Stat 432 Homework 1

Giang Le (gianghl2)

Assigned: Aug 23, 2021; Due: 11:59 PM CT, Aug 31, 2021

Contents

Question 1 (random number generation and basic statistics)

X_1, X_2, ..., X_n are i.i.d. $\mathcal{N}(\mu, \sigma^2)$ random variables, where $\mu = 3$ and $\sigma = 2$.

a. Generate a set of $n = 100$ observations from this distribution. Only display the first 10 observations in your R output. Make sure that you set seed properly in order to replicate the result.

```
set.seed(13)
x = rnorm(100, mean = 3, sd = 2)
head(x, 10)
```

```
##  [1] 4.108654 2.439456 6.550327 3.374640 5.285052 3.831052 5.459013 3.473359
##  [9] 2.269234 5.210289
```

b. What is the statistical formula of the sample mean and sample variance (unbiased estimation)? Type the answer using latex.

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

$$s = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n - 1} \tag{2}$$

c. Calculate the above quantities (in b and c) using R functions. You need to use your own code to calculate these quantities and then match the results with default `R` functions.

My calculations are below

```
x_bar = sum(x)/100
x_bar
```

```
## [1] 2.876349
```

```
variance = sum((x - x_bar)^2) / (100 - 1)
variance
```

```
## [1] 3.616386
```

Here I use R default functions.

```
mean(x)
```

```
## [1] 2.876349
```

```
var(x)
```

```
## [1] 3.616386
```

d. Write a new function called `mysummarystat` that takes the data vector as the input, and output an vector of two elements: the sample mean and variance. Call the function using your data to validate.

```
mysummarystat <- function(input_vec) {
  mean <- sum(input_vec)/length(input_vec)
  variance <- sum((x - mean)^2) / (length(input_vec) - 1)
  output <- c(mean, variance)
  return(output)
}

mysummarystat(x)
```

```
## [1] 2.876349 3.616386
```

Question 2 (data manipulation, plots and linear model)

Perform the following tasks on the `iris` dataset. For each question, output necessary information to check that your completed the required operation.

a. Change the class labels of the `Species` variable from `virginica`, `versicolor`, and `setosa` to `Species_1`, `Species_2` and `Species_3`, respectively.

```
iris_data <- iris
head(iris_data)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
levels(iris_data$Species)
```

```
## [1] "setosa"     "versicolor" "virginica"
```

```
levels(iris_data$Species) <- c("Species_3", "Species_2", "Species_1")
levels(iris_data$Species)
```

```
## [1] "Species_3" "Species_2" "Species_1"
```

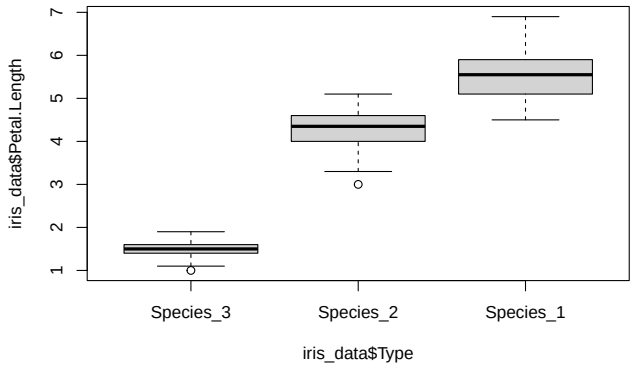
b. Change the variable name from `Species` to `Type`. Note that for both questions a) and b), you need to change the original variable, not creating a new variable and replacing the old one.

```
names(iris_data)[names(iris_data) == "Species"] <- "Type"
colnames(iris_data)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Type"
```

c. Create a boxplot for the variable `Petal.Length` that shows different boxes for different levels of `Type`. Adjust chunk options so that the plot is at the center and occupies 60% of the page width.

```
boxplot(iris_data$Petal.Length ~ iris_data$Type)
```



d. Use a linear model to estimate `Petal.Length` using all other four covariates. Make sure that the `Type` variable is specified as a factor. Report the coefficients and the most significant variable. To obtain the most significant variable, you must extract the p-value from the fitted object, instead of reading the value from the R output on your screen. If you do not know how to extract the p-value, use google to search for an answer with relevant keywords. Cite your reference by providing a link to it.

```
factor(iris_data$Type)
```

```
lm_model = lm(iris_data$Petal.Length ~ iris_data$Petal.Width + iris_data$Sepal.Length +
                      iris_data$Sepal.Width + iris_data$Type)
```

The coefficient estimates are

```
coef(summary(lm_model))[, "Estimate"]
```

```
##              (Intercept)   iris_data$Petal.Width  iris_data$Sepal.Length
##               -1.1109888               0.6022215               0.6080058
##   iris_data$Sepal.Width iris_data$TypeSpecies_2 iris_data$TypeSpecies_1
##               -0.1805236               1.4633709               1.9742229
```

The most significant variable is

```
sort(coef(summary(lm_model))[, "Pr(>|t|)"])[1]
```

```
## iris_data$Sepal.Length
##           1.073592e-23
```

Reference on how to sort: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/sort

e. Save the `iris` data into a `.csv` file, and then read the data from that file back into `R`. Make sure that the values in this new data is the same to the original one.

```
iris_orig <- iris
write.csv(iris_orig,'iris_data_data.csv')
iris_data_from_file <- read.csv('iris_data_data.csv')
head(iris_data_from_file)
```

```
##   X Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1 1          5.1         3.5          1.4         0.2  setosa
## 2 2          4.9         3.0          1.4         0.2  setosa
## 3 3          4.7         3.2          1.3         0.2  setosa
## 4 4          4.6         3.1          1.5         0.2  setosa
## 5 5          5.0         3.6          1.4         0.2  setosa
## 6 6          5.4         3.9          1.7         0.4  setosa
```