MATH 208 FINAL VERSION

Alyzeh Jiwani

03/10/2019

Q1

(a) Determine the (i) mode and (ii) class of the ToothGrowth data object

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.2

## Warning: package 'tibble' was built under R version 3.5.2

## Warning: package 'tidyr' was built under R version 3.5.2

## Warning: package 'purrr' was built under R version 3.5.2

## Warning: package 'dplyr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.2

## Warning: package 'forcats' was built under R version 3.5.2

## -- Conflicts ------------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data("ToothGrowth")

mode(ToothGrowth)
```

```
## [1] "list"
```

```
class(ToothGrowth)
```

```
## [1] "data.frame"
```

The mode of the ToothGrowth data object is a list, the class of the data object is data frame.

(b) Determine how many rows and columns the object has using R functions

```
nrow(ToothGrowth)
```

```
## [1] 60
```

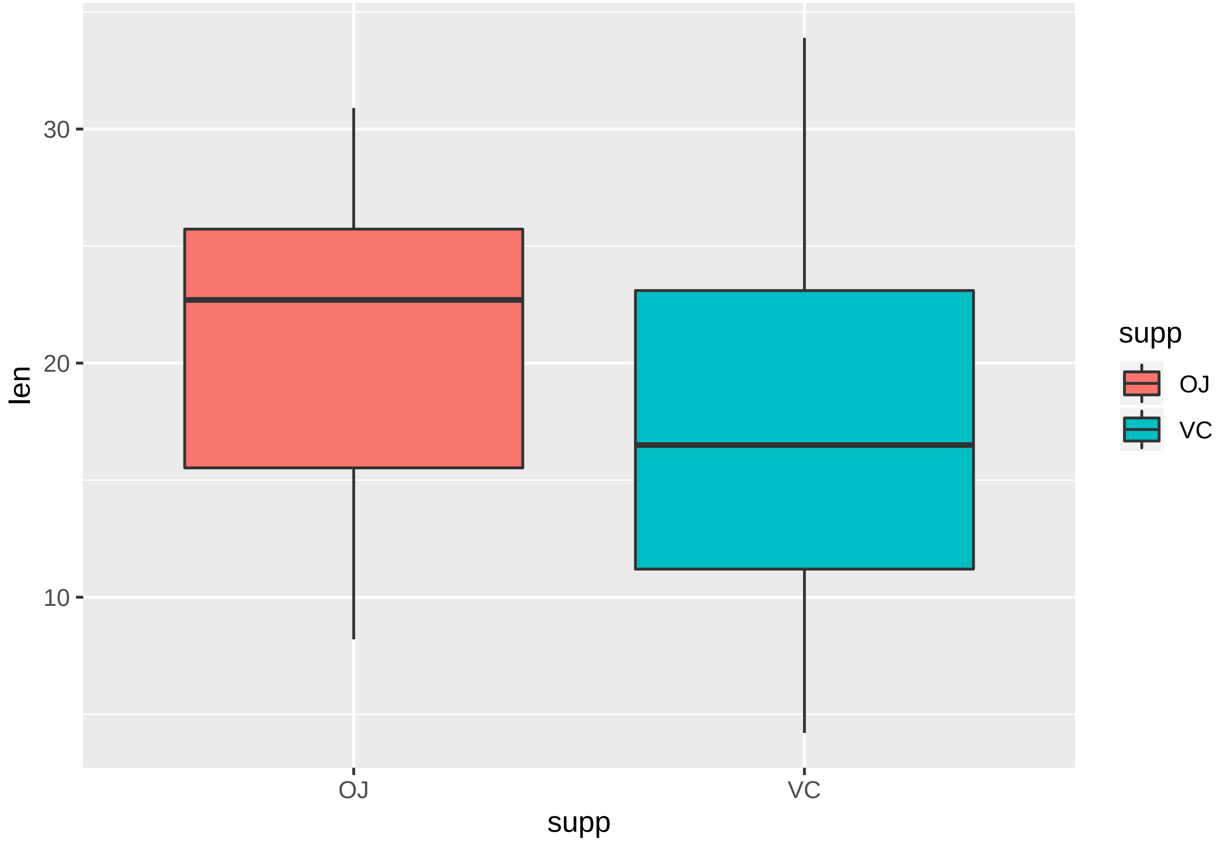
```
ncol(ToothGrowth)
```

```
## [1] 3
```

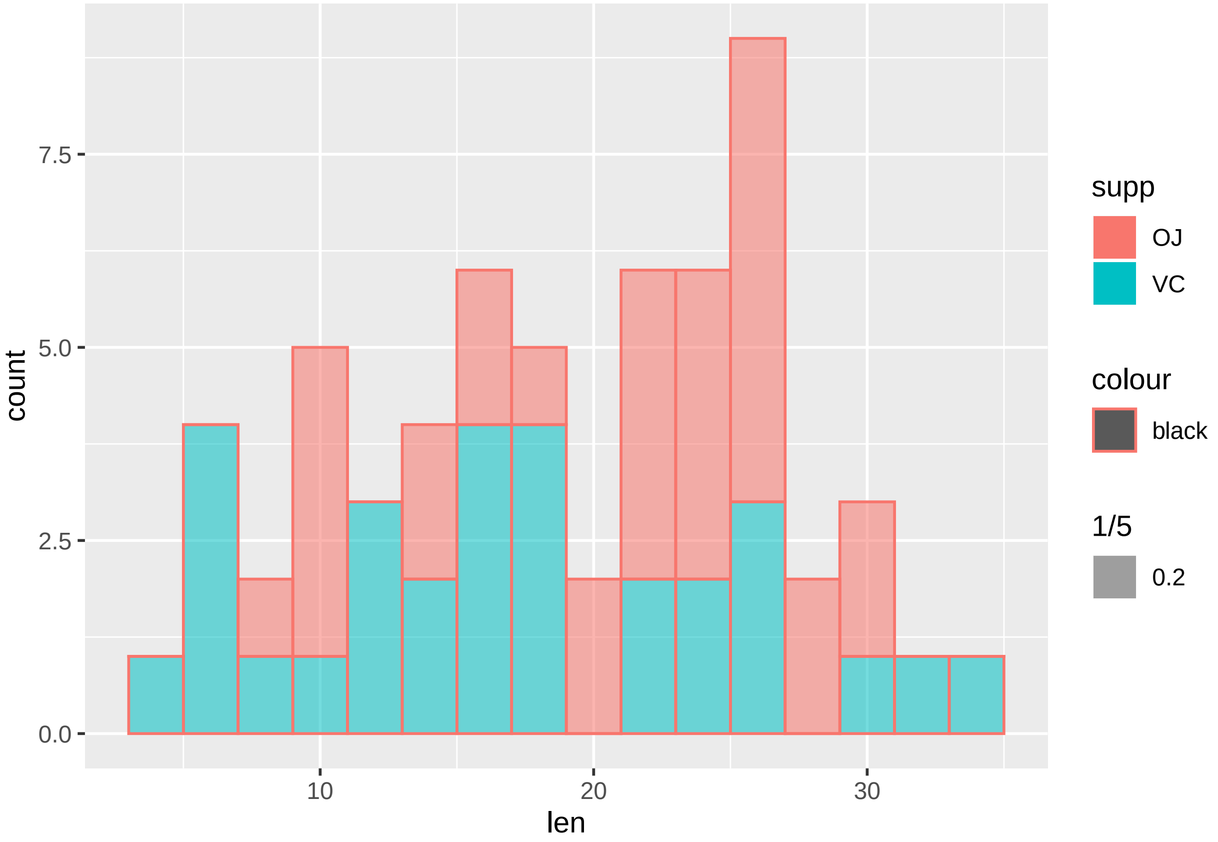
The ToothGrowth data object has 60 rows and 3 columns

(c) Use boxplots, histograms, and density plots to describe the distribution of odontoblast lengths by supplement type. Does one supplement seem to be associated with greater lengths? Explain your answer.

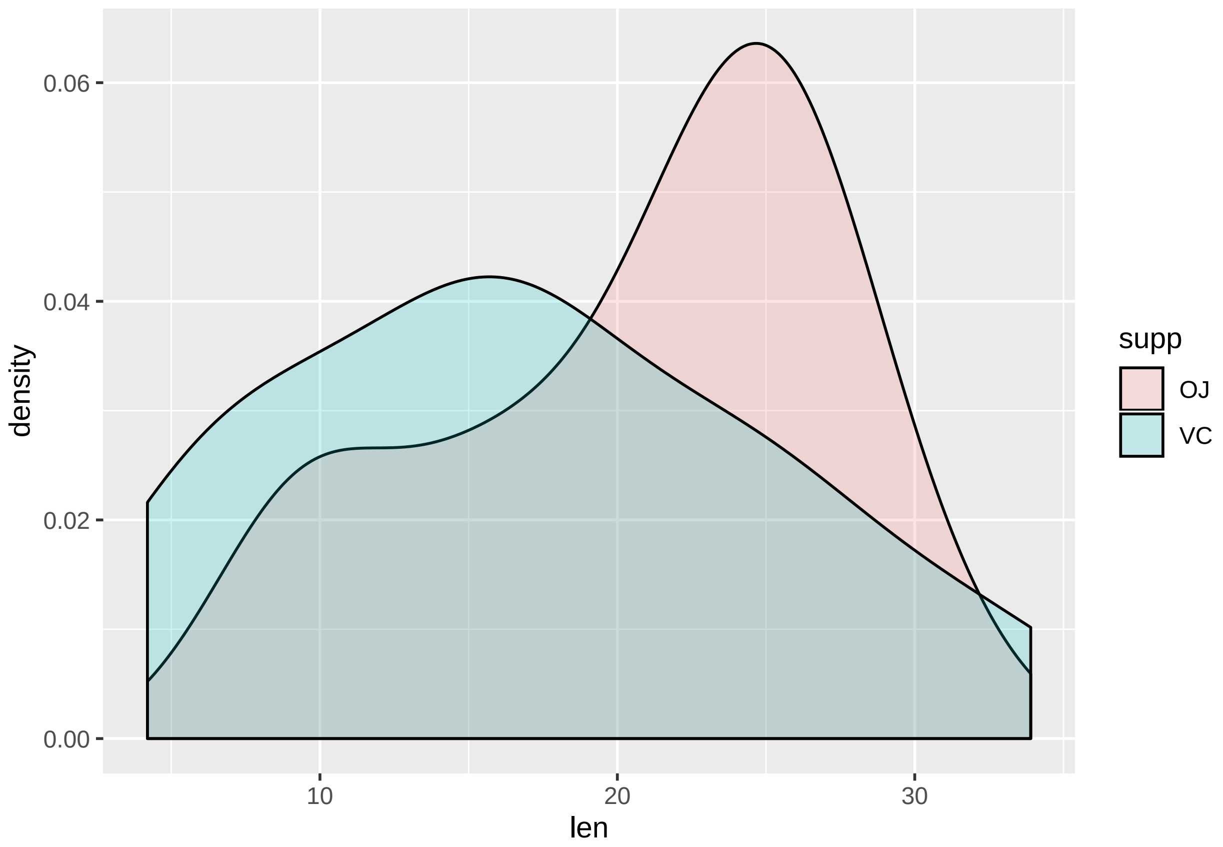
```
#Boxplot
ggplot(ToothGrowth, aes(x = supp, y=len, fill= supp))+
  geom_boxplot()
```



```
#Histogram
ggplot(ToothGrowth, aes(x=len, fill=supp, colour="black", alpha=1/5))+
  geom_histogram(binwidth=2)
```



```
#Density Plot
ggplot(ToothGrowth, aes(x=len, fill=supp))+
  geom_density(alpha=1/5)
```

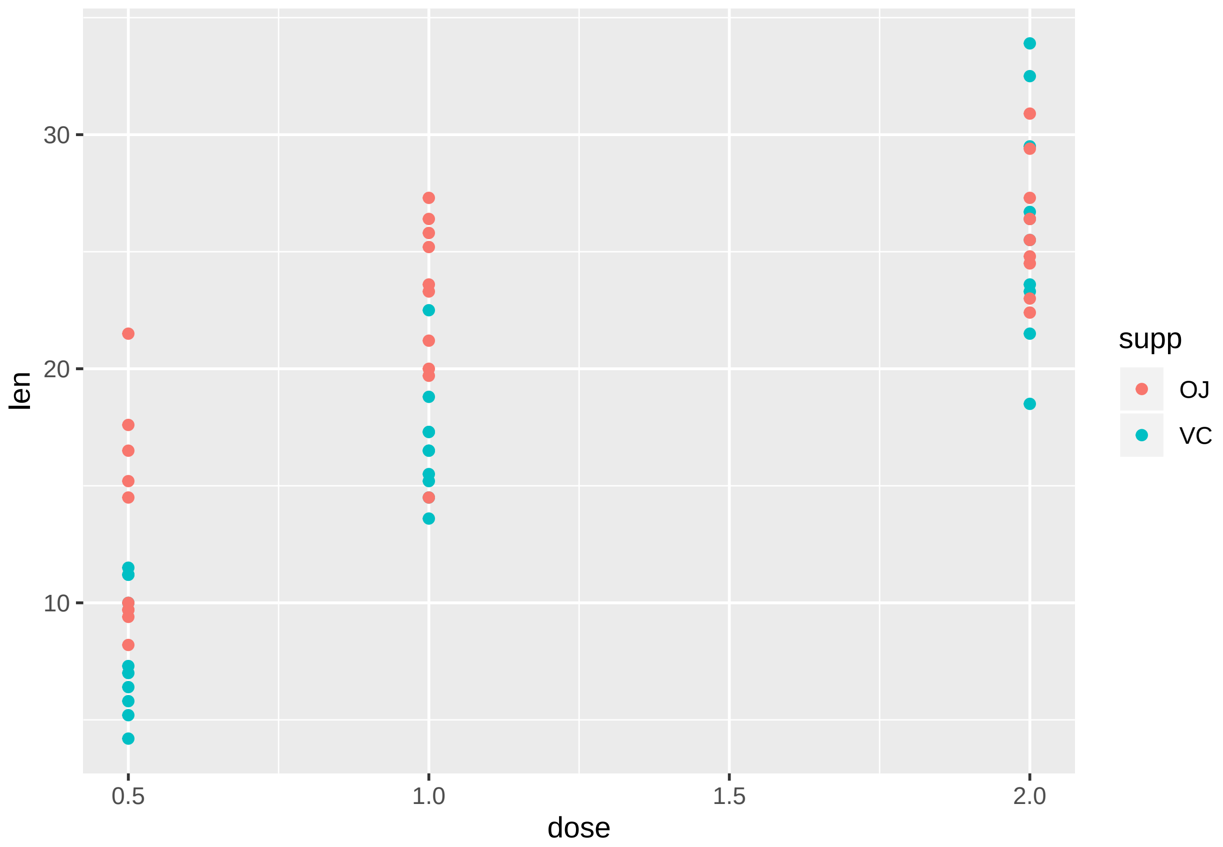


The OJ supplement appears to be associated with greater lengths. As seen in the density graph, length 24 has the highest density at above 0.06,likewise thius is also seen on the historgam where the count for length 24 is the higher than any other length on the graph.This is also demonstarted in the boxplots where the OJ supplement is seen to be clearly having a median that is significantly higher than that of the VC supplement. Moreover, from the boxplots we can see that the upper and lower quartiles of the OJ supplement surpass theur counterparts in the VC boxplot, despit the fact that the VC supplemnt has a higher upper bound. This indicates that the OJ supplement appears to be associated with higher lengths overall, even though the highest length itself is associated with the VC supplement.

(d) Based on your output from part (c), which plot do you think is most effective for assessing whether there is a difference in distribution of lengths between the two groups? Explain your answer. I think that the density plot is most effective for assessing whether there is a difference in the distribution of lengths between the two groups as it can be clearly seen that there is a difference in the medians of the two supplements and that the spread of their respective datas also differ, VC is more spread out whereas OJ is more concentrated around the medians.

(e) Create an appropriate scatterplot to assess the association between the dose of the supplement and the lengths and to determine whether the nature of the association depends on the type of supplement. Does the association between length and dose seem to depend on the type of supplement? Explain your answer.

```
#Scatterplot
 ggplot(ToothGrowth, aes(x=dose, y=len, colour=supp))+
  geom_point()
```



The chart suggest that the association between length and dose does seem to depend on supplement type. As seen on the plot, for the lower doses OJ presents the higher lengths, whereas once the dose reaches 2.0 VC and OJ have similar lengths, with VC having the highest ones. One could hypothesise that as the dose gets higher, the efficacy of the OJ supplement plateaus, whilst that of the VC supplement continues to increase.

(f) Generate a summary table that contains the mean, median, and standard deviation of the lengths for each supplement type

```
(ToothGrowth_Summary <- ToothGrowth %>% group_by(supp) %>%
  summarise(Mean = mean(len),
            Median = median(len),
            Standard_Deviation = sd(len)))
```

```
## # A tibble: 2 x 4
##   supp   Mean Median Standard_Deviation
##   <fct> <dbl>  <dbl>              <dbl>
## 1 OJ     20.7   22.7               6.61
## 2 VC     17.0   16.5               8.27
```

Q2

(a) Read in the data directly to a tibble object from the URL by using the read_csv() function

```
(Abalone <- as_tibble(read.csv(url("https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/al
```

```
## # A tibble: 4,176 x 9
##    M     X0.455 X0.365 X0.095 X0.514 X0.2245 X0.101 X0.15   X15
##    <fct>  <dbl>  <dbl>  <dbl>  <dbl>   <dbl>  <dbl> <dbl> <int>
##  1 M      0.35   0.265  0.09   0.226  0.0995 0.0485 0.07      7
##  2 F      0.53   0.42   0.135  0.677  0.256  0.142  0.21      9
##  3 M      0.44   0.365  0.125  0.516  0.216  0.114  0.155    10
##  4 I      0.33   0.255  0.08   0.205  0.0895 0.0395 0.055     7
##  5 I      0.425  0.3    0.095  0.352  0.141  0.0775 0.12      8
##  6 F      0.53   0.415  0.15   0.778  0.237  0.142  0.33     20
##  7 F      0.545  0.425  0.125  0.768  0.294  0.150  0.26     16
##  8 M      0.475  0.37   0.125  0.509  0.216  0.112  0.165     9
##  9 F      0.55   0.44   0.15   0.894  0.314  0.151  0.32     19
## 10 F      0.525  0.38   0.14   0.606  0.194  0.148  0.21     14
## # ... with 4,166 more rows
```

(b) Assign names to the columns of the tibble. The columns are in order of the measuremnts given in the table.

```
(Abalone_1 <- Abalone %>%
   rename(
     Sex = M,
     Length = X0.455,
     Diameter = X0.365,
     Height = X0.095,
     Whole_Weight = X0.514,
     Shucked_Weight = X0.2245,
     Viscera_Weight = X0.101,
     Shell_Weight = X0.15,
     Rings = X15))
```

```
## # A tibble: 4,176 x 9
##    Sex   Length Diameter Height Whole_Weight Shucked_Weight Viscera_Weight
##    <fct>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 M      0.35     0.265  0.09         0.226         0.0995         0.0485
##  2 F      0.53     0.42   0.135        0.677         0.256          0.142
##  3 M      0.44     0.365  0.125        0.516         0.216          0.114
##  4 I      0.33     0.255  0.08         0.205         0.0895         0.0395
##  5 I      0.425    0.3    0.095        0.352         0.141          0.0775
##  6 F      0.53     0.415  0.15         0.778         0.237          0.142
##  7 F      0.545    0.425  0.125        0.768         0.294          0.150
##  8 M      0.475    0.37   0.125        0.509         0.216          0.112
##  9 F      0.55     0.44   0.15         0.894         0.314          0.151
## 10 F      0.525    0.38   0.14         0.606         0.194          0.148
## # ... with 4,166 more rows, and 2 more variables: Shell_Weight <dbl>,
## #   Rings <int>
```

```
##The headings of the original tibble were data, which have now been replaced by new headings
##I will add a new row to the Abalone data frame containing this deleted data

Abalone_2 <- add_row(Abalone_1, Sex = "M", Length = 0.455, Diameter = 0.365, Height = 0.095, Whole_Weig
```

(c) Create a new column for the radius of the abalone shell by using the diameter

```
(Abalone_3 <- mutate(Abalone_2, Radius = Diameter/2))
```

```
## # A tibble: 4,177 x 10
##    Sex   Length Diameter Height Whole_Weight Shucked_Weight Viscera_Weight
##    <fct>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 M      0.35     0.265  0.09         0.226         0.0995         0.0485
##  2 F      0.53     0.42   0.135        0.677         0.256          0.142
##  3 M      0.44     0.365  0.125        0.516         0.216          0.114
##  4 I      0.33     0.255  0.08         0.205         0.0895         0.0395
##  5 I      0.425    0.3    0.095        0.352         0.141          0.0775
##  6 F      0.53     0.415  0.15         0.778         0.237          0.142
##  7 F      0.545    0.425  0.125        0.768         0.294          0.150
##  8 M      0.475    0.37   0.125        0.509         0.216          0.112
##  9 F      0.55     0.44   0.15         0.894         0.314          0.151
## 10 F      0.525    0.38   0.14         0.606         0.194          0.148
## # ... with 4,167 more rows, and 3 more variables: Shell_Weight <dbl>,
## #   Rings <dbl>, Radius <dbl>
```

(d) Find the maximum and minimum number of rings for each value of the Sex variable by using R functions

```
##First I seperate the data set into M, F, and I
(Abalone_M <- filter(Abalone_3, Sex == "M"))
```

```
## # A tibble: 1,528 x 10
##    Sex   Length Diameter Height Whole_Weight Shucked_Weight Viscera_Weight
##    <fct>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 M      0.35     0.265  0.09         0.226         0.0995         0.0485
##  2 M      0.44     0.365  0.125        0.516         0.216          0.114
##  3 M      0.475    0.37   0.125        0.509         0.216          0.112
##  4 M      0.43     0.35   0.11         0.406         0.168          0.081
##  5 M      0.49     0.38   0.135        0.542         0.218          0.095
##  6 M      0.5      0.4    0.13         0.664         0.258          0.133
##  7 M      0.365    0.295  0.08         0.256         0.097          0.043
##  8 M      0.45     0.32   0.1          0.381         0.170          0.075
##  9 M      0.355    0.28   0.095        0.246         0.0955         0.062
## 10 M      0.59     0.445  0.14         0.931         0.356          0.234
## # ... with 1,518 more rows, and 3 more variables: Shell_Weight <dbl>,
## #   Rings <dbl>, Radius <dbl>
```

```
(Abalone_F <- filter(Abalone_3, Sex == "F"))
```

```
## # A tibble: 1,307 x 10
##    Sex   Length Diameter Height Whole_Weight Shucked_Weight Viscera_Weight
##    <fct>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 F      0.53     0.42   0.135        0.677          0.256         0.142
##  2 F      0.53     0.415  0.15         0.778          0.237         0.142
##  3 F      0.545    0.425  0.125        0.768          0.294         0.150
##  4 F      0.55     0.44   0.15         0.894          0.314         0.151
##  5 F      0.525    0.38   0.14         0.606          0.194         0.148
##  6 F      0.535    0.405  0.145         0.684          0.272         0.171
##  7 F      0.47     0.355  0.1           0.476          0.168         0.0805
##  8 F      0.44     0.34   0.1           0.451          0.188         0.087
##  9 F      0.565    0.44   0.155         0.940          0.428         0.214
## 10 F      0.55     0.415  0.135         0.764          0.318         0.21
## # ... with 1,297 more rows, and 3 more variables: Shell_Weight <dbl>,
## #   Rings <dbl>, Radius <dbl>
```

```
(Abalone_I <- filter(Abalone_3, Sex == "I"))
```

```
## # A tibble: 1,342 x 10
##    Sex   Length Diameter Height Whole_Weight Shucked_Weight Viscera_Weight
##    <fct>  <dbl>    <dbl>  <dbl>        <dbl>          <dbl>          <dbl>
##  1 I      0.33     0.255  0.08          0.205         0.0895         0.0395
##  2 I      0.425    0.3    0.095         0.352         0.141          0.0775
##  3 I      0.355    0.28   0.085         0.290         0.095          0.0395
##  4 I      0.38     0.275  0.1           0.226         0.08           0.049
##  5 I      0.24     0.175  0.045         0.07          0.0315         0.0235
##  6 I      0.205    0.15   0.055         0.042         0.0255         0.015
##  7 I      0.21     0.15   0.05          0.042         0.0175         0.0125
##  8 I      0.39     0.295  0.095         0.203         0.0875         0.045
##  9 I      0.325    0.245  0.07          0.161         0.0755         0.0255
## 10 I      0.52     0.41   0.12          0.595         0.238          0.111
## # ... with 1,332 more rows, and 3 more variables: Shell_Weight <dbl>,
## #   Rings <dbl>, Radius <dbl>
```

```
##Then I select the Rings column from each new data frame
(Abalone_M_Rings <- select(Abalone_M, Rings))
```

```
## # A tibble: 1,528 x 1
##    Rings
##    <dbl>
##  1     7
##  2    10
##  3     9
##  4    10
##  5    11
##  6    12
##  7     7
##  8     9
##  9    11
## 10    12
## # ... with 1,518 more rows
```

```
(Abalone_F_Rings <- select(Abalone_F, Rings))
```

```
## # A tibble: 1,307 x 1
##    Rings
##    <dbl>
##  1     9
##  2    20
##  3    16
##  4    19
##  5    14
##  6    10
##  7    10
##  8    10
##  9    12
## 10     9
## # ... with 1,297 more rows
```

```
(Abalone_I_Rings <- select(Abalone_I, Rings))
```

```
## # A tibble: 1,342 x 1
##    Rings
##    <dbl>
##  1     7
##  2     8
##  3     7
##  4    10
##  5     5
##  6     5
##  7     4
##  8     7
##  9     6
## 10     8
## # ... with 1,332 more rows
```

```
##Now I can find the maximum and minimum number of rings for each value of the Sex variable
max(Abalone_M_Rings)
```

```
## [1] 27
```

```
min(Abalone_M_Rings)
```

```
## [1] 3
```

```
max(Abalone_F_Rings)
```

```
## [1] 29
```

```
min(Abalone_F_Rings)
```

```
## [1] 5
```

```
max(Abalone_I_Rings)
```

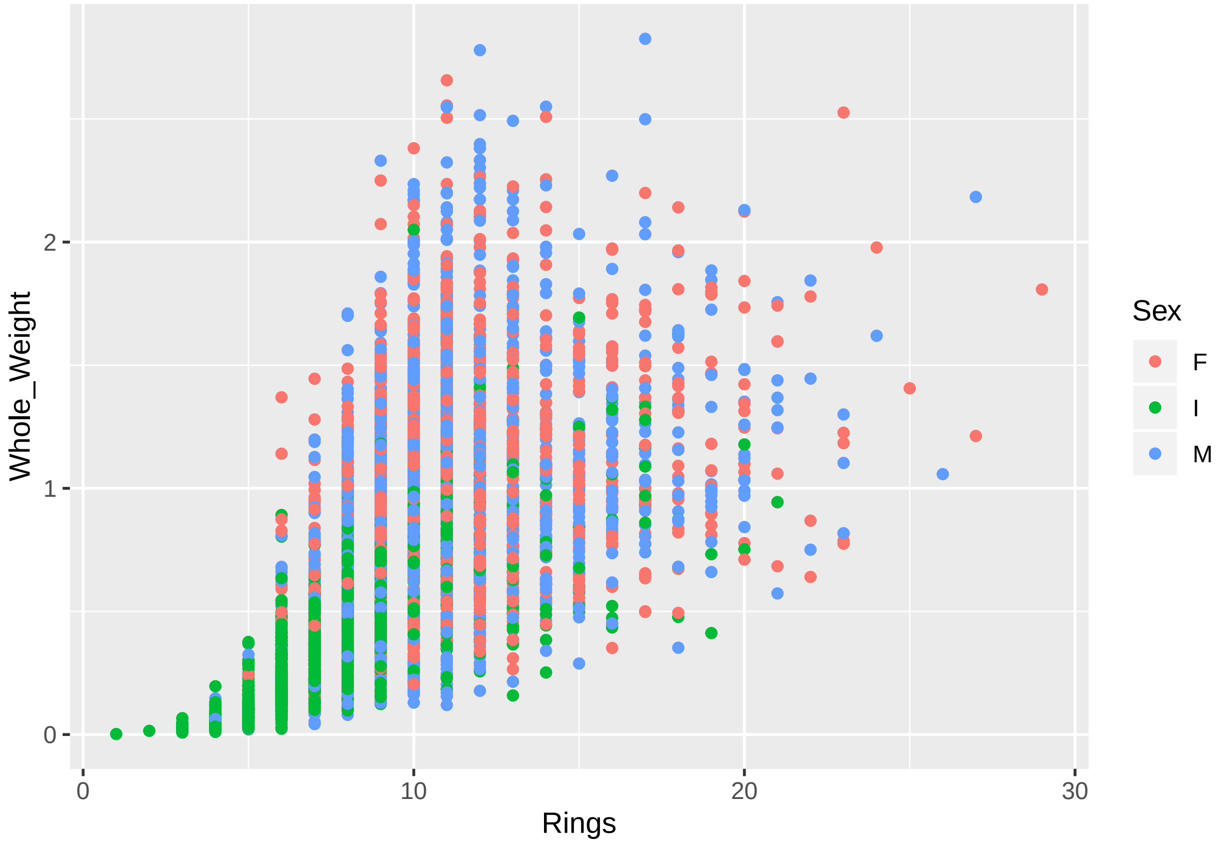
```
## [1] 21
```

```
min(Abalone_I_Rings)
```

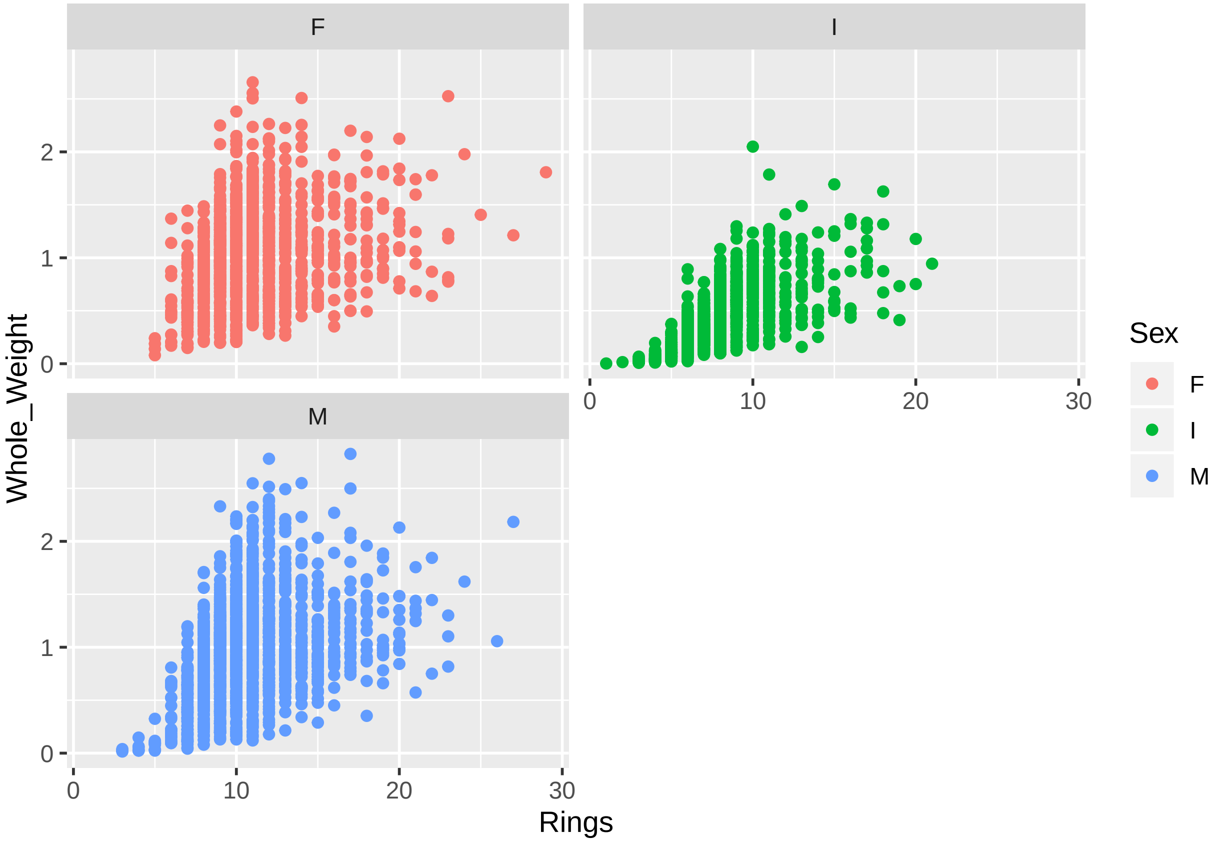
```
## [1] 1
```

(e) Using only plots, explain whther you think the association between total weight and the number of rings depends on the value for Sex

```
##First we draw a scatterplots

##M, F, and I on the same axis

ggplot(Abalone_3, aes(x=Rings, y=Whole_Weight, colour=Sex))+
  geom_point()
```

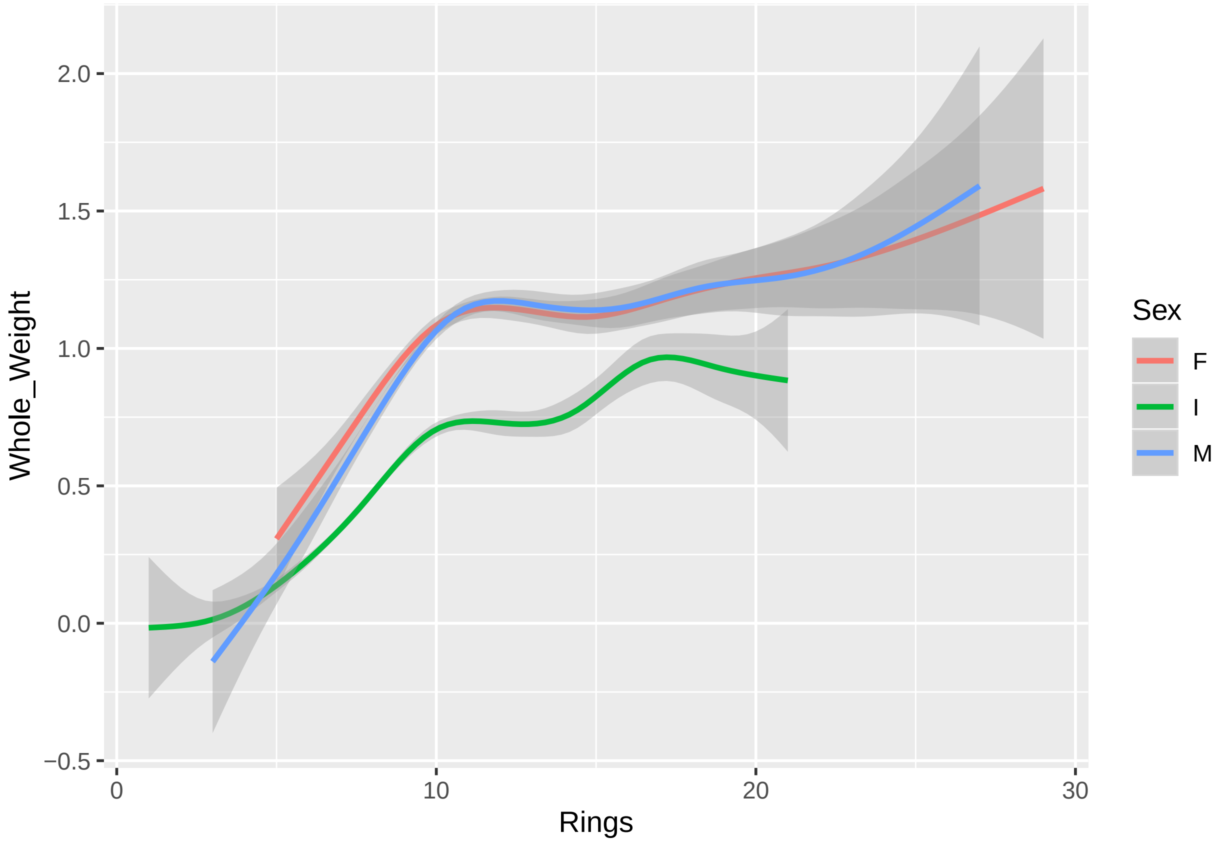


```
##M, F, and I on different axes

ggplot(Abalone_3, aes(x=Rings, y=Whole_Weight, colour=Sex))+
  geom_point()+
  facet_wrap(~Sex, nrow=2)
```



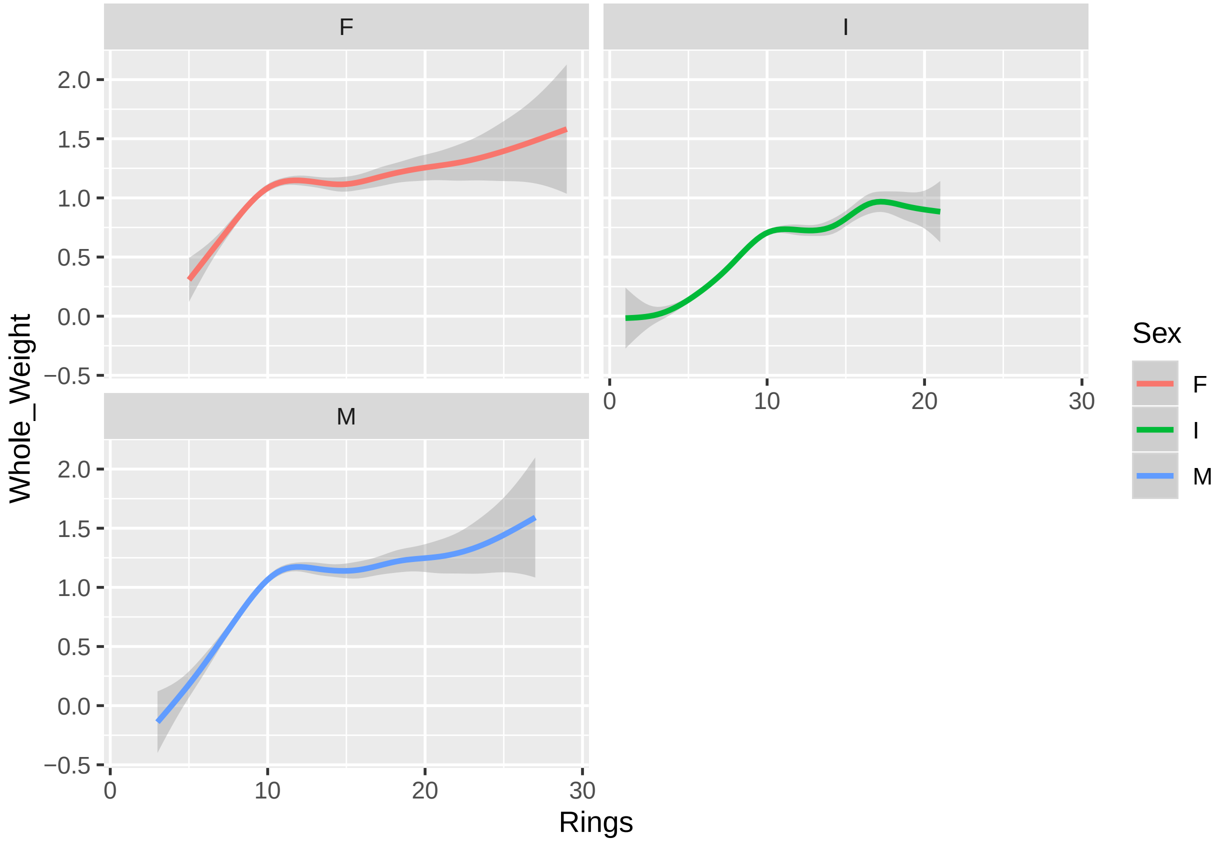
```r
##Now we have bar charts

##Same axis

ggplot(Abalone_3, aes(x=Rings, y=Whole_Weight, colour=Sex))+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
##Different axes

ggplot(Abalone_3, aes(x=Rings, y=Whole_Weight, colour=Sex))+
  geom_smooth()+
  facet_wrap(~Sex, nrow=2)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



From the data, it seems that for Sexes F and M, the no. of Rings and total weight appear to have the same spread of data, i.e. there doesnt appear to be any differences caused by Sex. However for Sex I there appears to be a significant difference, the number of Rings is smaller, as is the total weight. Thus, from the appearance of the data, it can be suggested that the no. of Rings and total weight depend on whether Sex = I.

Q3

(a) What objects (or values) are returned by the following lines of R code?

```
shopping_list <- list(
  Grocery = list(
    Dairy = c("Milk","Cheese"),
    Meat = c("Chicken","Sausage","Bacon"),
    Spices = c("Cinnamon")
    ),
  Pharmacy = c("Soap","Toothpaste","Toilet Paper")
  )

(shopping_list$Pharmacy)
```

```
## [1] "Soap"         "Toothpaste"   "Toilet Paper"
```

```
##Returns the values "Soap" "Toothpaste" "Toilet Paper"
## (shopping_list[1][[2]])
##Doesnt return any values as the subscript is out of bounds
(shopping_list[[1]][[3]])
```

```
## [1] "Cinnamon"
```

```
##Returns "Cinnamon"
(shopping_list[2][1])
```

```
## $Pharmacy
## [1] "Soap"         "Toothpaste"   "Toilet Paper"
```

```
##Returns $Pharmacy [1]"Soap" "Toothpaste" "Toilet Paper"
```

(b) Using R code, show which statement yields the following three results:

Result 1: [1]“Soap” “Toothpaste” “Toilet Paper”

```
shopping_list[2][[1]]
```

```
## [1] "Soap"         "Toothpaste"   "Toilet Paper"
```

Result 2: $Pharmacy [1]“Soap” “Toothpaste” “Toilet Paper”

```
shopping_list[2][1]
```

```
## $Pharmacy
## [1] "Soap"         "Toothpaste"   "Toilet Paper"
```

Result 3: [1]“Sausage”

```
shopping_list[[1]][[2]][[2]]
```

```
## [1] "Sausage"
```