

Exam 3

**All screenshots that are included are from my own R Script files. You can access them with this link: <https://github.com/kiseraidan/STAT-3010/tree/main/Exam%203>
I will include the R script files in the submission as well.**

1.

(a).

Code:

```
12 # a.  
13  
14 # calc the mean, median, quantiles, and min/max for wing.length  
15 mean_wl <- mean(rubythroats$wing.length, na.rm = TRUE)  
16 median_wl <- median(rubythroats$wing.length, na.rm = TRUE)  
17 quantiles_wl <- quantile(rubythroats$wing.length, probs = c(0.25, 0.75),  
18                          na.rm = TRUE)  
19 min_wl <- min(rubythroats$wing.length, na.rm = TRUE)  
20 max_wl <- max(rubythroats$wing.length, na.rm = TRUE)  
21  
22 # print  
23 print(paste("Mean:", mean_wl))  
24 print(paste("Median:", median_wl))  
25 print(paste("First Quartile:", quantiles_wl[1]))  
26 print(paste("Third Quartile:", quantiles_wl[2]))  
27 print(paste("Min:", min_wl))  
28 print(paste("Max:", max_wl))
```

Output:

```
> source("~/Documents/Spring 2024 Classes/Statistics for Engineers & Scientists/Exams/Exam 3/Take Home/Exam3_Question1.R")
Rows: 78 Columns: 10
— Column specification —————
Delimiter: ","
chr (3): survival, nestling.fate, second.clutch
dbl (7): weight, wing.length, tarsus.length, first.clutch.size, carotenoid.chroma, bib.area, total.brig...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
[1] "Mean: 50.2948717948718"
[1] "Median: 50.5"
[1] "First Quartile: 49"
[1] "Third Quartile: 51.5"
[1] "Min: 46.5"
[1] "Max: 54"
> |
```

(b).

Code:

```
30 # b.
31
32 # calc the standard deviation for wing.length
33 sd_wl <- sd(rubythroats$wing.length, na.rm = TRUE)
34
35 # calc the correlation between wing.length and tarsus.length
36 correlation <- cor(rubythroats$wing.length, rubythroats$tarsus.length,
37                   use = "complete.obs")
38
39 # print
40 print(paste("Standard Deviation of Wing Length:", sd_wl))
41 print(paste("Correlation between Wing Length and Tarsus Length:", correlation))
```

Output:

```

> source("~/Documents/Spring 2024 Classes/Statistics for Engineers & Scientists/Exams/Exam 3/Take Home/Exam3_Question1.R")
Rows: 78 Columns: 10
— Column specification —————
Delimiter: ","
chr (3): survival, nestling.fate, second.clutch
dbl (7): weight, wing.length, tarsus.length, first.clutch.size, carotenoid.chro...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
[1] "Mean: 50.2948717948718"
[1] "Median: 50.5"
[1] "First Quartile: 49"
[1] "Third Quartile: 51.5"
[1] "Min: 46.5"
[1] "Max: 54"
[1] "Standard Deviation of Wing Length: 1.68922463085282"
[1] "Correlation between Wing Length and Tarsus Length: 0.258471110980964"
> |

```

(c).

Code:

```

43 # c.
44
45 # calc the 99% confidence interval for the average wing.length
46 ci <- t.test(rubythroats$wing.length, conf.level = 0.99)
47
48 # print
49 print(ci)
50 |

```

Output:

```
> source("~/Documents/Spring 2024 Classes/Statistics for Engineers & Scientists/Exams/Exam 3/Take Home/Exam3_Question1.R")
Rows: 78 Columns: 10
— Column specification —
Delimiter: ",",
chr (3): survival, nestling.fate, second.clutch
dbl (7): weight, wing.length, tarsus.length, first.clutch.size, carotenoid.chroma, bib.area, total.brightness

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
[1] "Mean: 50.2948717948718"
[1] "Median: 50.5"
[1] "First Quartile: 49"
[1] "Third Quartile: 51.5"
[1] "Min: 46.5"
[1] "Max: 54"
[1] "Standard Deviation of Wing Length: 1.68922463085282"
[1] "Correlation between Wing Length and Tarsus Length: 0.258471110980964"

One Sample t-test

data: rubythroats$wing.length
t = 262.96, df = 77, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99 percent confidence interval:
 49.78970 50.80005
sample estimates:
mean of x
 50.29487

> |
```

2.

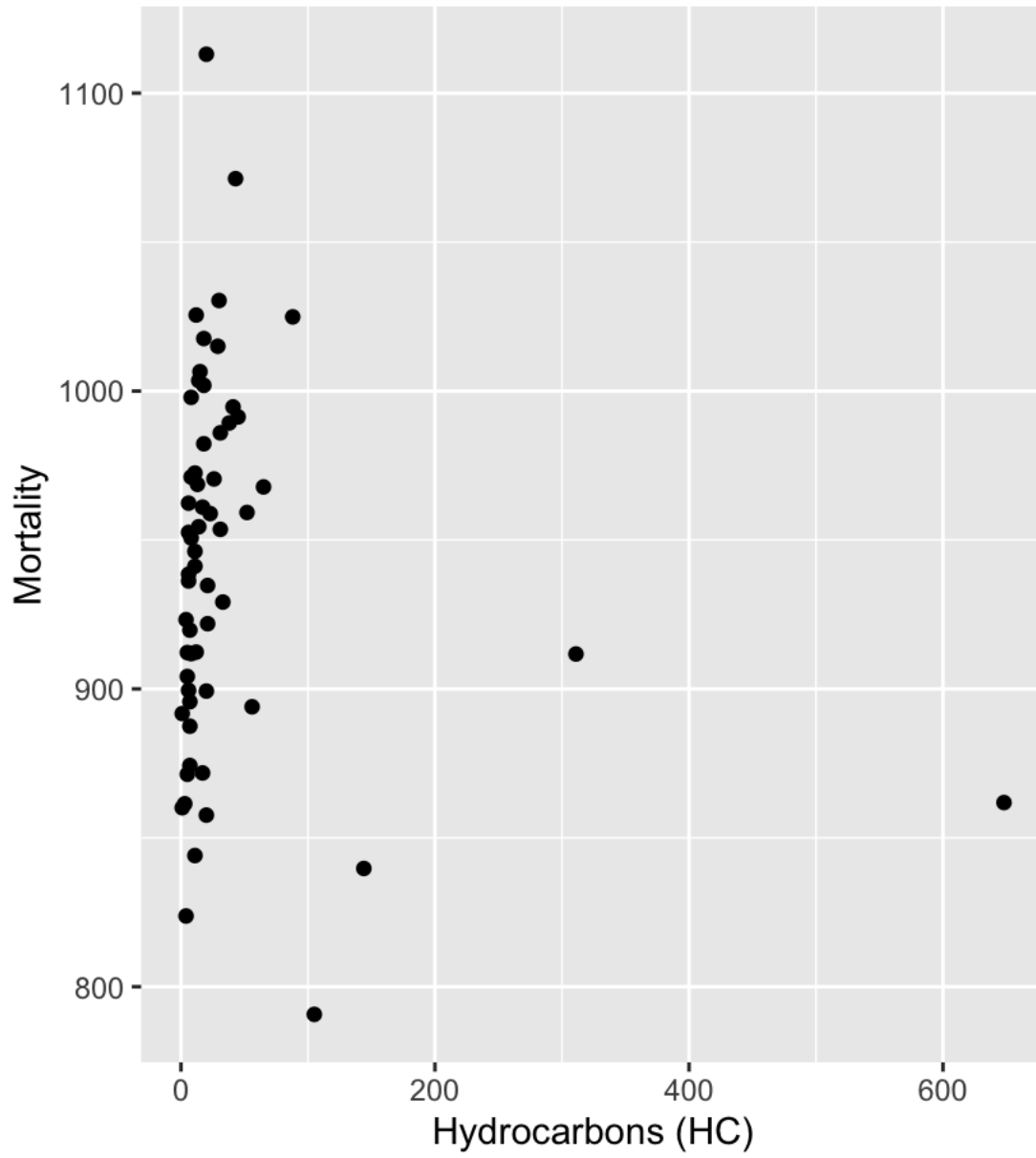
(a).

Code:

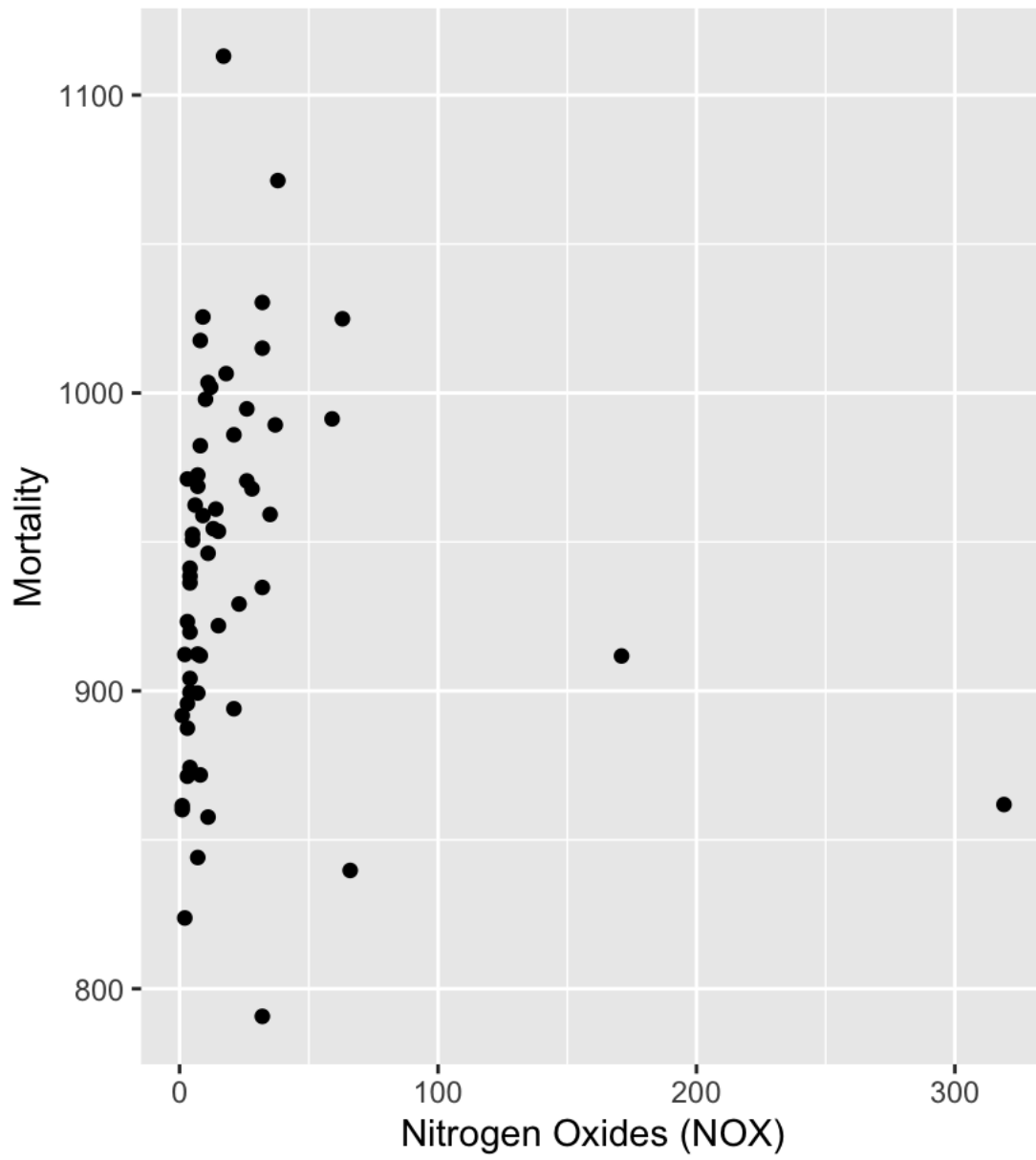
```
13 # a.
14
15 # create scatter plots
16 # plot for HC
17 p1 <- ggplot(mort_data, aes(x = HC, y = Mortality)) +
18   geom_point() +
19   labs(title = "Scatter plot of Mortality vs. HC", x = "Hydrocarbons (HC)", y = "Mortality")
20
21 # plot for NOX
22 p2 <- ggplot(mort_data, aes(x = NOX, y = Mortality)) +
23   geom_point() +
24   labs(title = "Scatter plot of Mortality vs. NOX", x = "Nitrogen Oxides (NOX)", y = "Mortality")
25
26 # plot for SO2
27 p3 <- ggplot(mort_data, aes(x = SO2, y = Mortality)) +
28   geom_point() +
29   labs(title = "Scatter plot of Mortality vs. SO2", x = "Sulfur Dioxide (SO2)", y = "Mortality")
30
31 # print the plots
32 print(p1)
33 print(p2)
34 print(p3)
```

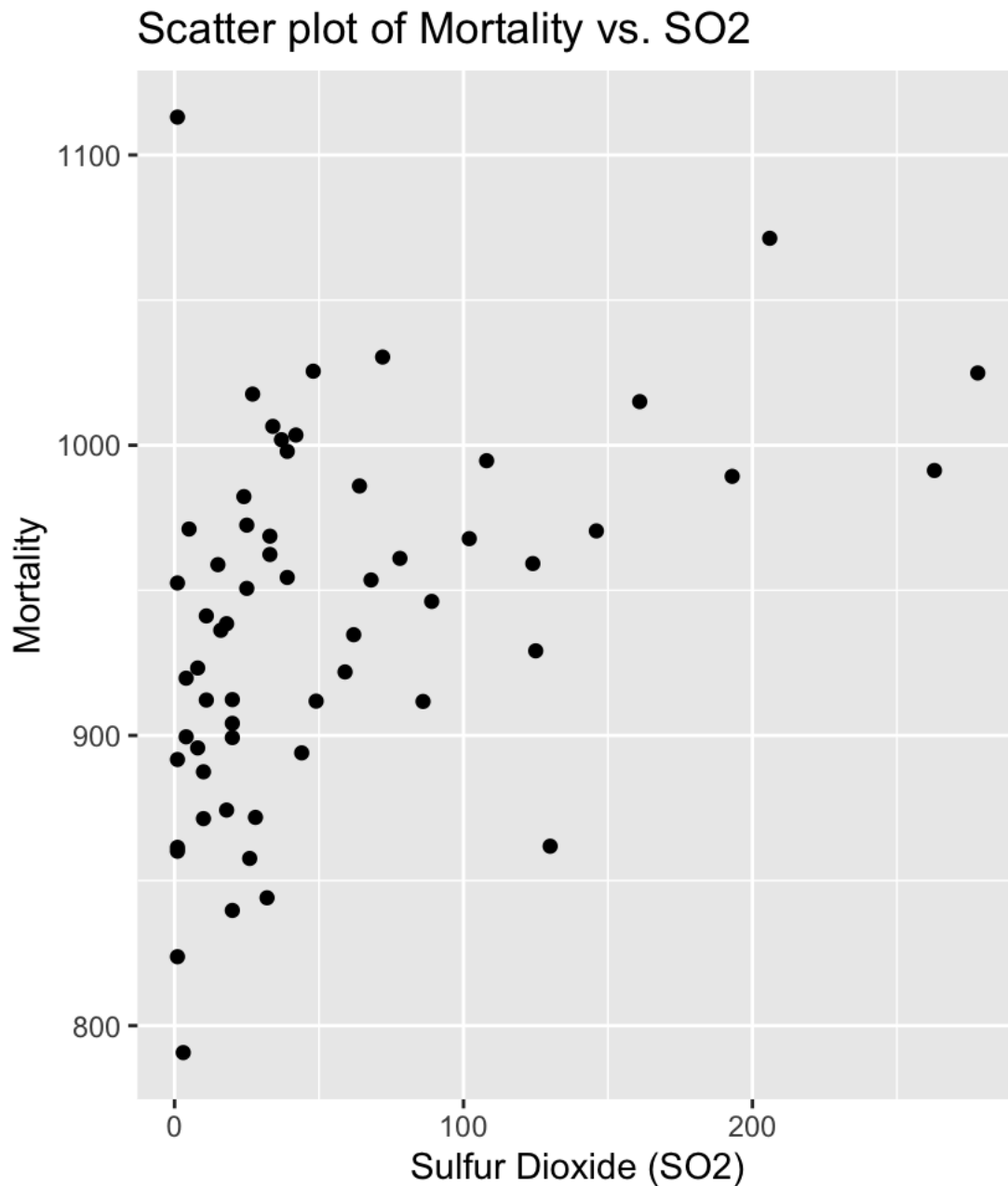
Plots:

Scatter plot of Mortality vs. HC



Scatter plot of Mortality vs. NOX





(b).

Code:

```
36 # b.  
37  
38 # run the linear regression with Mortality as the response and HC, NOX, SO2 as pre  
39 model <- lm(Mortality ~ HC + NOX + SO2, data = mort_data)  
40  
41 # summarize the model to get the regression equation and other statistics  
42 summary(model)
```

Output:

```
> summary(model)

Call:
lm(formula = Mortality ~ HC + NOX + SO2, data = mort_data)

Residuals:
    Min       1Q   Median       3Q      Max
-100.020  -33.053   -5.285   38.395  171.074

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  924.1631     8.9721  103.004  <2e-16 ***
HC           -1.6133     0.6068   -2.659   0.0102 *
NOX           2.9346     1.2666    2.317   0.0242 *
SO2           0.2007     0.1728    1.161   0.2505
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 51.84 on 56 degrees of freedom
Multiple R-squared:  0.3408,    Adjusted R-squared:  0.3055
F-statistic:  9.65 on 3 and 56 DF,  p-value: 3.123e-05
```

Therefore, The equation for the linear regression model with mortality as the response variable is:

$$\text{Mortality} = 924.1631 - 1.6133 \cdot \text{HC} + 2.9346 \cdot \text{NOX} + 0.2007 \cdot \text{SO2}$$

(c).

Code:

```
44 # c.
45
46 # retrieve the adjusted R-squared value from the model summary
47 adjusted_r_squared <- summary(model)$adj.r.squared
48
49 # print
50 adjusted_r_squared
```

Output:

```
> adjusted_r_squared
[1] 0.3054739
> |
```