Programming in R: Solutions for the Exam-part 1 (28/01/2022)

THERESSE JOY VILLARIEZ CALO (DL-2022)

---
output: html_document
use_bookdown: TRUE

Introduction

You can use this file to write your answers to the exam's queations. You need to submit the solution on:

- Date: 28/01/2022.
- Time: 17:00.

Part 1: the nassCDS data

In this part of the exam, we focus on the nassCDS data which is a US data from police-reported car crashes (1997-2002) in which there is a harmful event (people or property). Data are restricted to front-seat occupants, include only a subset of the variables recorded. More information about the dataset can be found using the following link: https://www.rdocumentation.org/packages/DAAG/versions/1.22/topics/nassCDS. The data is a part of the DAAG R package. To get an access to the data you first need to install the package.

```
library("DAAG")
data(nassCDS)
names(nassCDS)
```

```
##  [1] "dvcat"       "weight"      "dead"        "airbag"      "seatbelt"
##  [6] "frontal"     "sex"         "ageOFocc"    "yearacc"     "yearVeh"
## [11] "abcat"       "occRole"     "deploy"      "injSeverity" "caseid"
```

Question 1

In this question we use the nassCDS dataset and focus on the accident's outcome (the variable dead) and seatbelt usage (the variable seatbelt).

1. How many individuals used seatbelt?
2. What is the distribution of seatbelt usage across the accident's outcome factor ? Produce a 2X2 table that shows the number of seatbelt users (belted/none) and accident's outcome (alive/dead)?
3. Write a function that can be used to conduct inference for proportions in two independent populations. The null hypothesis is that there is no difference between the proportions in the two populations. Test the null hypothesis against a two sided alternative. The input of the function should be the 2X2 table in the previous item (Question 1.2) and the output should be the test statistic and the p value. Apply your function to test the null hypothesis that the proportion of deaths among individuals who used seatbelt is equal to the proportion of deaths among the individuals who did not use seatbelt.
4. Use a barplot to visualize the distribution of the seatbelt usage across the factor levels of the accident's outcome.

Solution for question 1.1

```
sum(table(nassCDS$seatbelt[nassCDS$seatbelt=='belted']))
```

```
## [1] 18573
```

Solution for question 1.2

```
table(nassCDS$seatbelt,nassCDS$dead)
```

```
##
##          alive  dead
##   none    6964   680
##   belted 18073   500
```

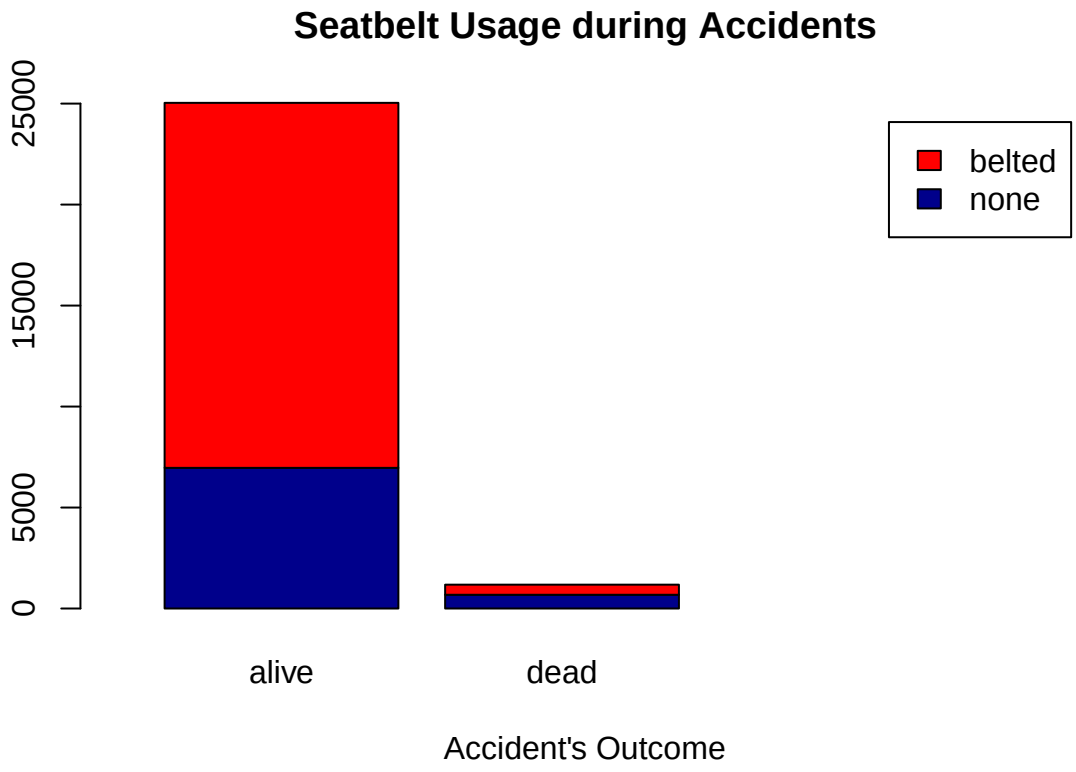
Solution for question 1.3

```
p.test<-function(z)
  {
  none<- z[1,1]+z[1,2]
  belted<-z[2,1]+z[2,2]
  g1.p<-z[1,2]/none
  g2.p<-z[2,2]/belted
  succcess<-c((none*g1.p),(belted*g2.p))
  sum<-c(none,belted)
  result<-prop.test(succcess,sum)
  return(result$p.value)
  }
z<-table(nassCDS$seatbelt,nassCDS$dead)
p.test(z)
```

```
## [1] 3.815086e-107
```

Solution for question 1.4

```
z<-table(nassCDS$seatbelt,nassCDS$dead)
barplot(z, main="Seatbelt Usage during Accidents", xlab="Accident's Outcome", col=c("darkblue","red"), l
```



Question 2

In this question we focus on the outcome of the accident (dead/alive, the variable dead) and the age of the occupant (the variable ageOFocc) in the nassCDS dataset.

1. What is the mean and standard deviation of the age of occupant by accident outcome?
2. Use a boxplot to visualize the distribution of the occupants' age by accident outcome and add the data points on the boxplot.
3. Calculate a 95% confidence interval for the mean difference of the age of occupant using t distribution.

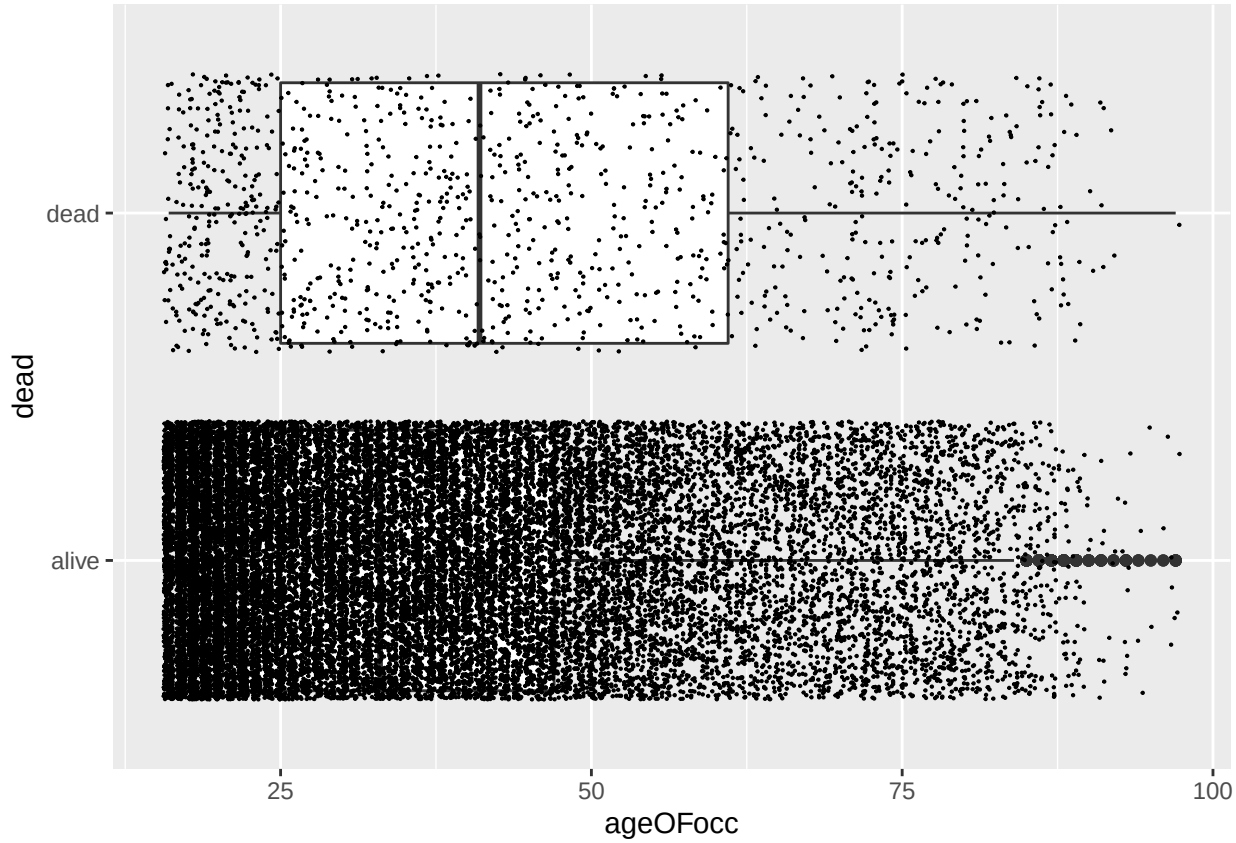
Solution for question 2.1

```
desc<-function(x,y)
  {
  a<-c(mean(x), mean(y))
  b<-c((var(x))*1/2, (var(y))*1/2)
  c<-rbind(a,b)
  colnames(c)<-c("dead", "alive")
  row.names(c)<-c("mean", "sd")
  return(c)
}
desc(nassCDS$ageOFocc[nassCDS$dead == 'dead'], nassCDS$ageOFocc[nassCDS$dead == 'alive'])
```

```
##           dead     alive
## mean  44.61525  36.85701
## sd   227.32202 155.87079
```

Solution for question 2.2

```
library(ggplot2)
library(tidyverse)
nassCDS%>%
  ggplot(aes(ageOFocc,dead))+
  geom_boxplot()+
  geom_jitter(size = 0.1)
```



Solution for question 2.3

```
ttest<-t.test(nassCDS$ageOFocc)
ttest$conf.int
```

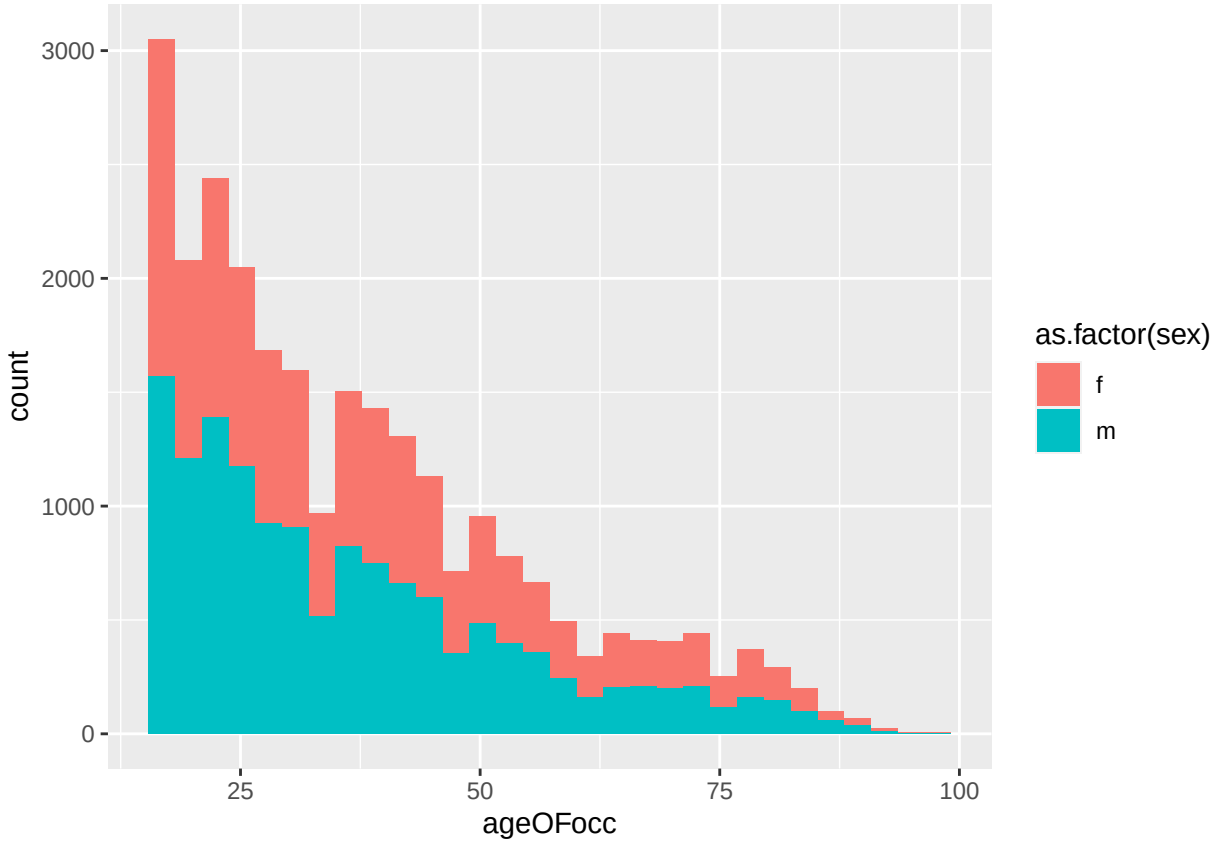
```
## [1] 36.9894 37.4230
## attr(,"conf.level")
## [1] 0.95
```

Question 3

1. Visualize the distribution of the occupant age by sex in the nassCDS dataset.
2. How many occupants over the age of 50 years old survived the accident?
3. Add a binary variable AgeOFocc_class that takes the value of 1 when the occupant age is over 50 years and 0 for when the occupant age is 50 years or less.
4. Create a data frame, nassCDS_o50, containing occupants older than 50 years old. This data frame should contain the variables dead, airbag, weight, and injSeverity. Remove the observations with missing values.
5. What is the dimension of the new data ?
6. Among the occupants who are older than 50 years old, use a barplot to visualize the distribution of airbag across the levels of the accident outcome (dead/alive). The variable dead should be on the x-axis.
7. Among the occupants who are older than 50 years old, visualize the distribution of airbag across the level of injury severity (the variable injSeverity).

Solution for question 3.1

```
nassCDS%>%
  ggplot(aes(x = ageOFocc, fill = as.factor(sex)))+
  geom_histogram(bins = 30)
```



Solution for question 3.2

```
sum(table(nassCDS$ageOFocc[nassCDS$ageOFocc > 50 & nassCDS$dead == 'alive']))
```

```
## [1] 5174
```

Solution for question 3.3

```
AgeOFocc_class<-ifelse(nassCDS$ageOFocc > 50, 1, 0)
```

Solution for question 3.4

```
install.packages("tidyverse")
library(tidyverse)
nassCDS_o50 <- na.omit(nassCDS)%>%
```

```
  filter(ageOFocc > 50)%>%
  select(dead, airbag, weight, injSeverity)
```

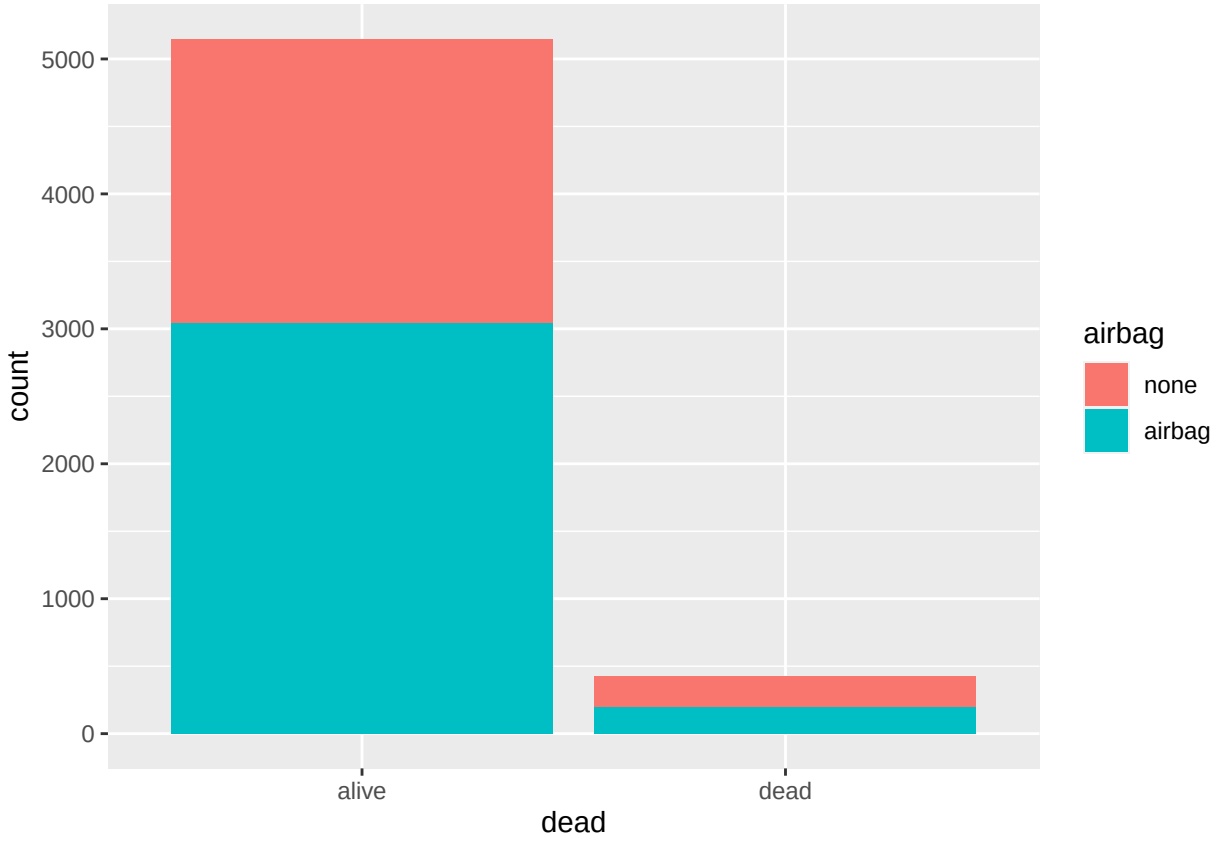
Solution for question 3.5

```
dim(nassCDS_o50)
```

```
## [1] 5573    4
```

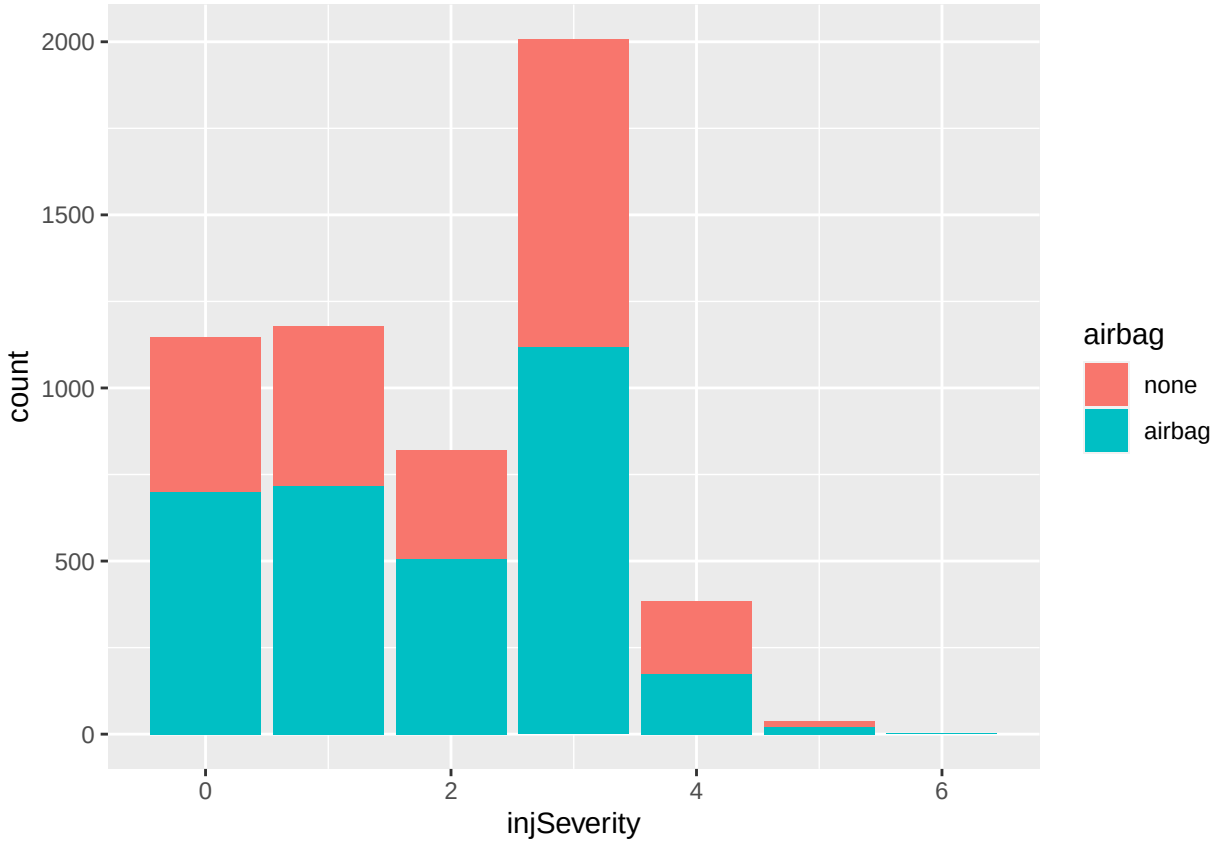
Solution for question 3.6

```
nassCDS_o50%>%
  ggplot(aes(x = dead, fill = airbag))+
  geom_bar()
```



Solution for question 3.7

```
library(ggplot2)
ggplot(nassCDS_o50, aes(x=injSeverity, fill=airbag))+ geom_bar(position="stack")
```



Question 4

Write a R function that receives as an input the nassCDS dataset. The function should conduct the following analysis:

1. Select only the observations for which the accident outcome is "dead".
2. Calculate percentage of deaths out of the overall number of observations.
3. Calculate the percentages of females and males among the occupants who died in the accident.
4. Show the most frequent severity of their injuries.
5. Calculate the minimum and maximum age of the occupant (the variable ageOFocc).
6. Produce a histogram with the severity of injuries on the x axis and the frequency of occupant's age on the y axis
7. This **SINGLE** Function should return **two** outputs:
    - Numerical output: 4.2,4.3,4.4 and 4.5 as a table.
    - Graphical output: 4.6 as a plot.

Solution for question 4.1

```
library(dplyr)
counts_dead<-function(x)
  {
  dead_obs<-x%>%filter(x[,3]=="dead")
```

```
  return(dead_obs)
}
new_nassCDS<-counts_dead(nassCDS)
```

Solution for question 4.2

```
percent_dead<-function(x)
  {
  p_dead<-(sum(table(x[,3][x[,3]=="dead"]))/sum(table(x[,3])))
  return(p_dead*100)
}
percent_dead(nassCDS)
```

```
## [1] 4.500896
```

Solution for question 4.3

```
counts_gender<-function(x)
{
total <- sum(table(x[7]))
male <- sum(table(x[7][x[7]=="m" & x[3]=="dead"]))
male_p <- ((male/total)*100)
female <- sum(table(x[7][x[7]=="f" & x[3]=="dead"]))
female_p <- ((female/total)*100)
y <- c(Male.Casualty.Percent=male_p,Female.Casualty.Percent=female_p)
return(y)
}
counts_gender(nassCDS)
```

```
##   Male.Casualty.Percent Female.Casualty.Percent
##                2.731052                1.769844
```

Solution for question 4.4

```
freq.inj <- function(x)
  {
  tally <- table(x[14])
  max.tally <- tally[which.max(tally)]
  return(max.tally)
}
freq.inj(nassCDS)
```

```
##    3
## 8495
```

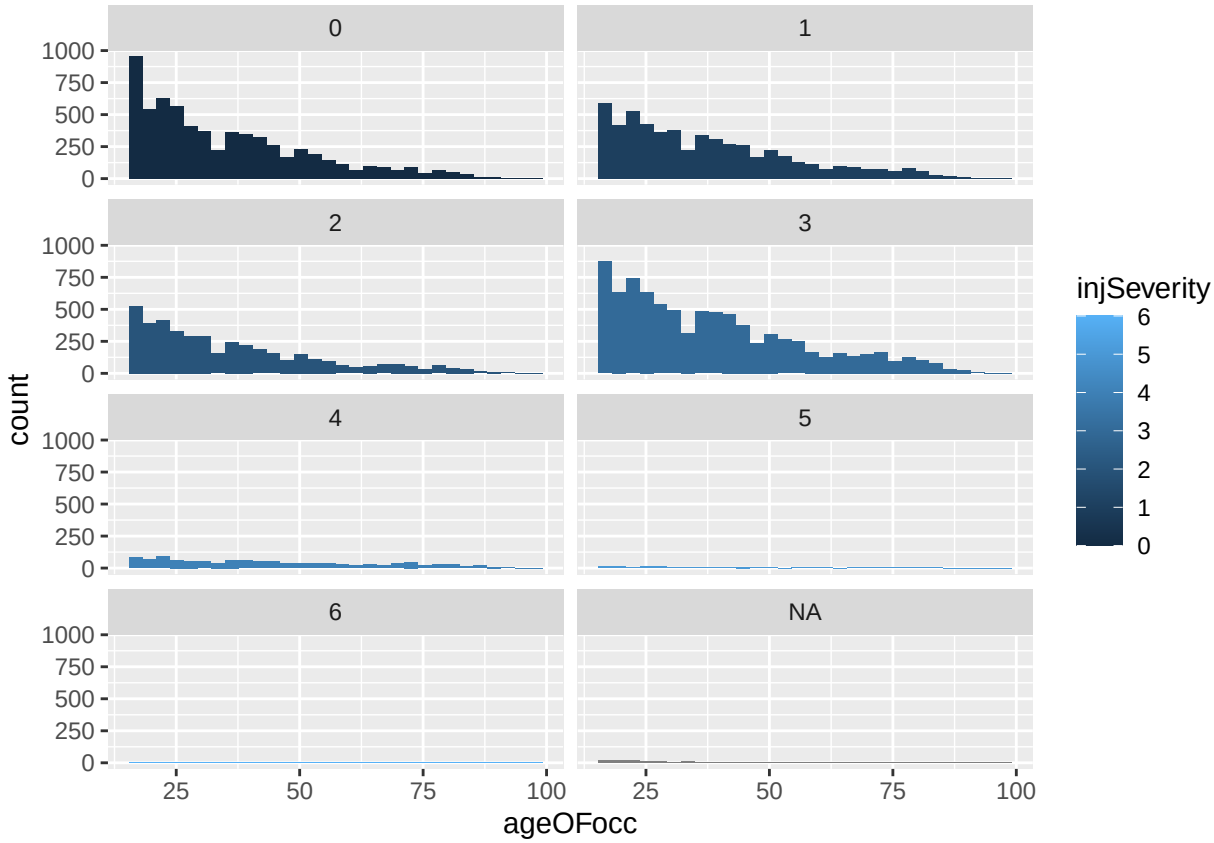
Solution for question 4.5

```
age.extent<-function(x)
  {
  age.data <- x[8]
  min.age <- min(age.data)
  max.age <- max(age.data)
  age.final <- c(MinimumAge = min.age, MaximumAge = max.age)
  return(age.final)
}
age.extent(nassCDS)
```

```
## MinimumAge MaximumAge
##         16         97
```

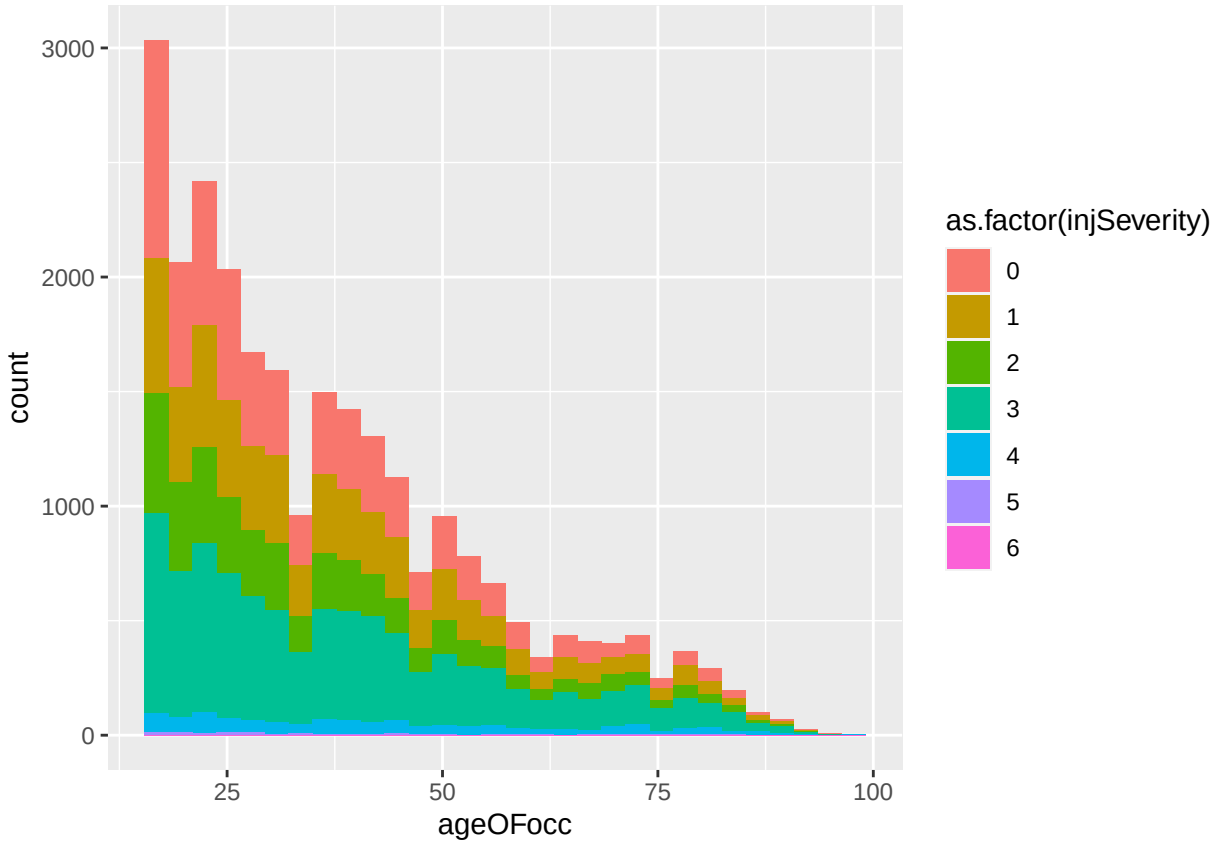
Solution for question 4.6

```
hist.inj<-function(x)
{
plot<-ggplot(x, aes(ageOFocc,fill = injSeverity)) + geom_histogram(bins = 30) + facet_wrap(~x$injSeveri
return(plot)
}
hist.inj(nassCDS)
```



Solution for question 4.7

```
hist.inj<-function(x)
{
plot<- na.omit(x)%>%
  ggplot(aes(ageOFocc,fill = as.factor(injSeverity)))+
  geom_histogram(bins = 30)
print(plot)
return(table)
}
hist.inj(nassCDS)
```




```
## function (..., exclude = if (useNA == "no") c(NA, NaN), useNA = c("no",
##     "ifany", "always"), dnn = list.names(...), deparse.level = 1)
## {
##     list.names <- function(...) {
##         l <- as.list(substitute(list(...)))[-1L]
##         nm <- names(l)
##         fixup <- if (is.null(nm))
##             seq_along(l)
##         else nm == ""
##         dep <- vapply(l[fixup], function(x) switch(deparse.level +
##             1, "", if (is.symbol(x)) as.character(x) else "",
##             deparse(x, nlines = 1)[1L]), "")
```

```
##         if (is.null(nm))
##             dep
##         else {
##             nm[fixup] <- dep
##             nm
##         }
##     }
##     miss.use <- missing(useNA)
##     miss.exc <- missing(exclude)
##     useNA <- if (miss.use && !miss.exc && !match(NA, exclude,
##         nomatch = 0L))
##         "ifany"
##     else match.arg(useNA)
##     doNA <- useNA != "no"
##     if (!miss.use && !miss.exc && doNA && match(NA, exclude,
##         nomatch = 0L))
##         warning("'exclude' containing NA and 'useNA' != \"no\"' are a bit contradicting")
##     args <- list(...)
##     if (length(args) == 1L && is.list(args[[1L]])) {
##         args <- args[[1L]]
##         if (length(dnn) != length(args))
##             dnn <- if (!is.null(argn <- names(args)))
##                 argn
##             else paste(dnn[1L], seq_along(args), sep = ".")
##     }
##     if (!length(args))
##         stop("nothing to tabulate")
##     bin <- 0L
##     lens <- NULL
##     dims <- integer()
##     pd <- 1L
##     dn <- NULL
##     for (a in args) {
##         if (is.null(lens))
##             lens <- length(a)
##         else if (length(a) != lens)
##             stop("all arguments must have the same length")
##         fact.a <- is.factor(a)
##         if (doNA)
##             aNA <- anyNA(a)
##         if (!fact.a) {
##             a0 <- a
##             a <- factor(a, exclude = exclude)
##         }
##         add.na <- doNA
##         if (add.na) {
##             ifany <- (useNA == "ifany")
##             anNAc <- anyNA(a)
##             add.na <- if (!ifany || anNAc) {
##                 ll <- levels(a)
##                 if (add.ll <- !anyNA(ll)) {
##                   ll <- c(ll, NA)
##                   TRUE
##                 }
```

```
##                 else if (!ifany && !anNAc)
##                   FALSE
##                 else TRUE
##             }
##             else FALSE
##         }
##         if (add.na)
##             a <- factor(a, levels = ll, exclude = NULL)
##         else ll <- levels(a)
##         a <- as.integer(a)
##         if (fact.a && !miss.exc) {
##             ll <- ll[keep <- which(match(ll, exclude, nomatch = 0L) ==
##                 0L)]
##             a <- match(a, keep)
##         }
##         else if (!fact.a && add.na) {
##             if (ifany && !aNA && add.ll) {
##                 ll <- ll[!is.na(ll)]
##                 is.na(a) <- match(a0, c(exclude, NA), nomatch = 0L) >
##                   0L
##             }
##             else {
##                 is.na(a) <- match(a0, exclude, nomatch = 0L) >
##                   0L
##             }
##         }
##         nl <- length(ll)
##         dims <- c(dims, nl)
##         if (prod(dims) > .Machine$integer.max)
##             stop("attempt to make a table with >= 2^31 elements")
##         dn <- c(dn, list(ll))
##         bin <- bin + pd * (a - 1L)
##         pd <- pd * nl
##     }
##     names(dn) <- dnn
##     bin <- bin[!is.na(bin)]
##     if (length(bin))
##         bin <- bin + 1L
##     y <- array(tabulate(bin, pd), dims, dimnames = dn)
##     class(y) <- "table"
##     y
## }
## <bytecode: 0x0000000015338330>
## <environment: namespace:base>
```

Question 5

1. Use the nassCDS dataset to create a new data frame which contains only occupants who used seatbelt.
2. How many occupants used seatbelt ?
3. Among the individuals who used seatbelt, how many died and how many survived the accident ?
4. Among the individuals who used seatbelt, how many were drivers among the individuals who died and how many were passengers among the individuals who survived the accident (use the variable occRole to identify drivers/passengers) ?

5. Sort the data frame according to the injury's severity and the occupant age.
6. Print the 25 occupants with the highest weight.

Solution for question 5.1

```
belted <- nassCDS %>% filter(seatbelt == "belted")
```

Solution for question 5.2

```
nrow(belted)
```

```
## [1] 18573
```

Solution for question 5.3

```
table(belted$dead)
```

```
##
## alive  dead
## 18073   500
```

Solution for question 5.4

```
belted%>%
  group_by(occRole, dead)%>%
  summarize(count=table(occRole))
```

```
## # A tibble: 4 x 3
## # Groups:   occRole [2]
##   occRole dead  count
##   <chr>   <fct> <table>
## 1 driver  alive 14550
## 2 driver  dead    363
## 3 pass    alive  3523
## 4 pass    dead    137
```

Solution for question 5.5

```
head(belted%>%arrange(injSeverity,ageOFocc), 30)
```

```
##        dvcat   weight  dead airbag seatbelt frontal sex ageOFocc yearacc
## 32   1-9km/h  745.351 alive   none   belted       1   m       16    1997
## 477    10-24   18.361 alive airbag   belted       0   m       16    1997
```

```
## 682    10-24   99.996 alive   none   belted       1   f       16    1997
## 1028   10-24   72.388 alive airbag   belted       1   m       16    1997
## 1029   10-24   72.388 alive airbag   belted       1   m       16    1997
## 1033   10-24 1054.928 alive   none   belted       1   m       16    1997
## 1046   10-24 1648.787 alive   none   belted       0   f       16    1997
## 1068   25-39   49.775 alive airbag   belted       0   f       16    1997
## 1077   10-24  241.148 alive   none   belted       1   m       16    1997
## 1112   10-24  378.095 alive airbag   belted       1   m       16    1997
## 1160   25-39  417.842 alive airbag   belted       1   m       16    1997
## 1243   10-24   55.613 alive   none   belted       1   m       16    1997
## 1370   10-24 1168.518 alive airbag   belted       1   f       16    1997
## 1371   10-24 1168.518 alive   none   belted       1   f       16    1997
## 1533 1-9km/h  881.053 alive airbag   belted       1   m       16    1997
## 1613 1-9km/h  366.762 alive   none   belted       0   m       16    1997
## 1760   10-24  489.014 alive airbag   belted       1   m       16    1997
## 1761 1-9km/h 1033.131 alive   none   belted       0   m       16    1997
## 1762 1-9km/h 1033.131 alive   none   belted       0   f       16    1997
## 2007   10-24   51.137 alive airbag   belted       1   m       16    1997
## 2074 1-9km/h 2287.349 alive   none   belted       1   f       16    1997
## 2117   25-39   38.595 alive   none   belted       1   m       16    1997
## 2611   40-54  294.141 alive   none   belted       1   m       16    1997
## 2715   10-24  964.986 alive airbag   belted       1   f       16    1997
## 3192   25-39   15.897 alive airbag   belted       1   f       16    1997
## 3254   10-24   14.037 alive   none   belted       1   m       16    1997
## 3255   25-39   17.315 alive   none   belted       0   f       16    1997
## 3285   25-39 3639.196 alive   none   belted       1   m       16    1997
## 3386   10-24   87.230 alive   none   belted       0   f       16    1997
## 3416   10-24  161.130 alive   none   belted       1   f       16    1997
##      yearVeh    abcat occRole deploy injSeverity   caseid
## 32      1988  unavail  driver      0           0   2:35:1
## 477     1992 nodeploy  driver      0           0   6:19:2
## 682     1994  unavail  driver      0           0  8:107:1
## 1028    1996 nodeploy  driver      0           0  11:33:1
## 1029    1996 nodeploy    pass      0           0  11:33:1
## 1033    1989  unavail    pass      0           0  11:39:1
## 1046    1989  unavail  driver      0           0  11:51:1
## 1068    1996 nodeploy  driver      0           0  11:64:1
## 1077    1986  unavail  driver      0           0  11:76:1
## 1112    1993   deploy  driver      1           0 11:106:1
## 1160    1992   deploy  driver      1           0 11:144:1
## 1243    1990  unavail  driver      0           0  12:21:1
## 1370    1991   deploy  driver      1           0 12:110:1
## 1371    1991  unavail    pass      0           0 12:110:1
## 1533    1997   deploy  driver      1           0 12:224:2
## 1613    1986  unavail  driver      0           0  13:54:1
## 1760    1997 nodeploy    pass      0           0 13:181:1
## 1761    1984  unavail  driver      0           0 13:183:1
## 1762    1984  unavail    pass      0           0 13:183:1
## 2007    1989 nodeploy  driver      0           0  43:11:1
## 2074    1989  unavail  driver      0           0  43:61:2
## 2117    1992  unavail  driver      0           0 43:100:2
## 2611    1986  unavail  driver      0           0  48:59:2
## 2715    1996   deploy  driver      1           0  49:47:1
## 3192    1995   deploy    pass      1           0  74:58:2
```

```
## 3254    1978  unavail  driver      0           0 74:144:3
## 3255    1992  unavail  driver      0           0 74:146:1
## 3285    1989  unavail    pass      0           0  75:12:1
## 3386    1986  unavail    pass      0           0  75:86:2
## 3416    1992  unavail    pass      0           0 75:125:1
```

Solution for question 5.6

```
print(belted%>%arrange(desc(weight))%>%top_n(25,weight))
```

```
##         dvcat   weight  dead airbag seatbelt frontal sex ageOFocc yearacc
## 6844    10-24 57871.59 alive airbag   belted       1   f       59    1998
## 6888    10-24 57871.59 alive airbag   belted       1   f       16    1998
## 6889    10-24 57871.59 alive airbag   belted       1   m       16    1998
## 2630    10-24 47463.09 alive airbag   belted       0   f       24    1997
## 25422   10-24 31694.04 alive airbag   belted       1   m       33    2002
## 25423   10-24 31694.04 alive airbag   belted       1   f       47    2002
## 10596   10-24 29533.13 alive airbag   belted       1   m       19    1999
## 24332   10-24 29301.14 alive   none   belted       1   m       23    2002
## 10572   10-24 28281.11 alive   none   belted       0   m       43    1999
## 20993   25-39 28215.77 alive   none   belted       1   m       35    2001
## 20994   25-39 28215.77 alive   none   belted       1   f       37    2001
## 11092   10-24 26789.34 alive airbag   belted       0   m       30    1999
## 10538   10-24 25988.01 alive   none   belted       0   f       34    1999
## 2507  1-9km/h 25688.37 alive airbag   belted       0   m       36    1997
## 19732   10-24 25029.20 alive airbag   belted       1   f       31    2001
## 19733   10-24 25029.20 alive   none   belted       1   m       37    2001
## 19704 1-9km/h 23428.61 alive   none   belted       0   f       31    2001
## 15682   10-24 23131.14 alive airbag   belted       1   m       18    2000
## 15683 1-9km/h 23131.14 alive airbag   belted       0   m       42    2000
## 15684 1-9km/h 23131.14 alive airbag   belted       0   f       43    2000
## 19612   10-24 22232.64 alive   none   belted       1   m       36    2001
## 10521 1-9km/h 20296.81 alive   none   belted       0   m       16    1999
## 10522 1-9km/h 20296.81 alive   none   belted       0   m       17    1999
## 24242   10-24 20081.63 alive airbag   belted       0   m       26    2002
## 24243   10-24 20081.63 alive airbag   belted       0   f       21    2002
##       yearVeh    abcat occRole deploy injSeverity   caseid
## 6844     1990   deploy  driver      1           2  48:97:1
## 6888     1998 nodeploy  driver      0           0 48:139:1
## 6889     1998 nodeploy    pass      0           0 48:139:1
## 2630     1995 nodeploy  driver      0           0  48:92:2
## 25422    1994 nodeploy  driver      0           0  75:24:2
## 25423    1994 nodeploy    pass      0           0  75:24:2
## 10596    1995   deploy  driver      1           0 43:191:1
## 24332    1984  unavail  driver      0           1  48:96:1
## 10572    1992  unavail  driver      0           0 43:174:1
## 20993    1987  unavail  driver      0           0 76:113:2
## 20994    1987  unavail    pass      0           1 76:113:2
## 11092    1996   deploy  driver      1           3 48:131:1
## 10538    1993  unavail  driver      0           0 43:151:1
## 2507     1993 nodeploy  driver      0           0 45:169:1
## 19732    1994   deploy  driver      1           0 48:117:1
```

```
## 19733    1994  unavail    pass      0           0 48:117:1
## 19704    1993  unavail  driver      0           3  48:94:2
## 15682    2000 nodeploy  driver      0           2 48:164:1
## 15683    1998 nodeploy  driver      0           0 48:164:2
## 15684    1998 nodeploy    pass      0           0 48:164:2
## 19612    1991  unavail  driver      0           0   48:8:2
## 10521    1990  unavail  driver      0           0 43:135:2
## 10522    1990  unavail    pass      0           0 43:135:2
## 24242    1996 nodeploy  driver      0           0  48:34:2
## 24243    1996 nodeploy    pass      0           0  48:34:2
```

Question 6

Prepare a presentation of 5-10 slides using R markdown about the connection between the usage of seatbelt, the outcome of the accident and the severity of the injury. Make sure that your presentation includes:

- A Title slide.
- At least one slide with text.
- At least one slide with a figure
- At least one slide with text and a figure.

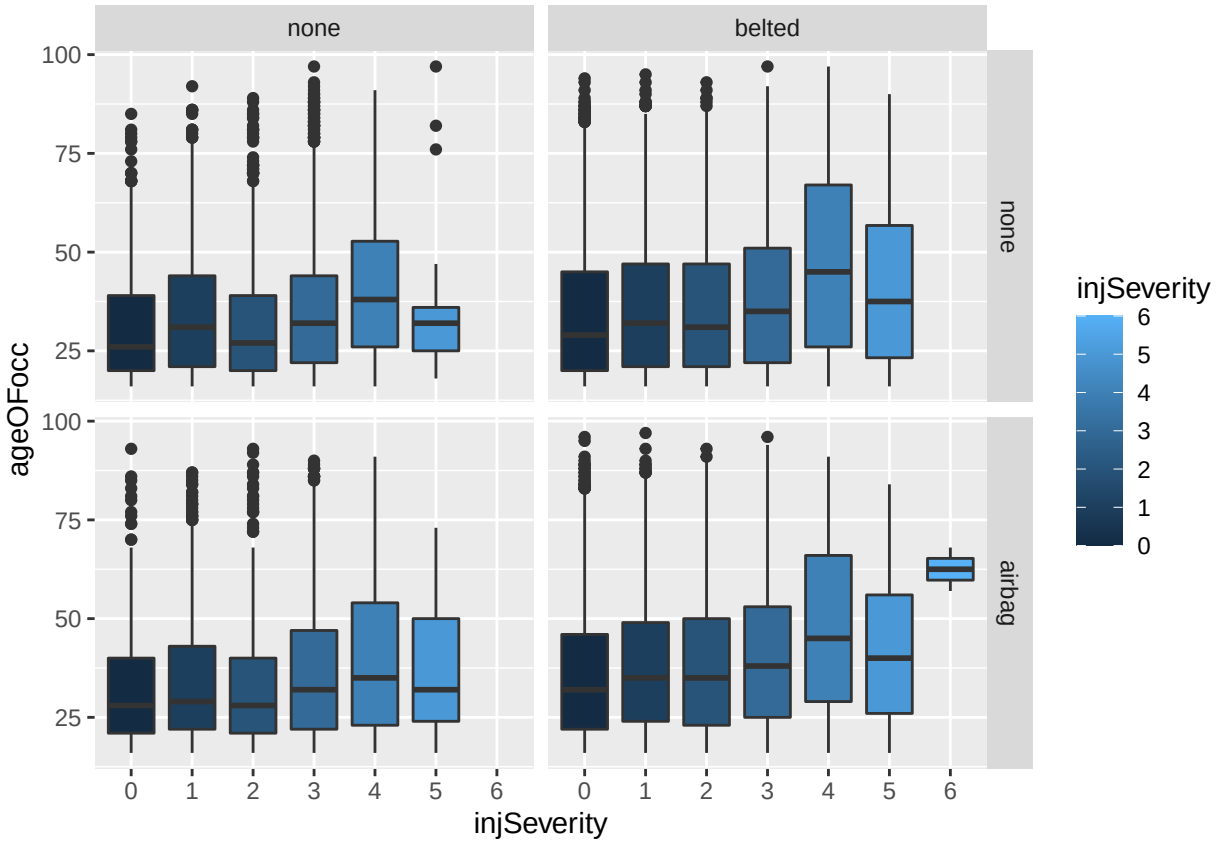
Please note that you **WILL NOT** be asked to give the presentation and you **WILL NOT** be asked questions about the presentation. Your aim in this question is to demonstrate that you know how to use R markdown to make a presentation about your analysis. More details how to make a presentation using R markdown: https://rmarkdown.rstudio.com/lesson-11.html.

Question 7

1. Use the nassCDS dataset to produce Figure 7.1 presented below.

Solution for question 7.1

```
nassCDS %>%
  drop_na() %>%
  ggplot(aes(x = as.factor(injSeverity), y = ageOFocc , fill = injSeverity))+
  geom_boxplot() +
  facet_grid(airbag~seatbelt) +
  xlab("injSeverity")
```



Part 2: the atmos dataset

In this part, the questions are focused on the atmos dataset which is a part of the nasaweather R package. To access the data you need to install the package. More information can be found in https://github.com/hadley/nasaweather. You can use the code below to access the data.

```
library(nasaweather)
data(atmos)
names(atmos)
```

```
##  [1] "lat"       "long"      "year"      "month"     "surftemp"  "temp"
##  [7] "pressure"  "ozone"     "cloudlow"  "cloudmid"  "cloudhigh"
```

```
dim(atmos)
```

```
## [1] 41472    11
```

Question 8

1. Use the atmos dataset to calculate the mean temperature (the variable temp) by year.
2. Create a new data frame in which only data from 1995 are included.
3. For 1995, calculate the mean, trimmed mean (10%) and median temperature by month.

4. Produce a dotplot for the temperature by month and include in the plot the mean temperature by month.
5. Use a for loop to calculate a 95% confidence interval for the mean temperature by month.
6. Write a function that calculate the Pearson correlation between two vectors and applied this function, within a for loop, for the temperature and ozone (the variable ozone) level by year.
7. Plot the yearly correlation between temperature and ozone and add a line for the overall correlation.

Solution for question 8.1

```
atmos %>% group_by(year) %>% summarize(temp = mean(temp))
```

```
## # A tibble: 6 x 2
##    year  temp
##   <int> <dbl>
## 1  1995  297.
## 2  1996  297.
## 3  1997  298.
## 4  1998  299.
## 5  1999  298.
## 6  2000  298.
```

Solution for question 8.2

```
atmos95 <- atmos %>% filter(year == '1995')
```

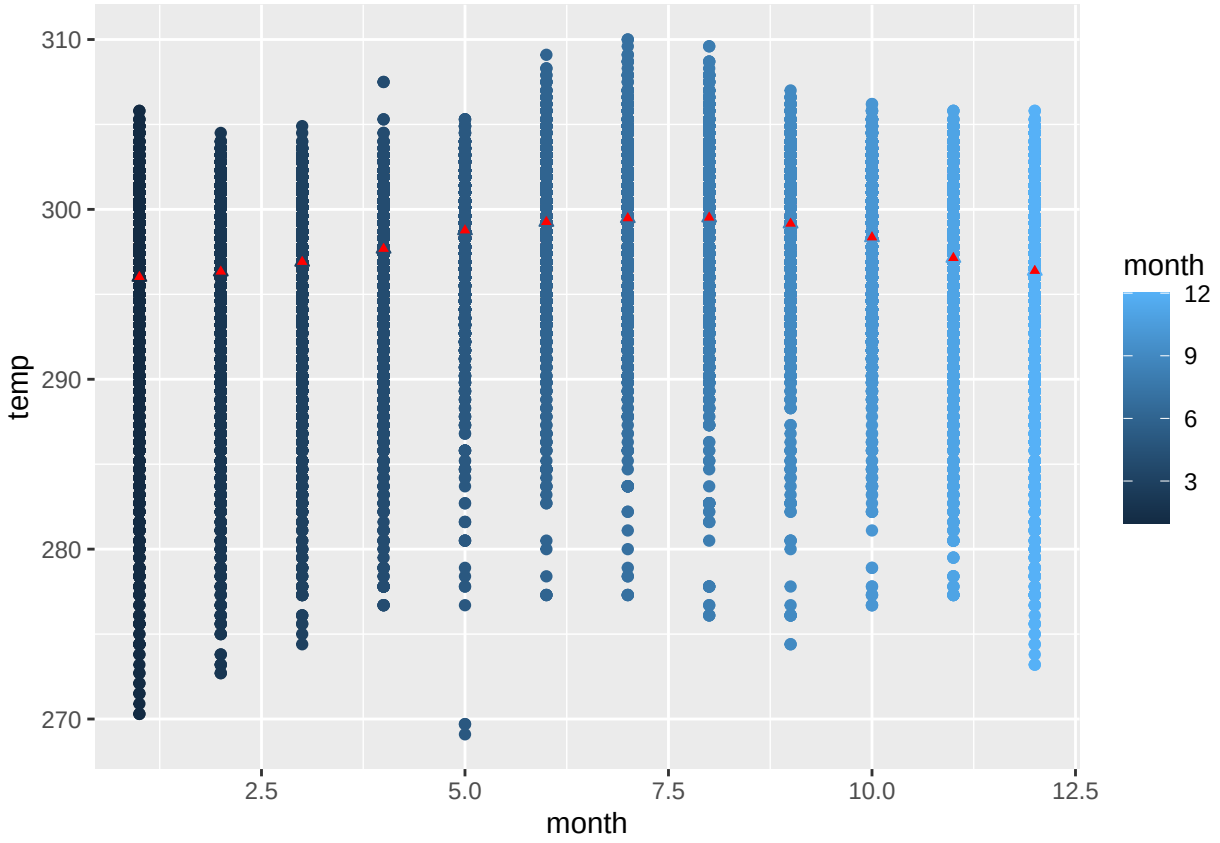
Solution for question 8.3

```
atmos95 %>%
  group_by(month) %>%
  summarize(mean = mean(temp), trimmed = mean(temp, trim=0.10), median = median(temp))
```

```
## # A tibble: 12 x 4
##    month  mean trimmed median
##    <int> <dbl>   <dbl>  <dbl>
##  1     1  295.    296.   297.
##  2     2  295.    296.   297.
##  3     3  296.    297.   298.
##  4     4  297.    298.   298.
##  5     5  298.    298.   298.
##  6     6  299.    299.   299.
##  7     7  299.    300.   300.
##  8     8  299.    300.   300.
##  9     9  298.    299.   300.
## 10    10  298.    298.   299.
## 11    11  297.    297.   297.
## 12    12  296.    297.   297.
```

Solution for question 8.4

```
atmos %>%
  ggplot(aes(x=month, y=temp, col=month))+
  geom_point()+
  stat_summary(geom = "point",fun = "mean", fill = "red", shape = 24)
```



Solution for question 8.5

```
for(i in 1:12)
  {
  monthly<-t.test(atmos$temp[atmos$month == i])
  conf.int<-monthly$conf.int
  print(conf.int)
  }
```

```
## [1] 295.8431 296.2122
## attr(,"conf.level")
## [1] 0.95
## [1] 296.168 296.519
## attr(,"conf.level")
## [1] 0.95
## [1] 296.7466 297.0819
## attr(,"conf.level")
## [1] 0.95
```

```
## [1] 297.5366 297.8318
## attr(,"conf.level")
## [1] 0.95
## [1] 298.6516 298.8850
## attr(,"conf.level")
## [1] 0.95
## [1] 299.1431 299.3836
## attr(,"conf.level")
## [1] 0.95
## [1] 299.3475 299.6158
## attr(,"conf.level")
## [1] 0.95
## [1] 299.3735 299.6667
## attr(,"conf.level")
## [1] 0.95
## [1] 299.0190 299.3085
## attr(,"conf.level")
## [1] 0.95
## [1] 298.2232 298.4915
## attr(,"conf.level")
## [1] 0.95
## [1] 296.9882 297.3000
## attr(,"conf.level")
## [1] 0.95
## [1] 296.2084 296.5622
## attr(,"conf.level")
## [1] 0.95
```

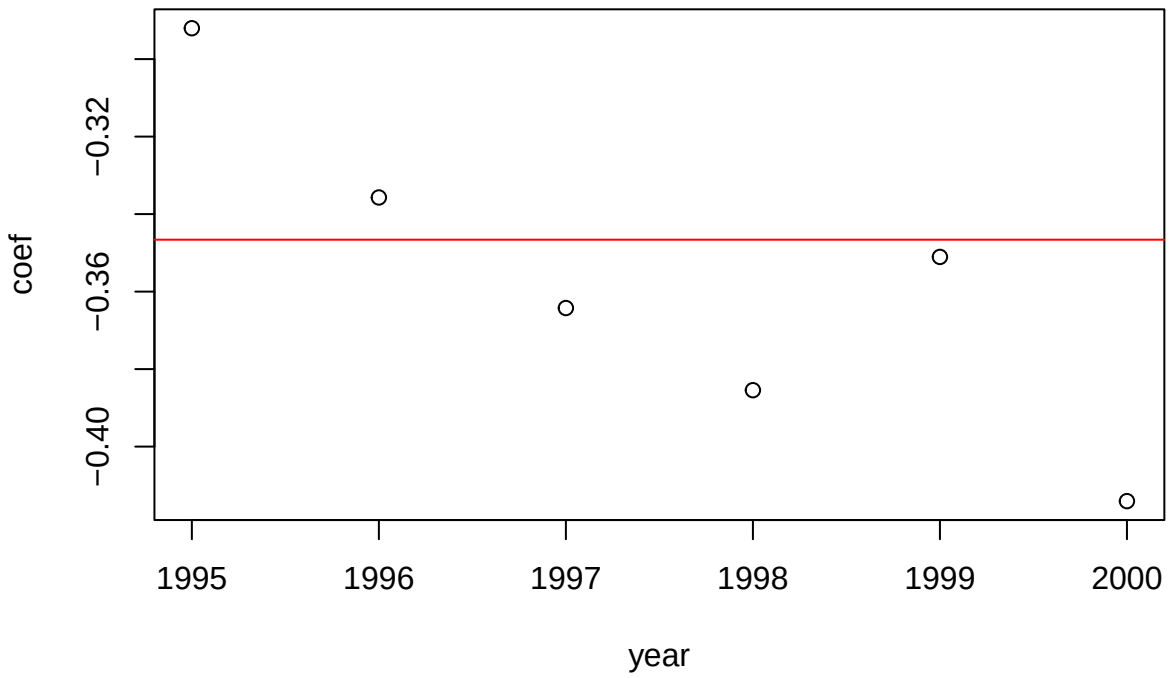
Solution for question 8.6

```
year <- c(1:6)
coef<-c(1:6)

for(i in 1:6)
{
by <-c(1994+i)
yearly<- cor(atmos$temp[atmos$year == by],atmos$ozone[atmos$year == by])
year[i] <- c(by)
coef[i]<-c(yearly)
cor.coef <-cbind(year, coef)
}

cor.coef
```

```
##      year       coef
## [1,] 1995 -0.2920363
## [2,] 1996 -0.3357104
## [3,] 1997 -0.3642401
## [4,] 1998 -0.3854426
## [5,] 1999 -0.3510499
## [6,] 2000 -0.4140677
```

Solution for question 8.7

```
plot(cor.coef)
abline(h=cor(atmos$temp, atmos$ozone), col = "red")
```



Question 9

In the atmos dataset, the variable cloudlow is the Mean Low Cloud Coverage, that is the monthly mean percent of the sky covered by clouds with cloud top pressure greater than 680 mb. Let q75 be the 75% quantile of the cloudlow distribution.

1. Define a R object which is equal to the 75% quantile.
2. Produce a histogram for cloudlow and add a vertical line (in red) which represents the 75% quantile of the distribution.
3. Create a new data frame which includes only observations with cloudlow $>$ q75. How many observations were included in the new data frame?
4. Produce a scatterplot of cloudlow (Y) versus ozone (X). Use different colors for data points of each month.
5. Produce a multiway histogram in which you plot the histogram for cloudlow for each year. Produce the histograms in a 3X2 panel (three columns with two figures in each column).

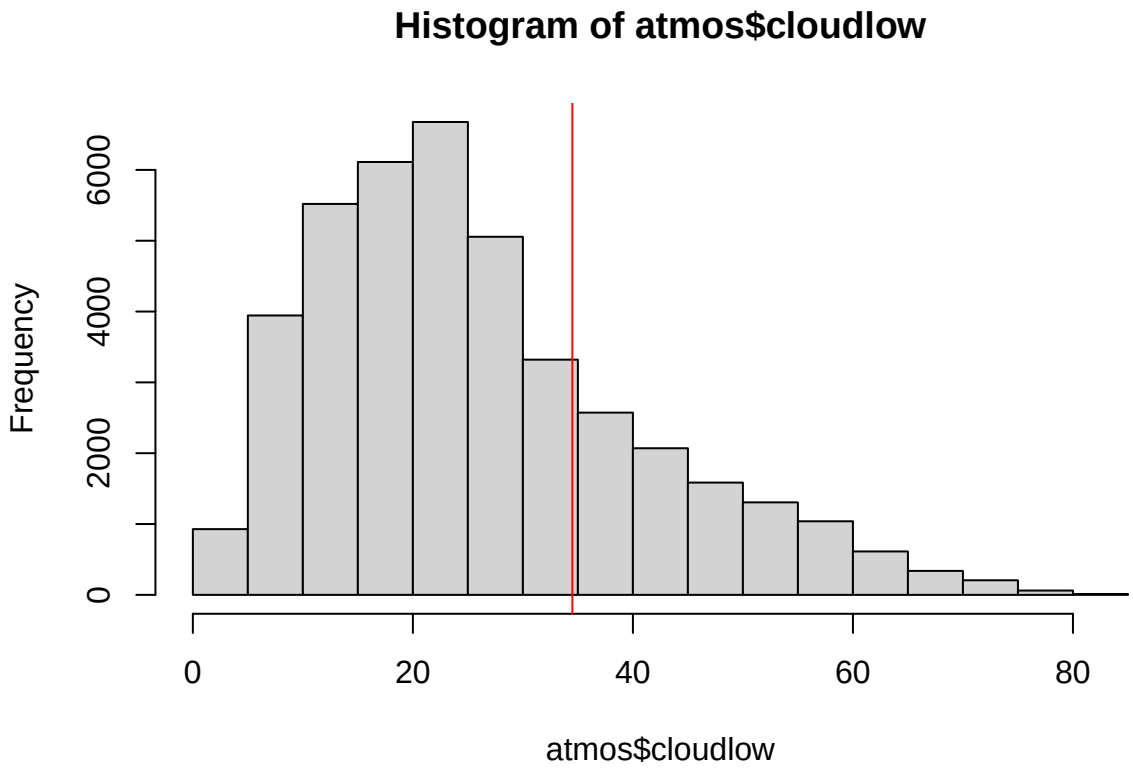
Solution for question 9.1

```
q75<-quantile(atmos$cloudlow, probs = c(0.75), na.rm = TRUE)
q75
```

```
##  75%
## 34.5
```

Solution for question 9.2

```
hist(atmos$cloudlow)
abline(v=q75, col = "red")
```



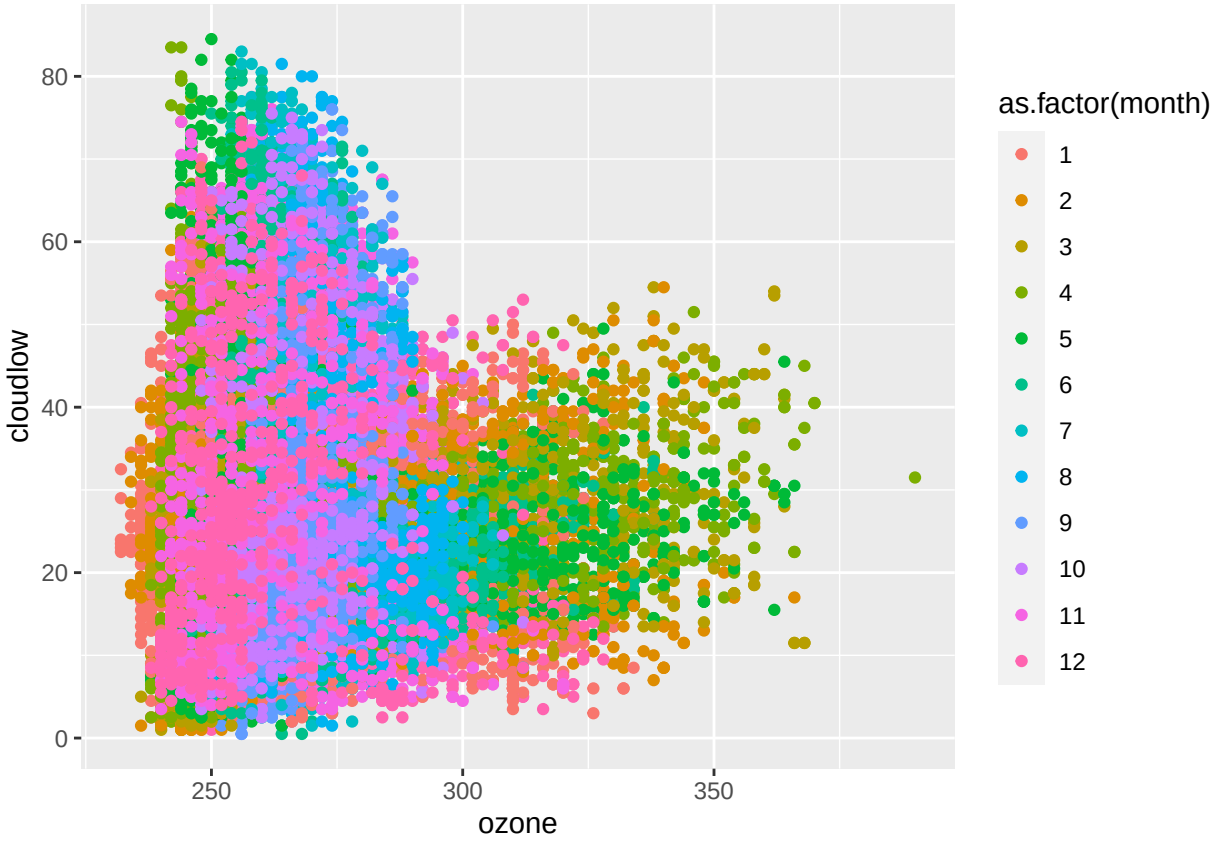
Solution for question 9.3

```
atmos_q75<-atmos%>%filter(cloudlow > q75)
nrow(atmos_q75)
```

```
## [1] 10103
```

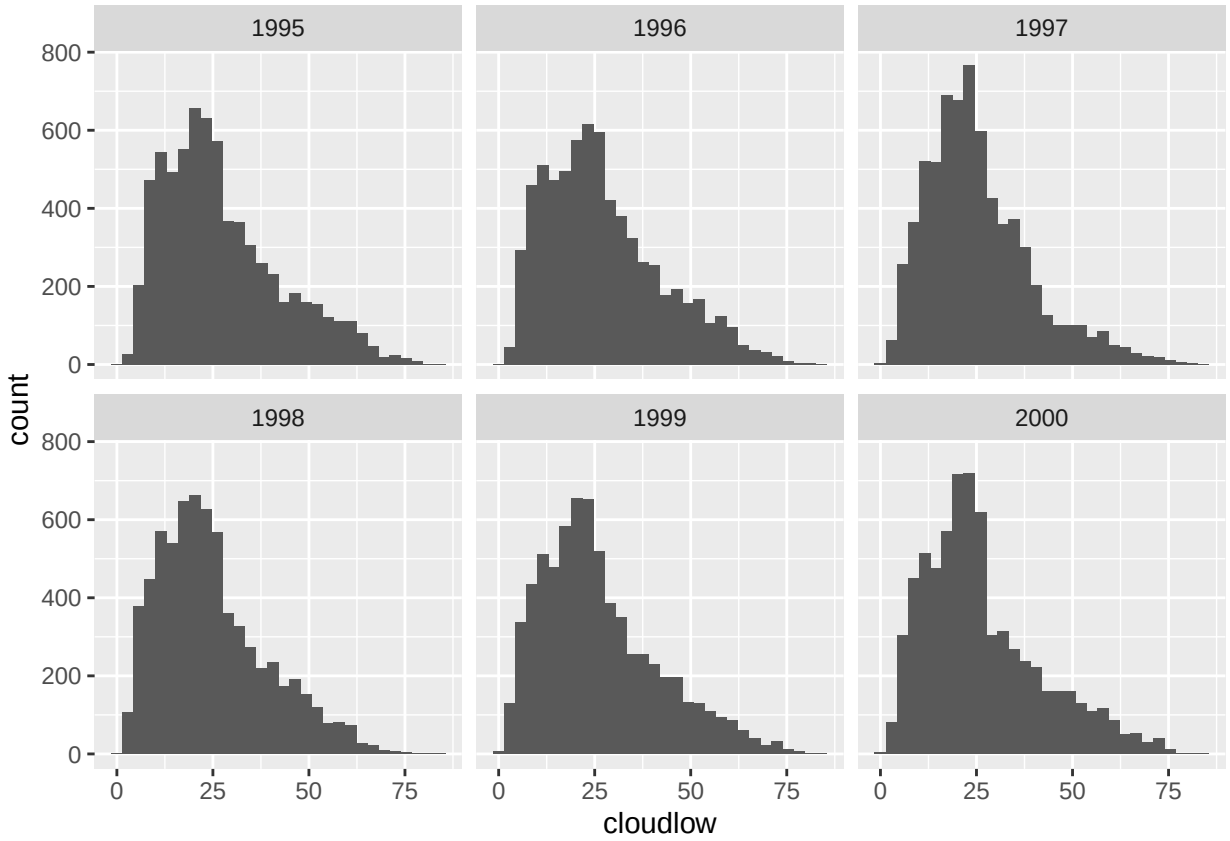
Solution for question 9.4

```
atmos%>%
  ggplot(aes(x = ozone, y=cloudlow)) +
  geom_point(aes(color = as.factor(month)))
```



Solution for question 9.5

```
atmos%>%
  ggplot(aes(x=cloudlow))+
  geom_histogram(bins = 30)+
  facet_wrap(~year,ncol = 3, nrow=2)
```



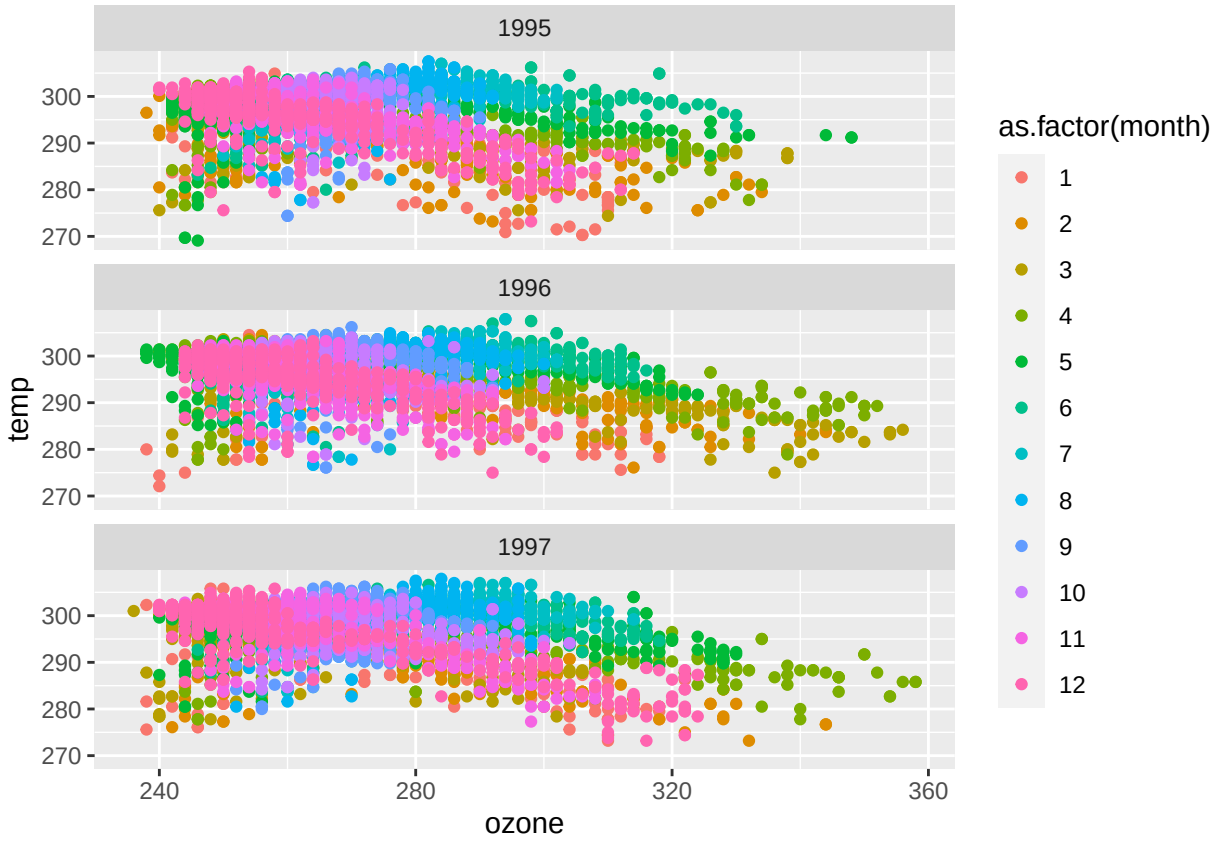
Question 10

1. Use the atmos dataset to produce Figure 10.1 presented below.

- Hint: based on the data and variables presented in the figure you need to create a new data frame and produce a figure using the new data frame.

Solution for question 10.1

```
atmos_567<-atmos%>%filter(year <= 1997)

atmos_567%>%
  ggplot(aes(x = ozone, y=temp, color=as.factor(month))) +
  geom_point() +
  facet_wrap(~year,ncol = 1)
```



Part 3: the Legosets data

In this part we focus on the Legosets dataset which can be downloaded from https://github.com/seankross/lego following the instructions in the github page. The Legosets dataset contains 6172 rows and 14 columns. Make sure that you install the devtools and the lego packages. Use the code below to install and access the data.

```
library(devtools)
install_github("seankross/lego")
library(lego)
data("legosets")
names(legosets)
```

```
##  [1] "Item_Number"  "Name"         "Year"         "Theme"        "Subtheme"
##  [6] "Pieces"       "Minifigures"  "Image_URL"    "GBP_MSRP"     "USD_MSRP"
## [11] "CAD_MSRP"     "EUR_MSRP"     "Packaging"    "Availability"
```

Question 11

1. Are there missing values in the Legosets dataset? If yes, how many?
2. Replace the missing values by the median for the numerical columns and by the most frequent value in the categorical columns. In the case of integer columns, round up the replaced values to a whole number(if they're decimal).

3. Create a function that summarizes the dataset, this function should output a table displaying the number of rows, number of columns, and the averages, max, and min values for the price in British Pounds(GBP) columns and the most frequent SubTheme, Packaging, and Availability for the observations with Star Wars theme.
4. Create a new dataframe data1 that contains Pieces, GBP_MSRP, and Theme sorted in decreasing order of Pieces and increasing order of GBP_MSRP.
5. Create a new dataframe data2 that consists of the variables Pieces, GBP_MSRP, Availability for the theme "City" and merge it with data1 based on the rows in data2.
6. Produce the merged data frame in 11.5 in a different way.
7. Fit a linear regression model on the filtered data frame from question 11.5 with the price (GBP_MSRP) being the response variable and the number of pieces (Pieces) as a predictor.
8. Produce a scatterplot of price (Y) versus pieces (X) and add the regression line to the scatterplot.

Solution for question 11.1

```
sum(is.na(legosets))
```

```
## [1] 13708
```

Solution for question 11.2

```
most.frequent<-function(x)
  {
  y<-sort(table(x), decreasing = TRUE)[1]
  z<-unique(x[x==names(y)])
  return(z)
}

str(legosets)
```

```
## tibble [6,172 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Item_Number : chr [1:6172] "10246" "10247" "10248" "10249" ...
##  $ Name        : chr [1:6172] "Detective's Office" "Ferris Wheel" "Ferrari F40" "Toy Shop" ...
##  $ Year        : int [1:6172] 2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ Theme       : chr [1:6172] "Advanced Models" "Advanced Models" "Advanced Models" "Advanced Models
##  $ Subtheme    : chr [1:6172] "Modular Buildings" "Fairground" "Vehicles" "Winter Village" ...
##  $ Pieces      : int [1:6172] 2262 2464 1158 898 13 39 32 105 13 11 ...
##  $ Minifigures : int [1:6172] 6 10 NA NA 1 2 2 3 2 2 ...
##  $ Image_URL   : chr [1:6172] "http://images.brickset.com/sets/images/10246-1.jpg" "http://images.br
##  $ GBP_MSRP    : num [1:6172] 132.99 149.99 69.99 59.99 9.99 ...
##  $ USD_MSRP    : num [1:6172] 159.99 199.99 99.99 79.99 9.99 ...
##  $ CAD_MSRP    : num [1:6172] 200 230 120 NA 13 ...
##  $ EUR_MSRP    : num [1:6172] 149.99 179.99 89.99 69.99 9.99 ...
##  $ Packaging   : chr [1:6172] "Box" "Box" "Box" "Box" ...
##  $ Availability: chr [1:6172] "Retail - limited" "Retail - limited" "LEGO exclusive" "LEGO exclusive
```

```
legosets$Item_Number<-replace_na(legosets$Item_Number,most.frequent(legosets$Item_Number))
legosets$Name<-replace_na(legosets$Name,most.frequent(legosets$Name))
legosets$Theme<-replace_na(legosets$Theme,most.frequent(legosets$Theme))
```

```
legosets$Subtheme<-replace_na(legosets$Subtheme,most.frequent(legosets$Subtheme))
legosets$Image_URL<-replace_na(legosets$Image_URL,most.frequent(legosets$Image_URL))
legosets$Packaging<-replace_na(legosets$Packaging,most.frequent(legosets$Packaging))
legosets$Availability<-replace_na(legosets$Availability,most.frequent(legosets$Availability))

legosets$GBP_MSRP<-replace_na(legosets$GBP_MSRP, median(legosets$GBP_MSRP, na.rm = TRUE))
legosets$USD_MSRP<-replace_na(legosets$USD_MSRP, median(legosets$USD_MSRP, na.rm = TRUE))
legosets$CAD_MSRP<-replace_na(legosets$CAD_MSRP, median(legosets$CAD_MSRP, na.rm = TRUE))
legosets$EUR_MSRP<-replace_na(legosets$EUR_MSRP, median(legosets$EUR_MSRP, na.rm = TRUE))

legosets$Year<-replace_na(legosets$Year, round(median(legosets$Year, na.rm = TRUE)))
legosets$Pieces<-replace_na(legosets$Pieces, round(median(legosets$Pieces, na.rm = TRUE)))
legosets$Minifigures<-replace_na(legosets$Minifigures, round(median(legosets$Minifigures, na.rm = TRUE)))

sum(is.na(legosets))
```

```
## [1] 0
```

Solution for question 11.3

```
star.wars<-function(x)
  {
  x%>%
    filter(Theme == "Star Wars")%>%
    summarize(rows = nrow(x), columns = ncol(x),
              mean = mean(GBP_MSRP, na.rm = TRUE), max = max(GBP_MSRP, na.rm = TRUE),
              min= min(GBP_MSRP, na.rm = TRUE), packaging =most.frequent(Packaging),
              availability = most.frequent(Availability))
  }

star.wars(legosets)
```

```
## # A tibble: 1 x 7
##    rows columns  mean   max   min packaging availability
##   <int>   <int> <dbl> <dbl> <dbl> <chr>     <chr>
## 1  6172      14  38.7  350.     0 Box       Retail
```

Solution for question 11.4

```
data1 <- legosets %>%
  select(Pieces, GBP_MSRP, Theme)%>%
  arrange(desc(Pieces), GBP_MSRP)

head(data1)
```

```
## # A tibble: 6 x 3
##   Pieces GBP_MSRP Theme
##    <dbl>    <dbl> <chr>
```

```
## 1   5922     200. Advanced Models
## 2   5195     342. Star Wars
## 3   4900     315. Serious Play
## 4   4287     210. Advanced Models
## 5   3803     275. Star Wars
## 6   3441     250. Star Wars
```

Solution for question 11.5

```
data2 <- legosets %>%
  filter(Theme == "City") %>%
  select(Pieces, GBP_MSRP, Availability)

head(merge(x = data1, y = data2, by = "Pieces", all.y = TRUE))
```

```
##   Pieces GBP_MSRP.x Theme GBP_MSRP.y Availability
## 1      2       2.99 Duplo      10.49       Retail
## 2      2       2.99 Duplo      10.49       Retail
## 3      2       2.99 Duplo      10.49       Retail
## 4      2       2.99 Duplo      10.49       Retail
## 5      2       2.99 Duplo      10.49       Retail
## 6      2       2.99 Duplo      10.49       Retail
```

Solution for question 11.6

```
library(data.table)
data1<-data.table(data1)
data2<-data.table(data2)
head(data1[data2, on = .(Pieces)])
```

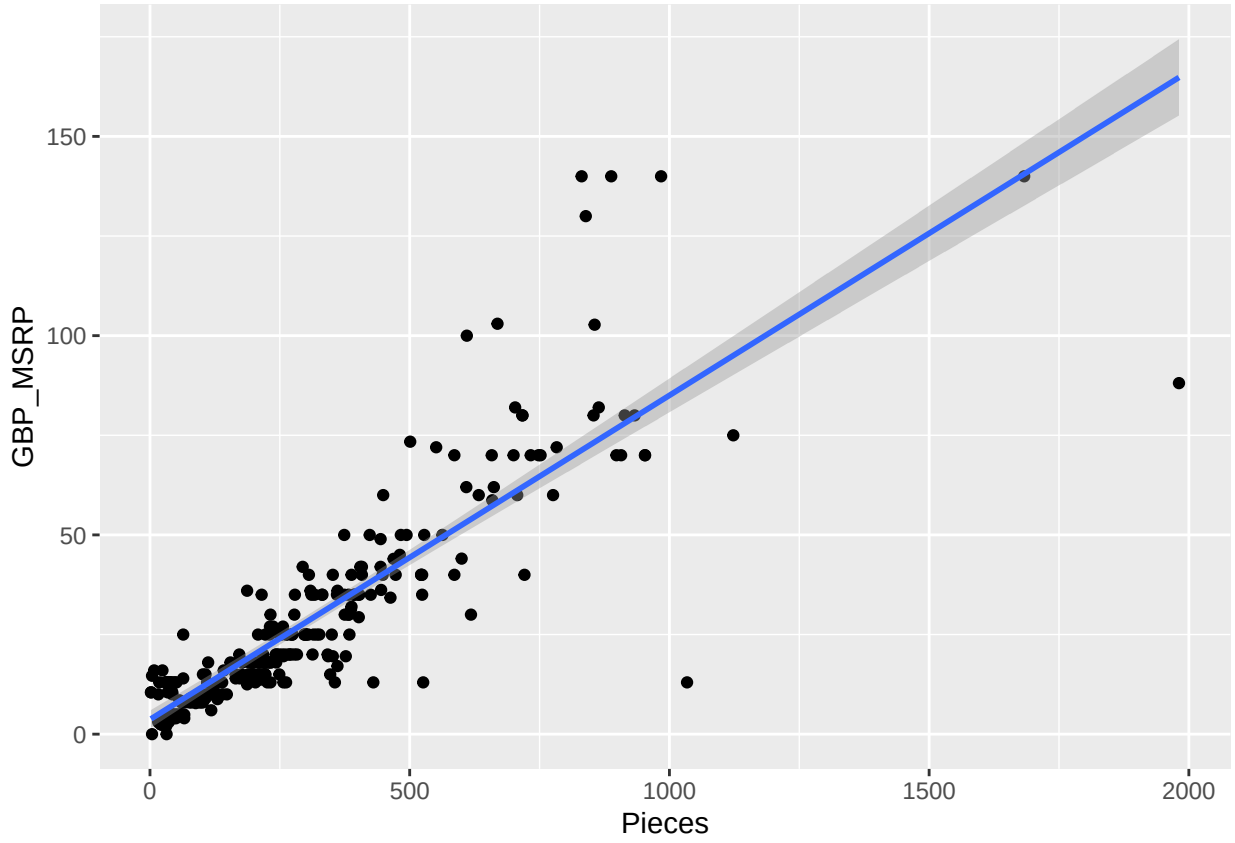
```
##    Pieces GBP_MSRP    Theme i.GBP_MSRP     Availability
## 1:     51     0.55 LEGOLAND      12.99 Retail - limited
## 2:     51     1.50     Town      12.99 Retail - limited
## 3:     51     2.99   Mixels      12.99 Retail - limited
## 4:     51     2.99   Mixels      12.99 Retail - limited
## 5:     51     2.99   Mixels      12.99 Retail - limited
## 6:     51     2.99 Seasonal      12.99 Retail - limited
```

Solution for question 11.7

```
 lm <- lm(data = data2, GBP_MSRP~Pieces)
```

Solution for question 11.8

```
ggplot(data = data2, aes(x = Pieces, y = GBP_MSRP))+
  geom_point() +
  geom_smooth(method = "lm")
```



Question 12

1. Use the Legosets dataset and create a data frame pieces_and_price which contains the columns Pieces and GBP_MSRP, and remove the missing values.
2. Using the median of Pieces, define a new indicator variable which takes the value of 0 if the number of pieces for the given item is less or equal to the median and 1 otherwise.
3. Calculate the correlation between Pieces and GBP_MSRP using Pearson's correlation coefficient.
4. Produce a scatterplot of the Pieces (X) vs GBP_MSRP across the levels of the factor defined in 12.2.
5. Produce a boxplot showing the distribution of GBP_MSRP across the levels of the factor defined in 12.2.
6. Are the variances of the prices of the two groups (below and above the median) homogeneous or not? Verify that using Fisher's F test (use the function var.test()).
7. Based on the results in 12.6, conduct a t-test, to test the null hypothesis that the price of items is equal for items that have less than median of pieces and those that have a higher number of pieces than the median against a two sided alternative.

Solution for question 12.1

```
pieces_and_price <- na.exclude(data.frame(legosets$Pieces, legosets$GBP_MSRP))
names(pieces_and_price) <- c("Pieces", "Price")
names(pieces_and_price)
```

```
## [1] "Pieces" "Price"
```

Solution for question 12.2

```
pieces_and_price<-mutate(pieces_and_price, Indicator = ifelse(Pieces <= median(Pieces), 0, 1))

head(pieces_and_price)
```

```
##   Pieces  Price Indicator
## 1   2262 132.99         1
## 2   2464 149.99         1
## 3   1158  69.99         1
## 4    898  59.99         1
## 5     13   9.99         0
## 6     39  16.99         0
```

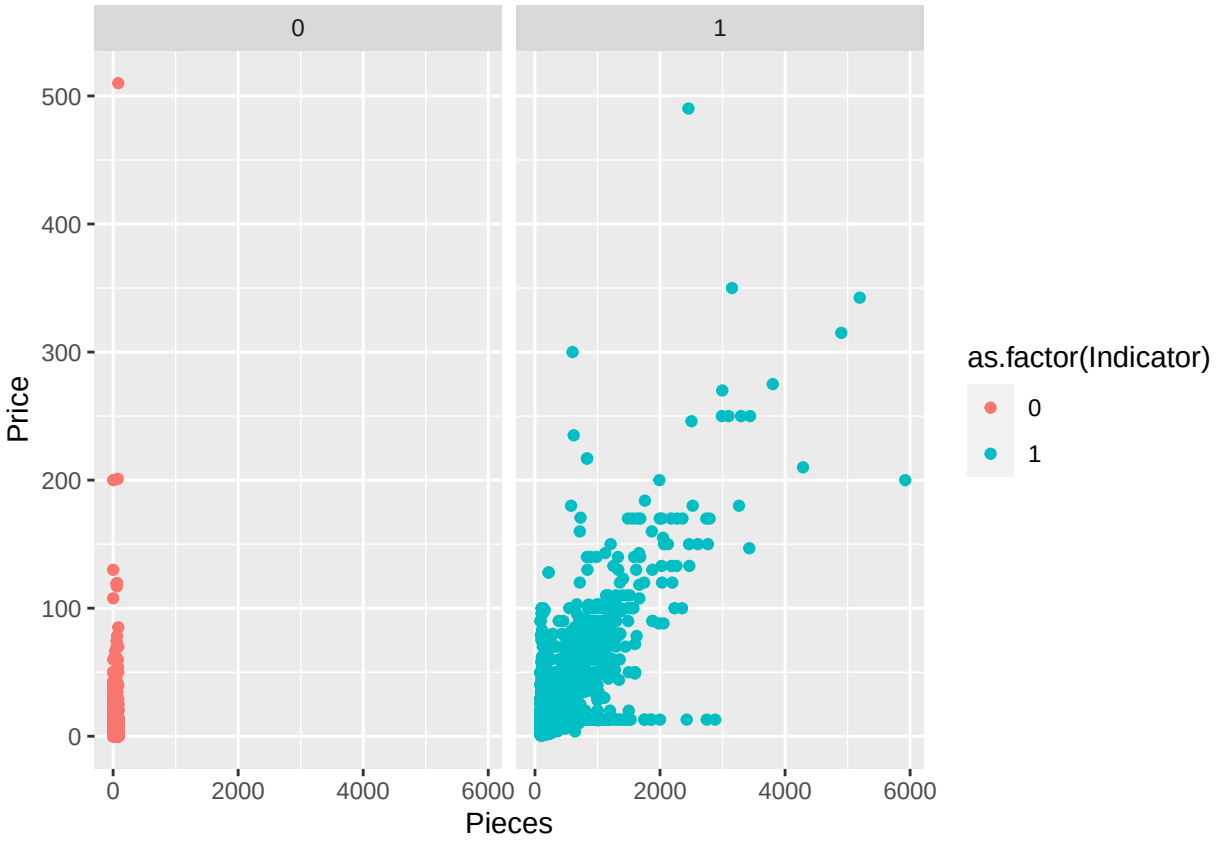
Solution for question 12.3

```
Pieces <- pieces_and_price$Pieces
Price <- pieces_and_price$Price
Indicator <- pieces_and_price$Indicator
cor(Pieces, Price)
```

```
## [1] 0.7300917
```

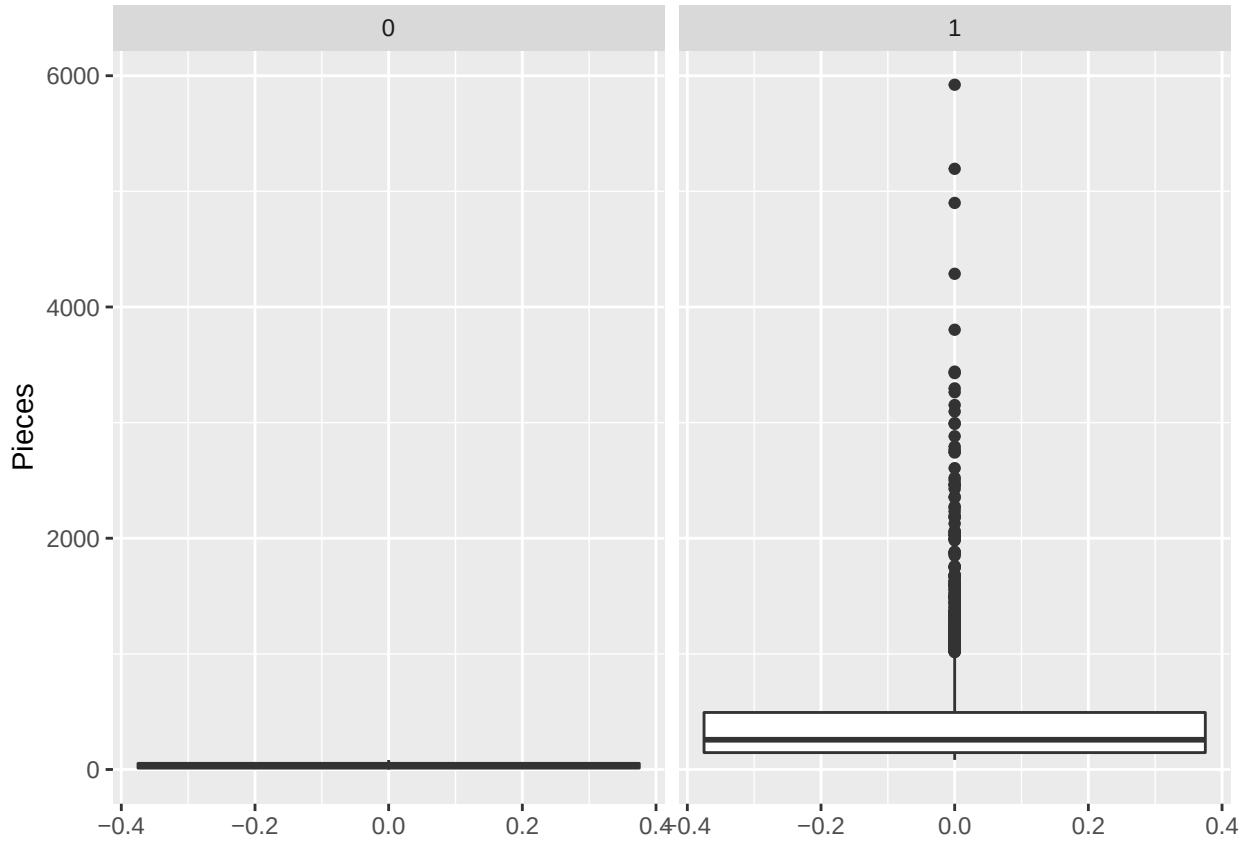
Solution for question 12.4

```
ggplot(data=pieces_and_price, aes(x = Pieces, y = Price))+
  geom_point(aes(col = as.factor(Indicator)))+
  facet_wrap(~Indicator)
```



Solution for question 12.5

```
ggplot(data=pieces_and_price, aes(y = Pieces))+
  geom_boxplot()+
  facet_wrap(~Indicator)
```



Solution for question 12.6

```
var.test(Pieces, Price)
```

```
##
##  F test to compare two variances
##
## data:  Pieces and Price
## F = 174.38, num df = 6171, denom df = 6171, p-value < 2.2e-16
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  165.8887 183.3004
## sample estimates:
## ratio of variances
##           174.3773
```

Solution for question 12.7

```
t.test(Price[Indicator == "0"], Price[Indicator == "1"], var.equal = FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  Price[Indicator == "0"] and Price[Indicator == "1"]
```

```
## t = -28.949, df = 4090.9, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -20.03149 -17.49039
## sample estimates:
## mean of x mean of y
##  10.92272  29.68366
```

Question 13

1. For the Legosets dataset, sum the price in USD_MSRP by the variable Theme. Identify the top 5 themes in terms of the highest total price (i.e., the top 5 most expensive themes with the highest sum).
2. Using the original Legosets data and the top 5 themes identified in Question 13.1, produce a density plot for the price in USD_MSRP across the level of the 5 themes that were identified in 13.1.

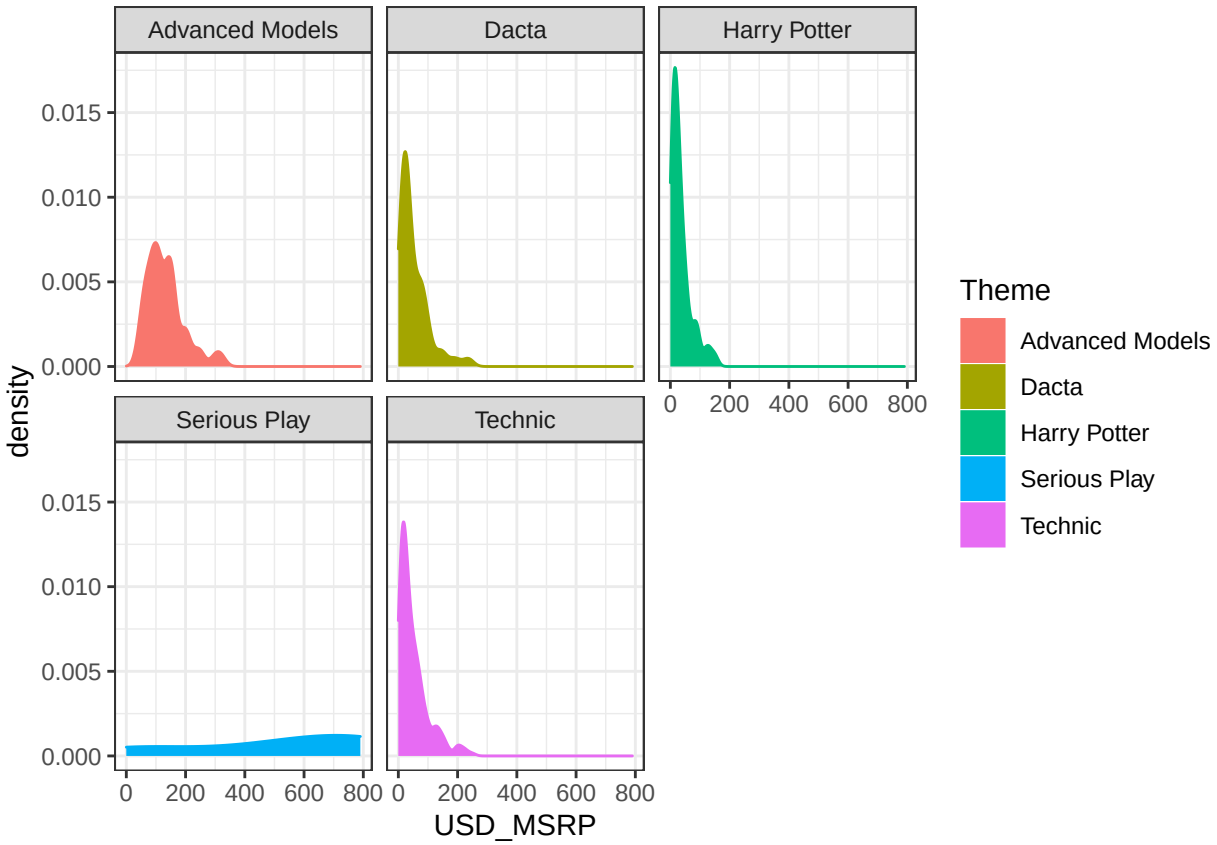
Solution for question 13.1

```
legosets.Theme <- legosets%>%
  group_by(Theme)%>%
  summarize(Price.Sum.USD = sum(USD_MSRP))%>%
  arrange(desc(Price.Sum.USD))%>%
  top_n(5, Price.Sum.USD)
legosets.Theme
```

```
## # A tibble: 5 x 2
##   Theme     Price.Sum.USD
##   <chr>             <dbl>
## 1 Star Wars        18437.
## 2 Technic          12142.
## 3 Duplo            11554.
## 4 City             10601.
## 5 Education         6480.
```

Solution for question 13.2

```
top.theme<-c("Technic", "Advanced Models", "Dacta", "Serious Play", "Harry Potter")

legosets.Density<-legosets%>%filter(Theme %in% top.theme)

legosets.Density%>%
ggplot(aes(x = USD_MSRP, col = Theme, fill = Theme))+
geom_density()+
facet_wrap(~Theme)+
theme_bw()
```



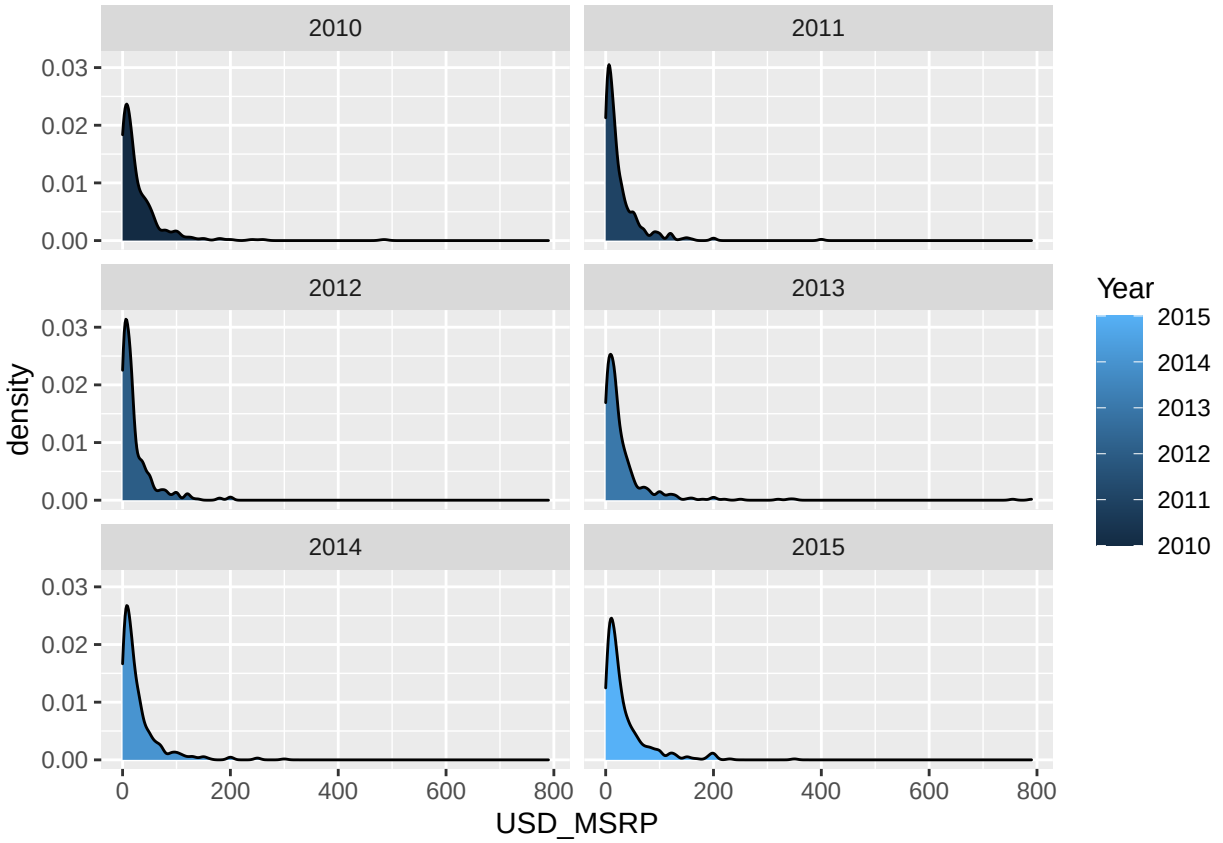
Question 14 The output below consists of a figure and a table with summary statistics for **the variable presented in Figure 14.1**.

1. Produce Figure 14.1 below.
2. Produce the table below.

- Hint 1: create a new data frame based on the variables presented in the figure.
- Hint 2: check missing values.

Solution for question 14.1

```
legosets.Year <- legosets%>%
select(USD_MSRP, Year)%>%
filter(Year >= 2010)

legosets.Year%>%
ggplot(aes(x = USD_MSRP,  fill = Year))+
geom_density()+
facet_wrap(~Year, nrow = 3)
```



Solution for question 14.2

```
legosets.Year.Complete <- na.omit(legosets)%>%
select(USD_MSRP, Year)%>%
filter(Year >= 2010)

legosets.Year.Complete%>%
group_by(Year)%>%
summarize(average = mean(USD_MSRP), standard_deviation = sd(USD_MSRP))
```

```
## # A tibble: 6 x 3
##    Year average standard_deviation
##   <dbl>   <dbl>              <dbl>
## 1  2010    30.3               45.6
## 2  2011    26.1               37.0
## 3  2012    23.9               31.9
## 4  2013    35.0               68.6
## 5  2014    29.3               37.8
## 6  2015    35.5               43.3
```

Part 4: the titanic data

In this part of the exam, we focus on titanic dataset. It contains data of survival status of passengers on the Titanic, together with their names (Name), age (Age), sex (Sex) and passenger class (PClass). This dataset is part of package lgrdata and will be available after installing the package. More information about the

data can be found in https://www.rdocumentation.org/packages/titanic/versions/0.1.0. To access the data you need to install the lgrdata package.

```
#install.packages("lgrdata")
library(lgrdata)
data(titanic)
dim(titanic)
```

```
## [1] 1313    5
```

```
names(titanic)
```

```
## [1] "Name"     "PClass"   "Age"      "Sex"      "Survived"
```

```
head(titanic)
```

```
##                                           Name PClass   Age    Sex Survived
## 1                 Allen, Miss Elisabeth Walton    1st 29.00 female        1
## 2                  Allison, Miss Helen Loraine    1st  2.00 female        0
## 3          Allison, Mr Hudson Joshua Creighton    1st 30.00   male        0
## 4 Allison, Mrs Hudson JC (Bessie Waldo Daniels)    1st 25.00 female        0
## 5                Allison, Master Hudson Trevor    1st  0.92   male        1
## 6                           Anderson, Mr Harry    1st 47.00   male        1
```

Question 15

1. For the analysis of this question, exclude all the observation with missing values in the titanic dataset. How many observations are included ? How many male and female there are among the passengers ?
2. Produce the 2×2 table below which shows the survival distribution (dead/alive) by gender.
3. Produce the data frame below. All your calculation should be done in R.
4. Produce Figure 15.1 and 15.2.

Solution 15.1

```
titanic.Complete <- na.omit(titanic)

addmargins(table(titanic.Complete$Sex),margin=1)
```

```
##
## female   male    Sum
##    288    468    756
```

Solution 15.2

```
table(titanic.Complete$Sex, titanic.Complete$Survived)
```

```
##
##            0   1
##   female  71 217
##   male   372  96
```

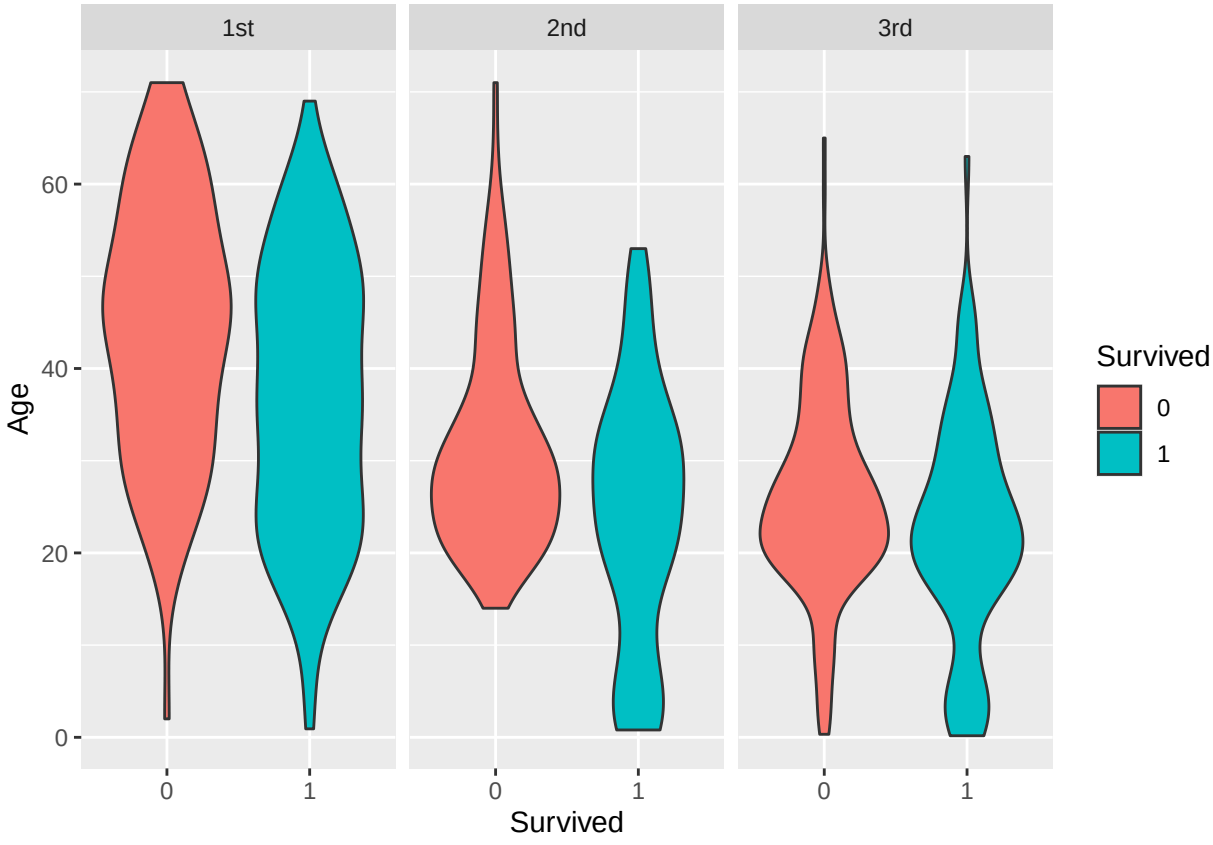
Solution 15.3

```
titanic.Death <- table(titanic.Complete$Sex,titanic.Complete$Survived)

titanic.Death <-addmargins(titanic.Death,margin=2)

colnames(titanic.Death) <- list("Died", "Survived", "%Died")
titanic.Death
```

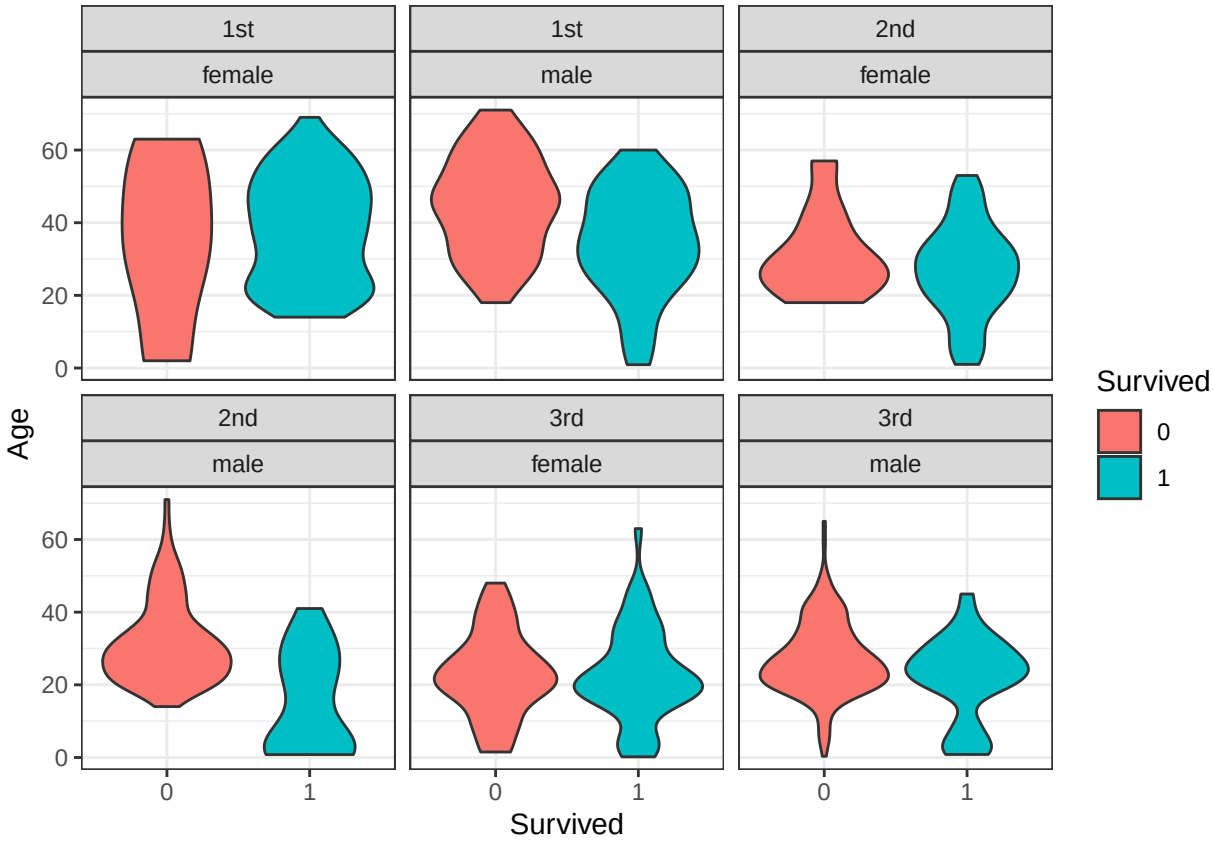
```
##
##          Died Survived %Died
##   female   71      217   288
##   male    372       96   468
```

Solution 15.4

```
titanic.Complete$Survived <- as.factor(titanic.Complete$Survived)

titanic.Complete%>%
ggplot(aes(x = Survived, y = Age, fill = Survived))+
geom_violin()+
facet_wrap(~PClass)
```



```
titanic.Complete%>%
ggplot(aes(x = Survived, y = Age, fill = Survived))+
geom_violin()+
facet_wrap(~PClass+Sex)+
theme_bw()
```



Question 16

1. For the analysis of this question, exclude all the observation with missing values in the titanic dataset. Create a new data frame which contains only female. How many observations are included ?
2. For the new data frame, calculate the mean age by class and create the data frame below.
3. Produce Figure 16.1 (density of age by class).
4. Test the hypothesis that (for female passengers) the mean age is equal across the classes using One-Way ANOVA model.
5. Let us define the standardized residuals as:

$$e_{si} = \frac{residual_i}{\sqrt{MSE}},$$

What is the estimate for the MSE obtained for the model in (3). Define a new R object that equal to e_{si} and produce Figure 16.2 (normal probability plot for the residuals).

Solution 16.1

```
dim(titanic.Female <- na.omit(titanic)%>%filter(Sex == 'female'))
```

```
## [1] 288   5
```

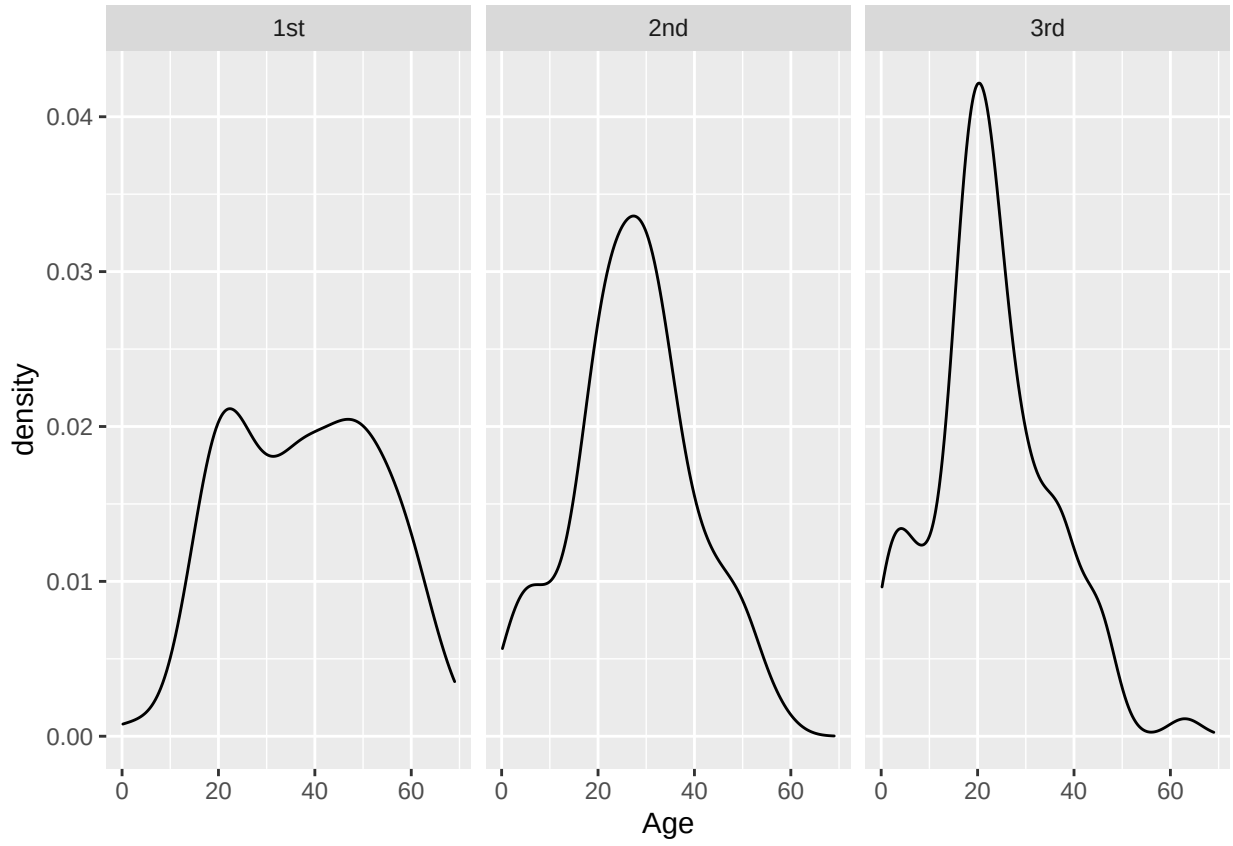
Solution 16.2

```
titanic.Female%>%
  group_by(PClass)%>%
  summarize(mean1 = round(mean(Age),digits = 1), median1 = median(Age))
```

```
## # A tibble: 3 x 3
##   PClass mean1 median1
##   <fct>  <dbl>   <dbl>
## 1 1st     37.8      38
## 2 2nd     27.4      28
## 3 3rd     22.8      22
```

Solution 16.3

```
titanic.Female%>%
  ggplot(aes(x = Age))+
  geom_density()+
  facet_wrap(~PClass)
```



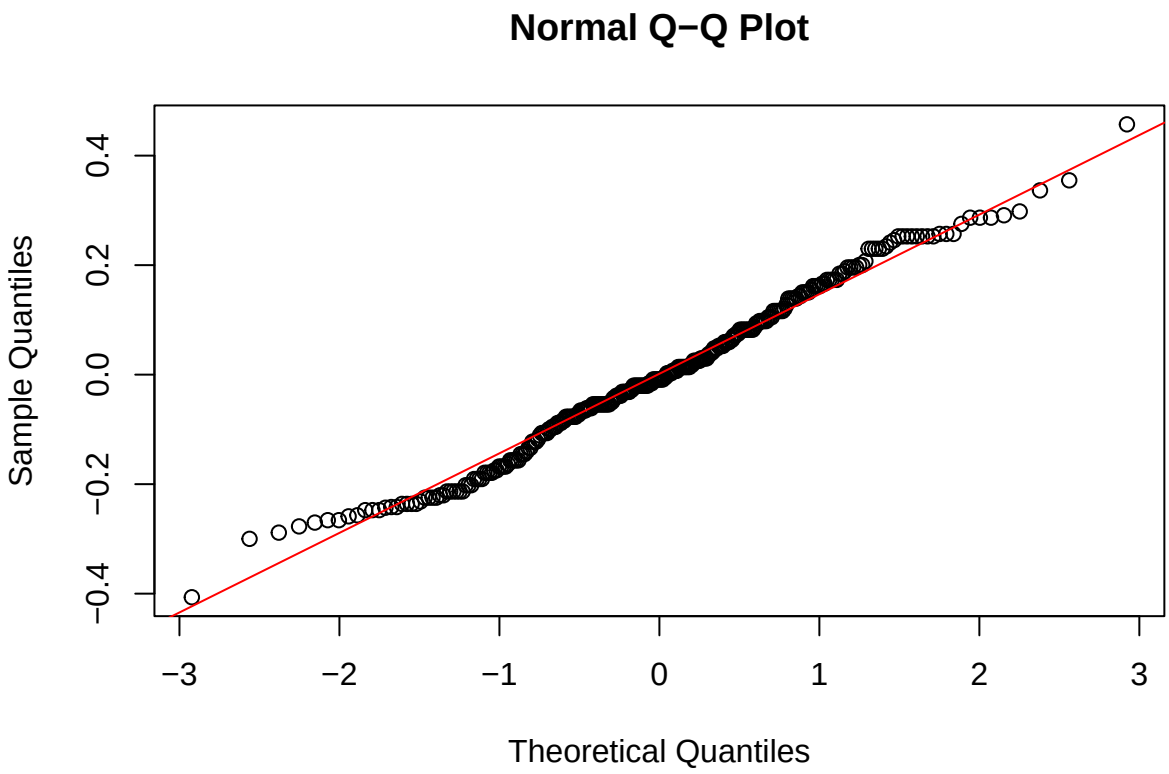
Solution 16.4

```
titanic.Female$PClass<- ordered(titanic.Female$PClass, levels = c("1st", "2nd", "3rd"))
fClass.aov <- aov(titanic.Female$Age~titanic.Female$PClass)
summary(fClass.aov)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## titanic.Female$PClass    2  11899    5949   33.45 8.89e-14 ***
## Residuals              285  50684     178
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Solution 16.5

```
lm.Female <- lm(Age~PClass, data = titanic.Female)
res.Female <- resid(lm.Female )
MSE.Female <- mean(lm.Female$residuals^2)
e.si <- (res.Female/ ((MSE.Female)^1/2))
qqnorm(e.si)
qqline(e.si, col = "red")
```


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−0.4
−0.2
0.0
0.2
0.4
−3
−2
−1
0
1
2
3