Final Exam (Long Answer)

Parker DeBruyne - V00837207

19/04/2022

Although not necessary, the dplyr package may save you some time if you know how to use it.

```
library(dbplyr)
```

Question 1: Consider the gapminder dataset (available by either loading into the R session or reading in the .csv file available in Brightspace).

(a) Create a new data frame called gapminder2007 which contains only data for the year 2007. DO NOT print the data.

```
gapminder = read.csv("gapminder.csv")
gapminder_index = which(gapminder$year == 2007)
gapminder2007 = gapminder[gapminder_index,]
```

(b) Create a table called continent2007 which gives the number of countries in each continent in gapminder2007.

```
continent2007 = table(gapminder2007$continent)
```

(c) Print continent2007.

```
continent2007
```

```
##
##   Africa Americas     Asia   Europe  Oceania
##       52       25       33       30        2
```

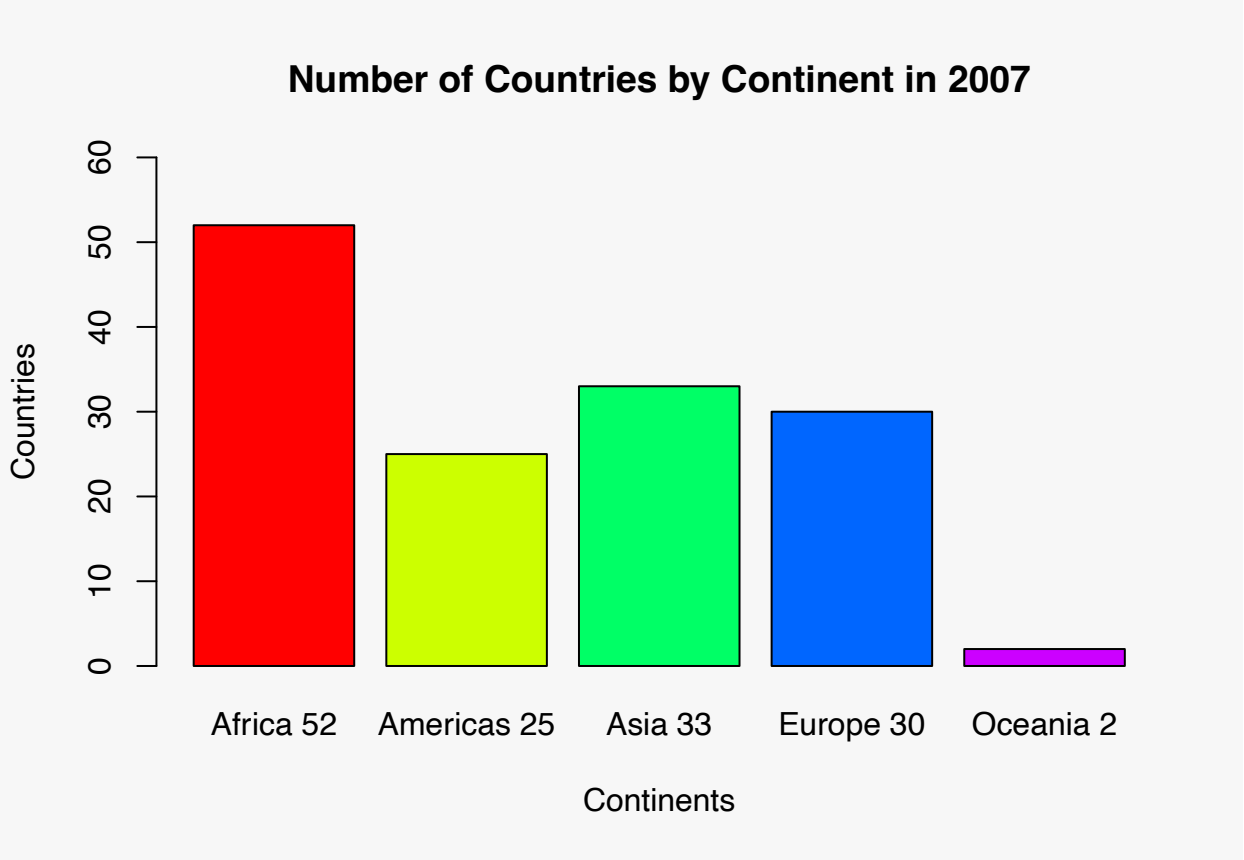
(d) Create a bar chart which displays the counts in continent2007. Make sure your graph has the following properties:

- a main title
- a different colour for each continent
- labels for each continent which include the counts.

```
labs = paste(names(continent2007), continent2007)


par(bg="grey97")
barplot(continent2007,
        main = "Number of Countries by Continent in 2007",
        xlab = "Continents",
        ylab = "Countries",
        ylim = c(0, 60),
```

```
        col = rainbow(length(names(continent2007))),
        names = labs)
```



Question 2: Consider again the gapminder data set.

(a) Create a variable called Europe_2007 which contains all of the rows of the gapminder data set corresponding to the continent Europe in the year 2007. You may subset the data in any way that you please.

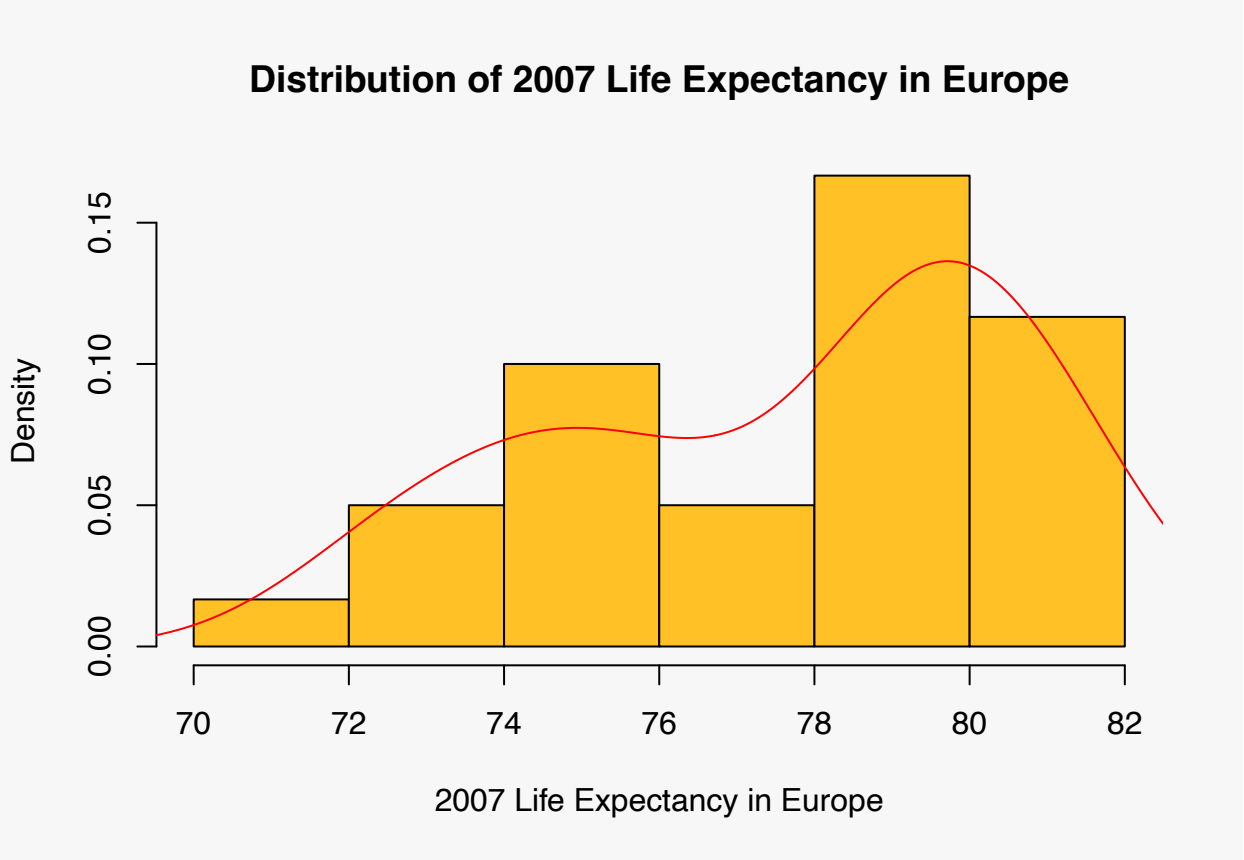
```
eu_index = which(gapminder2007$continent == "Europe")

Europe_2007 = gapminder2007[eu_index,]
```

(b) Plot the distribution of the life expectancy in European countries in 2007. You do not need any titles for your plot.

```
y = Europe_2007$lifeExp

par(bg="grey97")
hist(y,
     main="Distribution of 2007 Life Expectancy in Europe",
     xlab="2007 Life Expectancy in Europe",
     col="goldenrod1",
     prob=TRUE)
lines(density(y), col="red")
```



(c) Describe the shape of the distribution (symmetry, skewness, etc.).

```
# LEFT SKEWED. Not symmetric.
```

(d) What is the best measure of the centre of the distribution? Compute this value.

```
# MEDIAN.
median(Europe_2007$lifeExp)
```

```
## [1] 78.6085
```

(e) What is the best measure of the spread of the distribution? Compute the value(s).

```
#Quartiles Q1 and Q3
q1 = quantile(Europe_2007$lifeExp, 0.25)
q3 = quantile(Europe_2007$lifeExp, 0.975)

c(q1,q3)
```

```
##      25%    97.5%
## 75.02975 81.71640
```

(f) Suppose we are interested in a statistic that takes the minimum life expectancy value + the maximum life expectancy value and then divides that sum by 2. We will call this statistic "midpoint". Compute the observed value of the midpoint statistic for the sample of European life expectancies in 2007.

```
midpoint = (min(Europe_2007$lifeExp) + max(Europe_2007$lifeExp))/2
midpoint
```

```
## [1] 76.767
```

(g) Bootstrap 10000 sample midpoints of European life expectancies in 2007. Save the bootstrapped vector as boot_midpoint.

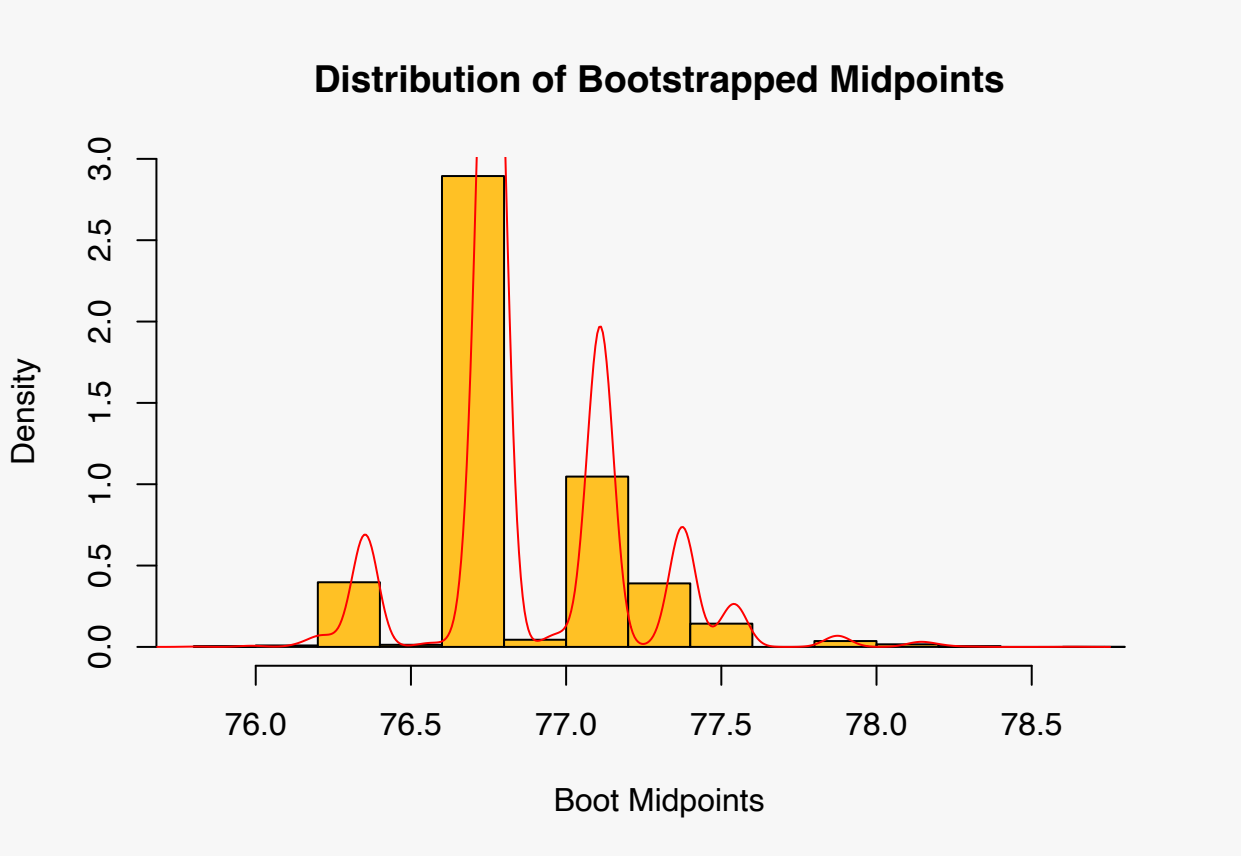
```
n = length(Europe_2007$lifeExp)
boot_midpoint = numeric()
for (i in 1:10000) {
  temp_samp = sample(Europe_2007$lifeExp, n, replace=TRUE)
  temp_midpoint = (min(temp_samp) + max(temp_samp))/2
  boot_midpoint[i] = temp_midpoint
}
mean(boot_midpoint)
```

```
## [1] 76.88146
```

** Note ** If you are unable to bootstrap this particular statistic, then bootstrap the median instead in order to be able to answer the remainder of the question.

(h) Plot the distribution of the bootstrapped midpoints. You do not need any titles for your plot.

```
par(bg="grey97")
hist(boot_midpoint,
     main="Distribution of Bootstrapped Midpoints",
     xlab="Boot Midpoints",
     col="goldenrod1",
     prob=TRUE)
lines(density(boot_midpoint), col="red")
```



(i) Describe the shape of the distribution. Does it appear normally distributed?

```
# No. The density curve is very jagged, not symetric, and not shaped like a bell.
```

(j) Compute a bootstrapped 90% confidence interval for the midpoint.

```
cv = qnorm(0.95)
ese = sd(boot_midpoint)/sqrt(n)

ci = c(mean(boot_midpoint) - cv*ese, mean(boot_midpoint) + cv*ese)
ci
```

```
## [1] 76.79093 76.97198
```

Question 3: Download, save, and read in the file "marketing.csv" from Brightspace. This dataset contains the advertising budget (in thousands of dollars) for three media outlets (youtube, facebook and newspaper) as well as the amount (in millions of dollars) of resulting sales (response).

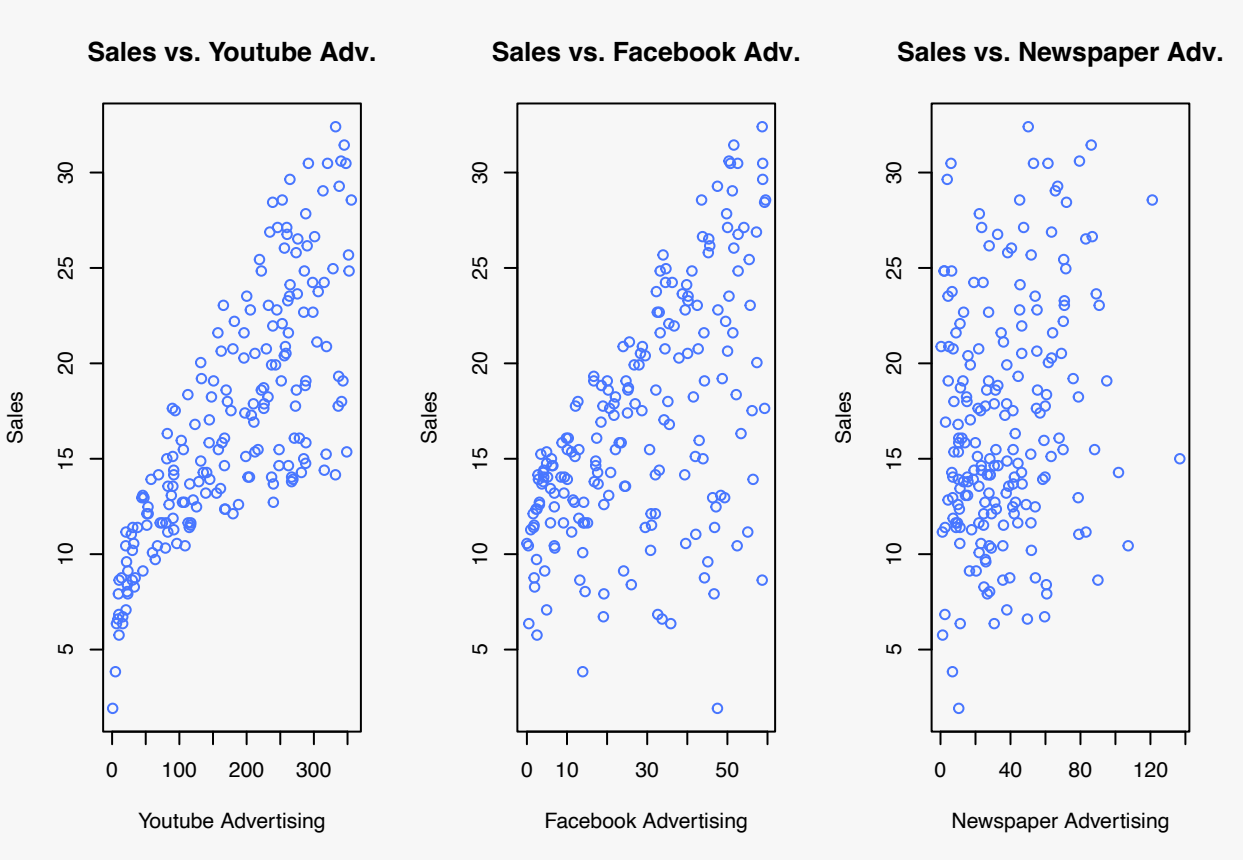
(a) Create 3 plots illustrating the relationship between the response variable and each of the explanatory variables.

```
mk = read.csv("marketing.csv")

y = mk$sales

yt = mk$youtube
fb = mk$facebook
np = mk$newspaper

par(bg="grey97")
par(mfrow=c(1,3))
plot(yt,y,
     main="Sales vs. Youtube Adv.",
     ylab="Sales",
     xlab="Youtube Advertising",
     col="royalblue1")
plot(fb,y,
     main="Sales vs. Facebook Adv.",
     ylab="Sales",
     xlab="Facebook Advertising",
     col="royalblue1")
plot(np,y,
     main="Sales vs. Newspaper Adv.",
     ylab="Sales",
     xlab="Newspaper Advertising",
     col="royalblue1")
```



(b) Fit a linear regression model including all of the explanatory variables. Be sure to write out the regression equation.

```
options(scipen=999)
model= lm(y~yt + fb + np)
summary(model)
```

```
##
## Call:
## lm(formula = y ~ yt + fb + np)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5932  -1.0690   0.2902   1.4272   3.3951
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept)  3.526667   0.374290   9.422 <0.0000000000000002 ***
## yt           0.045765   0.001395  32.809 <0.0000000000000002 ***
## fb           0.188530   0.008611  21.893 <0.0000000000000002 ***
## np          -0.001037   0.005871  -0.177                0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.023 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 0.00000000000000022
```

```
# y = 3.526667 + 0.045765*(yt) + 0.188530*(fb) - 0.001037*(np)
```

(c) Determine which variable(s) (if any) are not significant in the model using 0.05 as the criteria. Fit a new model including only significant variable(s). Write out the new regression equation.

```
significant_index = summary(model)$coefficients[,4] < 0.05
significant = names(summary(model)$coefficients[,4][significant_index])
significant
```

```
## [1] "(Intercept)" "yt"          "fb"
```

```
# y = 3.526667 + 0.045765*(yt) + 0.188530*(fb)
```

(d) Using the model from part (c) to predict the amount of sales for a company that spends 80 thousand on Youtube advertising, 46 thousand on Facebook advertising, and 50 thousand on newspaper advertising.

```
y = 3.526667 + 0.045765*(80) + 0.188530*(46)
print(paste(y, "(in millions of dollars)"))
```

```
## [1] "15.860247 (in millions of dollars)"
```