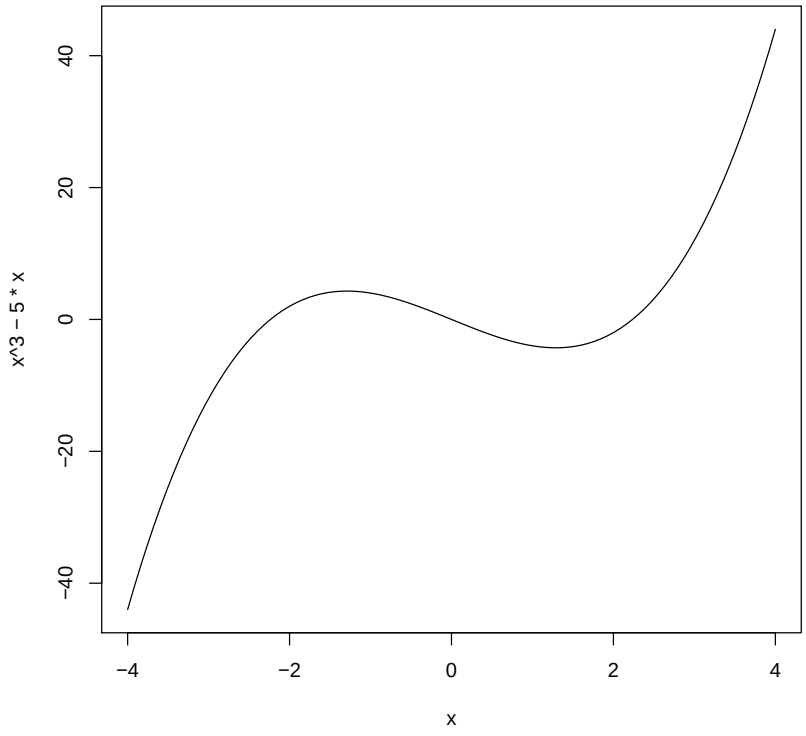
```
library(ggplot2)
qplot(interaction(supp, dose), len, data=ToothGrowth, geom="boxplot")
```




```
#Alternative Solution
library(ggplot2)
ggplot(ToothGrowth, aes(x=interaction(supp, dose), y=len)) + geom_boxplot()
```

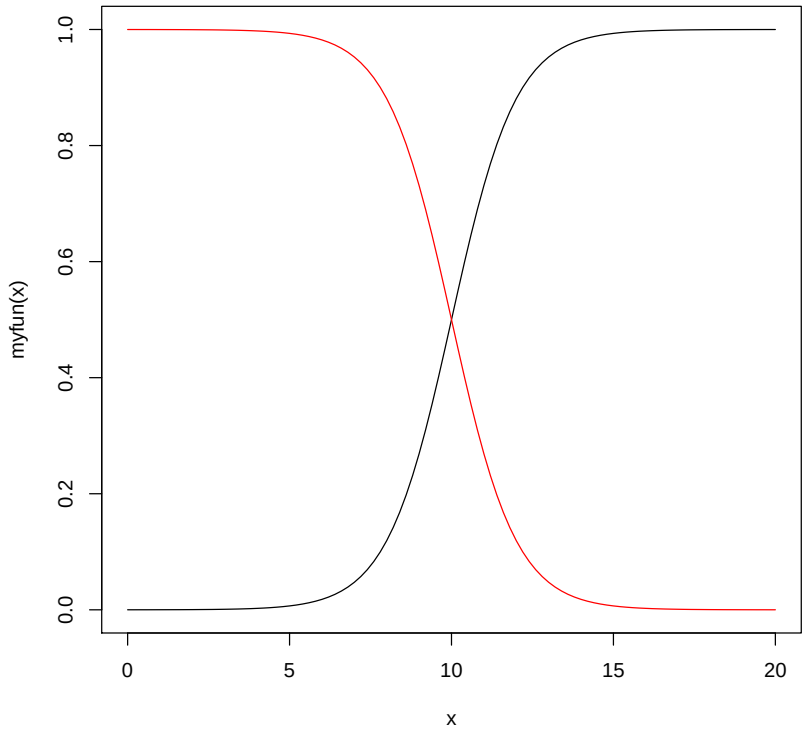
35. Plot the function $x^3 - 5x$ on the range $[-4, 4]$:

```
curve(x^3-5*x, from=-4, to=4)
```



36. Plot the curves $1/(1+\mathrm{e}^{-x+10})$ and $1-1/(1+\mathrm{e}^{-x+10})$ as shown below:

```
myfun <- function(xvar) {
  1/(1 + exp(-xvar + 10))
  }
curve(myfun(x), from=0, to=20)
curve(1-myfun(x), add = TRUE, col = "red")
```

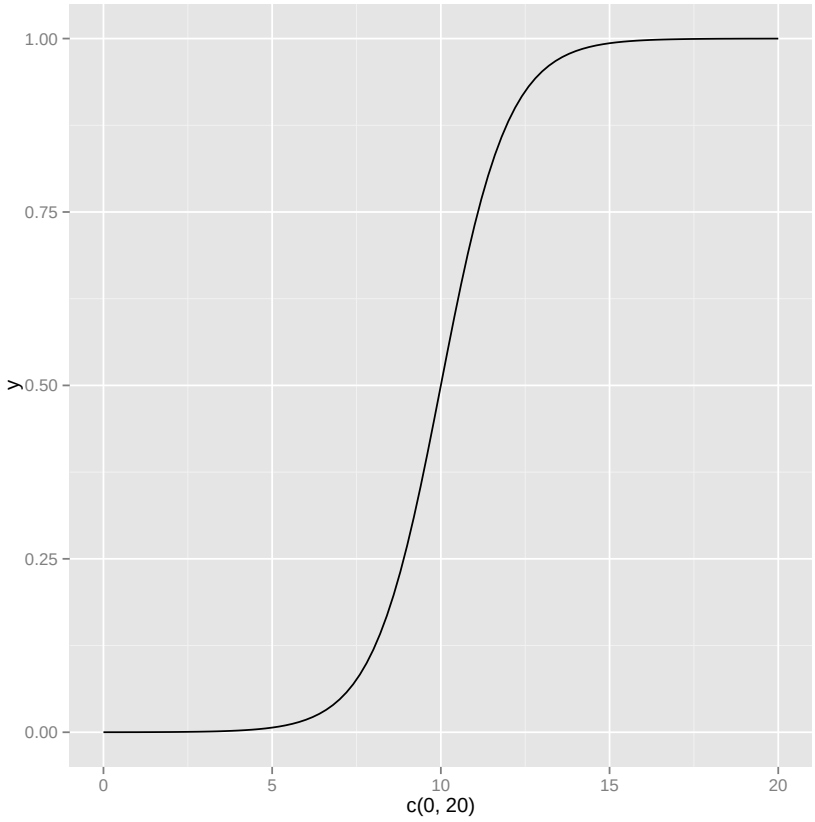


37. Run the following code below:

```
myfun <- function(xvar) {
  1/(1 + exp(-xvar + 10))
  }
```

This represents the function $f(x) = 1/(1 + e^{-x+10})$. Load the ggplot2 package and plot the function on the range $[0, 10]$:

```
qplot(c(0,20), fun=myfun, stat="function", geom="line")
```



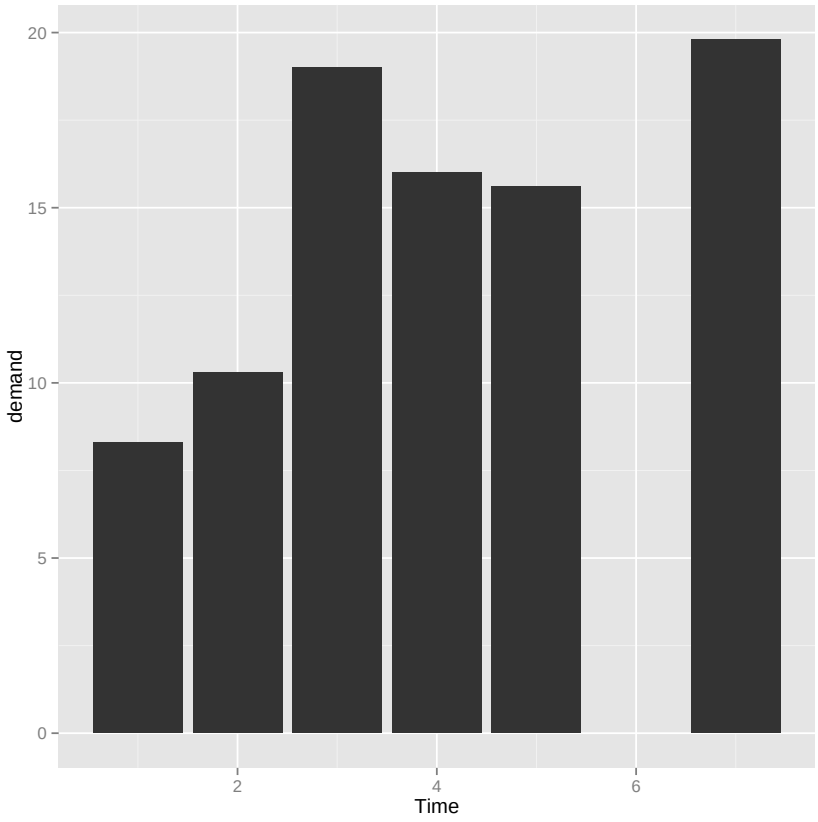

```
#Alternative solution
ggplot(data.frame(x=c(0, 20)), aes(x=x)) +
  stat_function(fun=myfun, geom="line")
```

Chapter 3

Bar Graphs

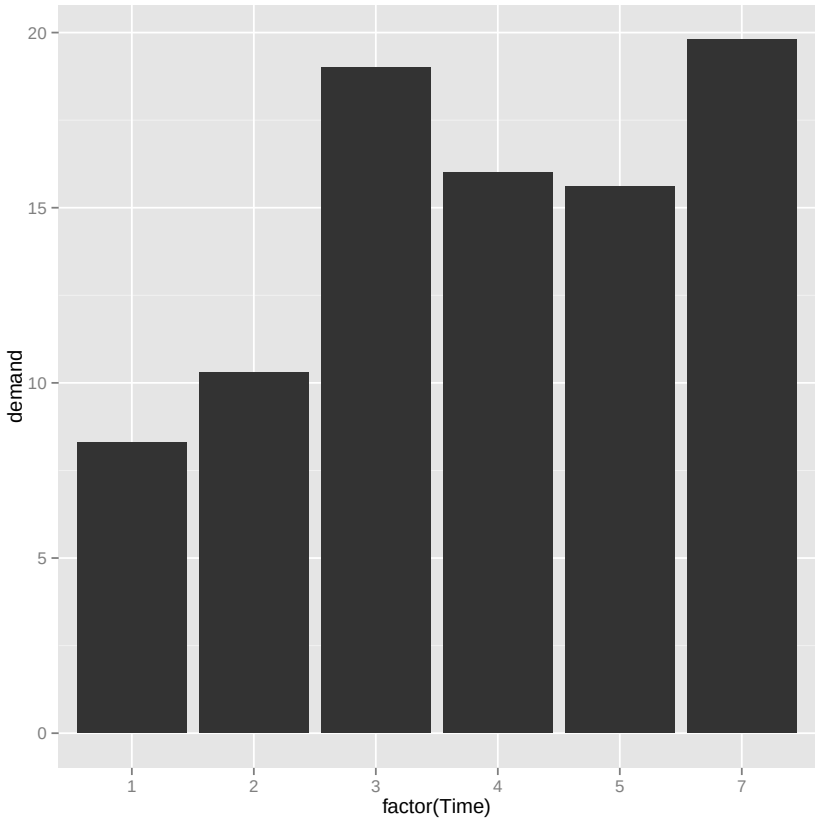
1. Load the ggplot2 package and produce the following plot with the BOD (automatically loaded with R) dataset:

```
ggplot(BOD, aes(x=Time, y=demand)) + geom_bar(stat="identity")
```



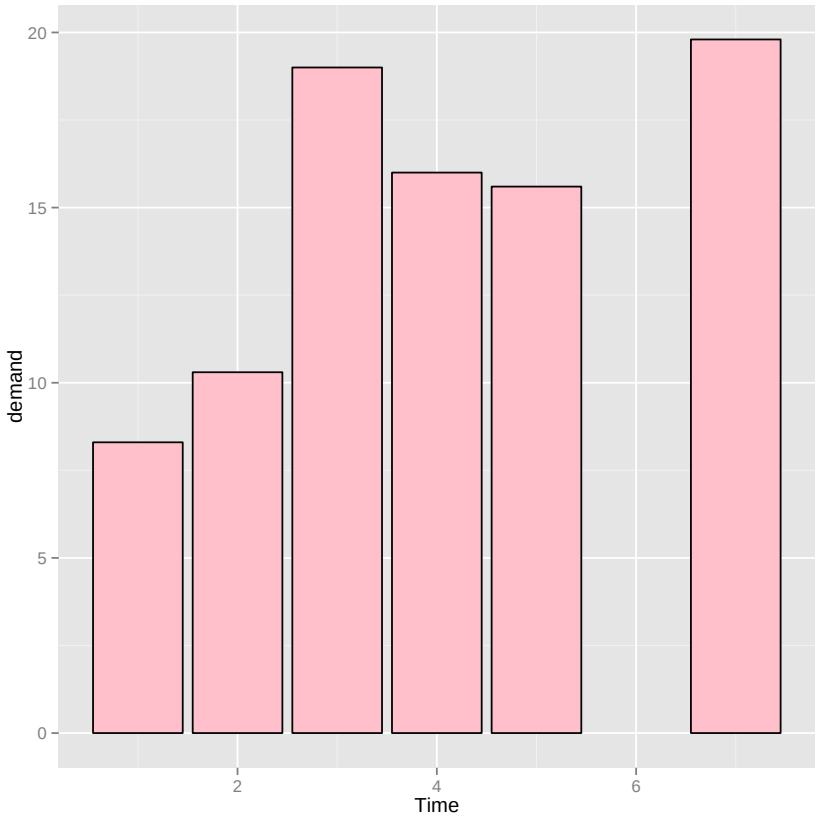
2. Load the ggplot2 package and produce the following plot with the BOD (automatically loaded with R) dataset:

```
ggplot(BOD, aes(x=factor(Time), y=demand)) + geom_bar(stat="identity")
```



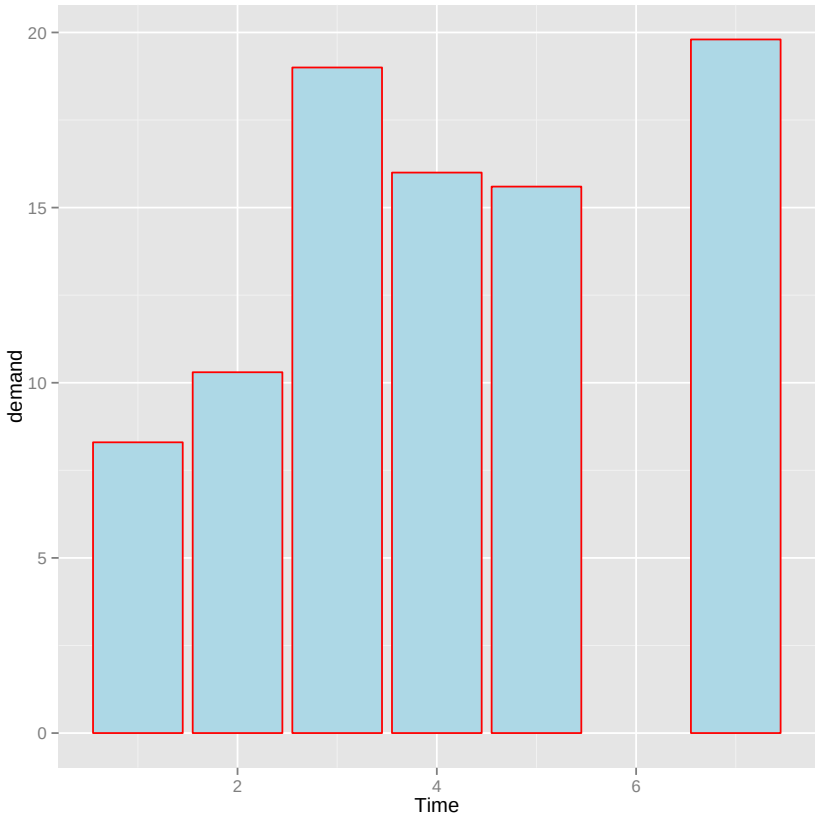
3. Load the ggplot2 package and produce the following plot with the BOD (automatically loaded with R) dataset:

```
ggplot(BOD,aes(x=Time, y=demand)) +
  geom_bar(stat="identity",fill="pink",colour="black")
```



4. Load the ggplot2 package and produce the following plot with the BOD (automatically loaded with R) dataset:

```
ggplot(BOD,aes(x=Time, y=demand)) +
  geom_bar(stat="identity",fill="lightblue",colour="red")
```

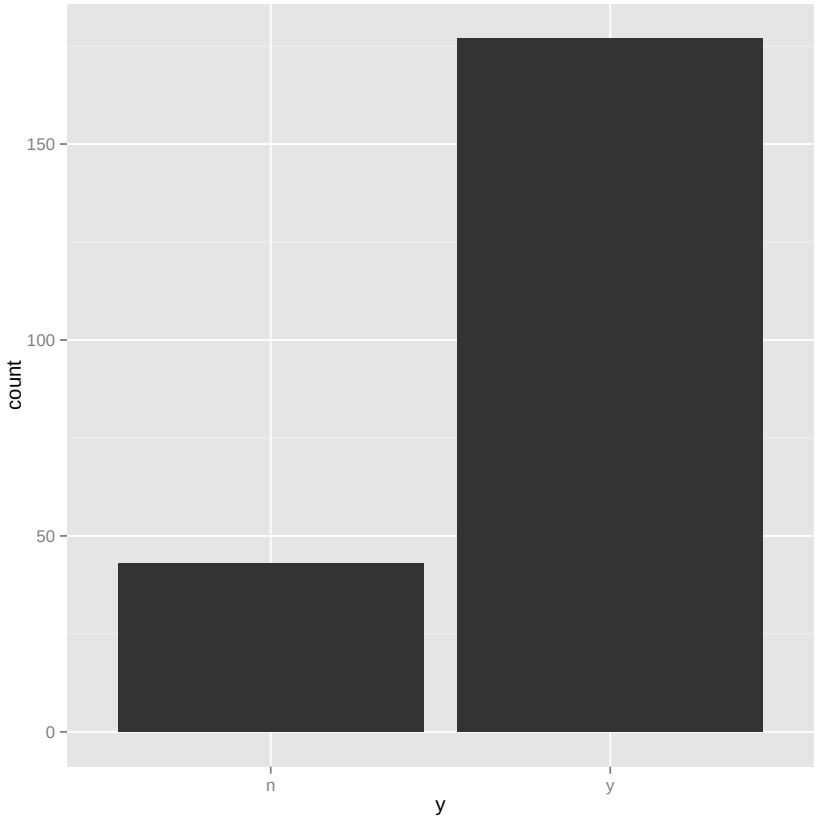


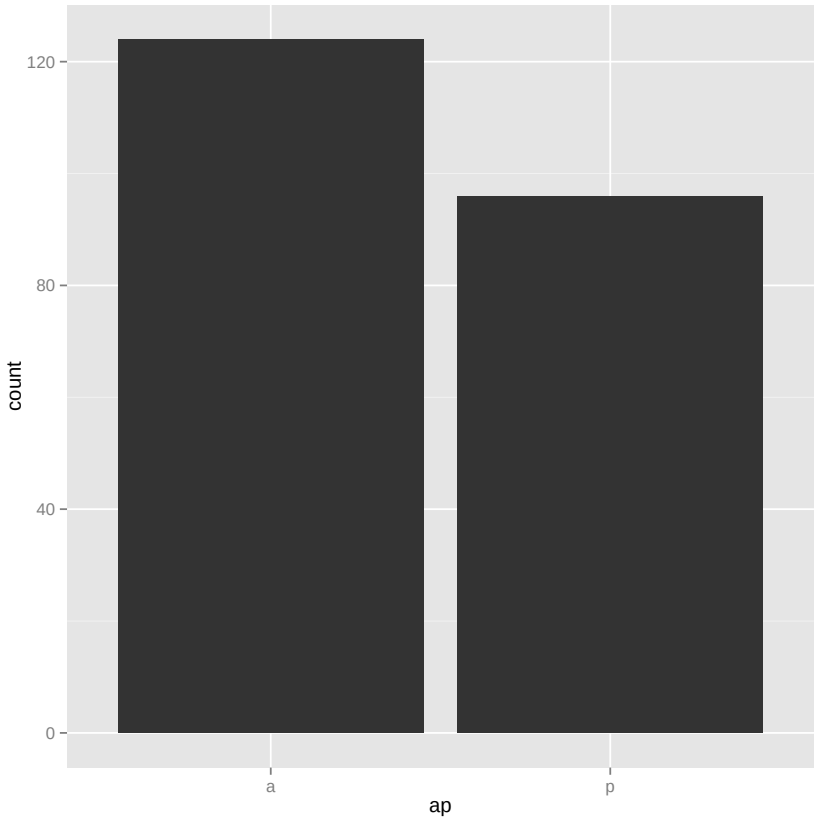
Chapter 4

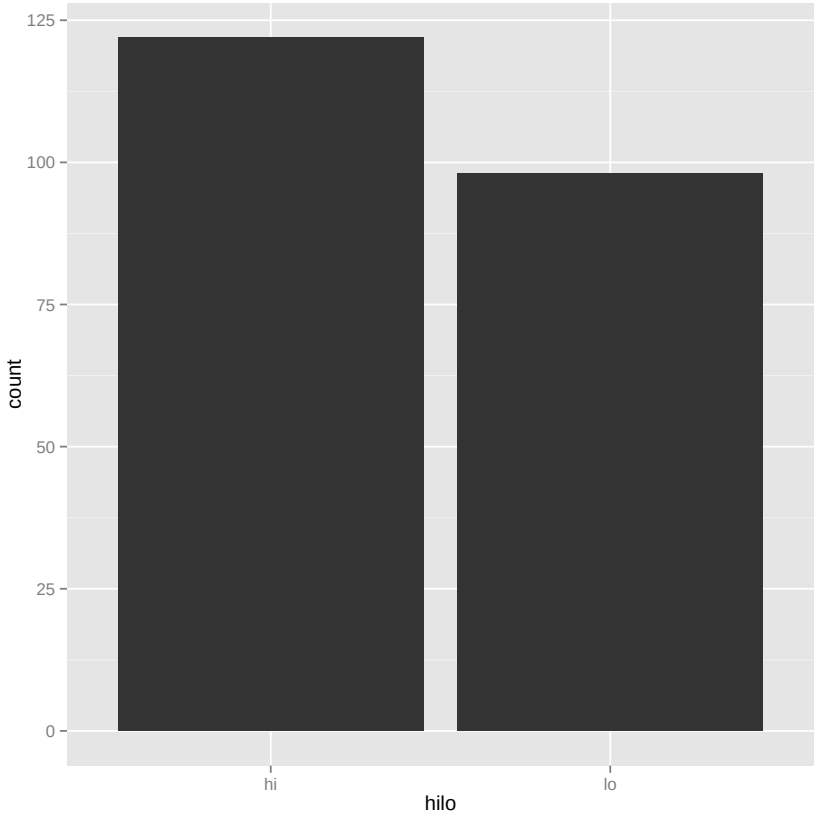
Miscellaneous

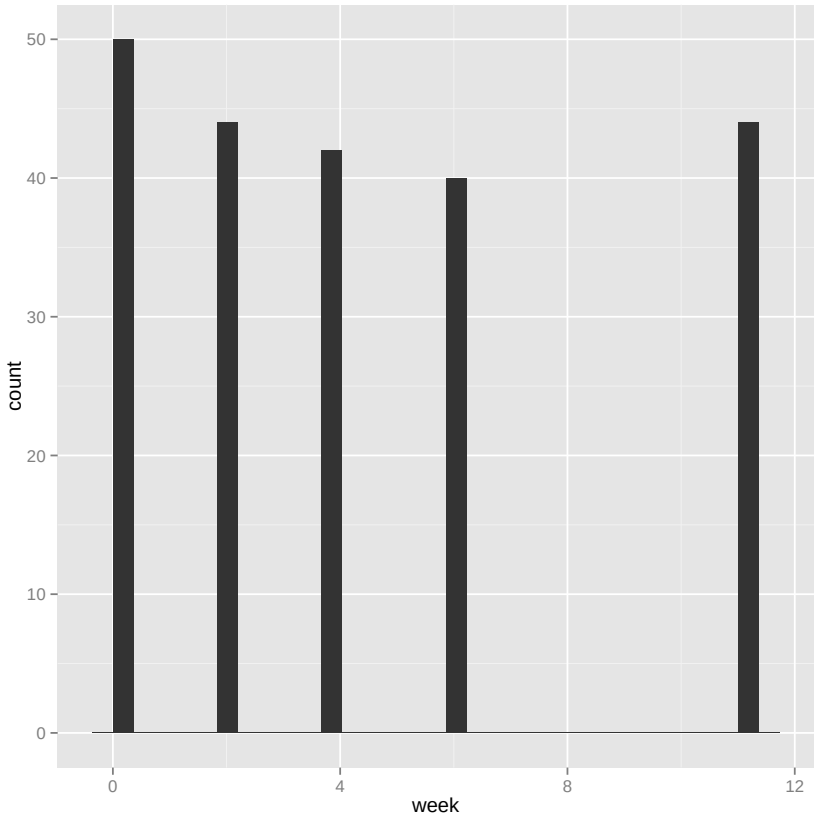
1. Load the package ggplot2 and produce the following 5 charts from the bacteria dataset from the MASS package using a single loop. Be sure to remove the ID variable from your dataset:

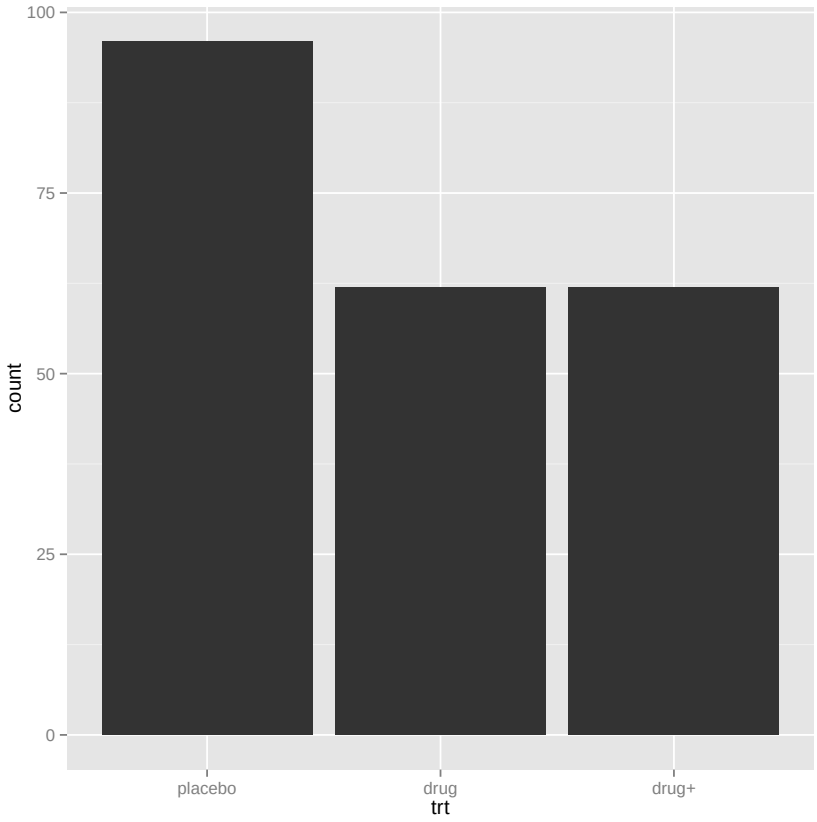
```
library(ggplot2)
library(MASS)

bacteria2 <- bacteria[which(names(bacteria) != "ID")]

for(i in 1:length(bacteria2)){
  print(ggplot(bacteria2,aes_string(x=names(bacteria2[i]))) +
          geom_histogram())
}
```




120
80
40
0
count
a
p
ap


125
100
75
50
25
0
count
hi
lo
hilo


50
40
30
20
10
0
count
0
4
8
12
week


100
75
50
25
0
count
placebo
drug
drug+
trt

Chapter 5

Actuarial Graphics