# Statistical Methods with R: Final Assignment

Ronja Larsson

2022-11-17

## The Dataset: Iris

The dataset I choose to explore is the `iris` dataset, which is one of the default datasets in R. The dataset has already been cleaned so there is no need for cleaning before exploring the data. `iris` is pretty famous within data science community, so I would like to take a deeper look at it. `iris` contains the measurements in centermeters of three different iris species' sepal length and width, as well as petal length and width.

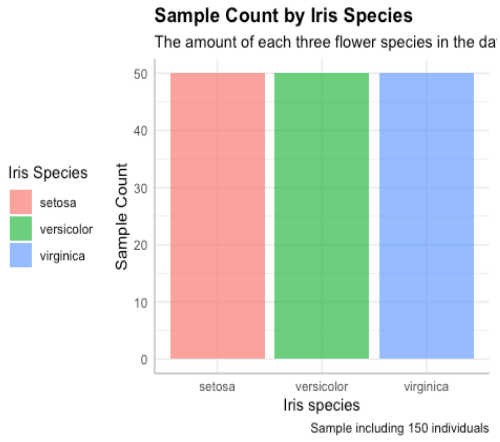
With the `str()` function, we can have an overview of the dataset's structure.

```
str(iris)
```

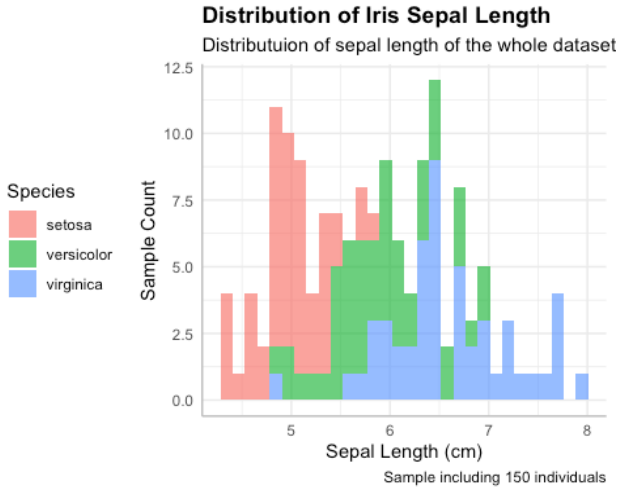
```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1
1 1 1 1 ...
```

There are 150 observations of five variables in `iris` dataset. Four of the variables have continuous, interval numbers, which makes them numerical/quantitative variable type. These four variables are `Sepal.Length`, `Sepal.Width`, `Patal.Length` and `Petal.Width`. The last variable is `Species` which is a nominal and categorical/qualitative variable type.

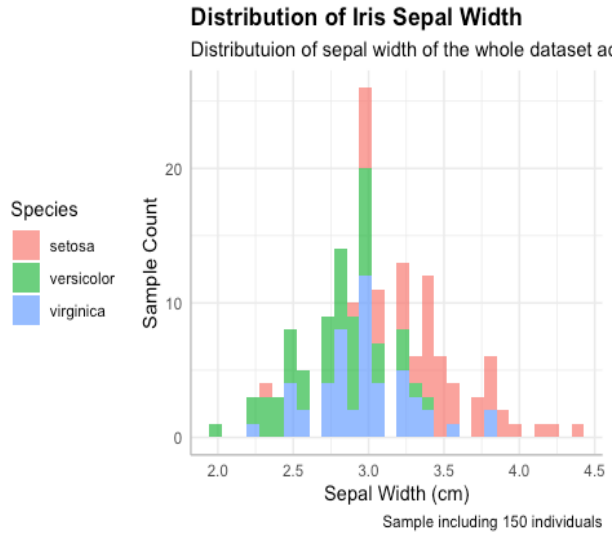
There are three species of iris flowers in the dataset, they are `setosa`, `versicolor` and `verginica`; and there are 50 individuals being observed from each specie. Below bar chart shows the count of each iris specie in the dataset.



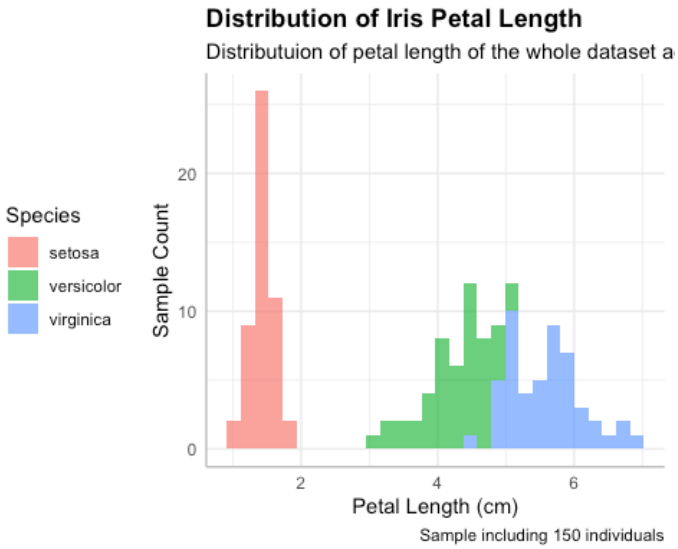
Let us take a look at the distributions of the quantitative variables in the dataset: `Sepal.Length`, `Sepal.Width`, `Petal.Length` and `Petal.Width`. I want to see how the distributions of the four quantitative variables look like across three iris species and see if there is anything that stands out for further exploration.



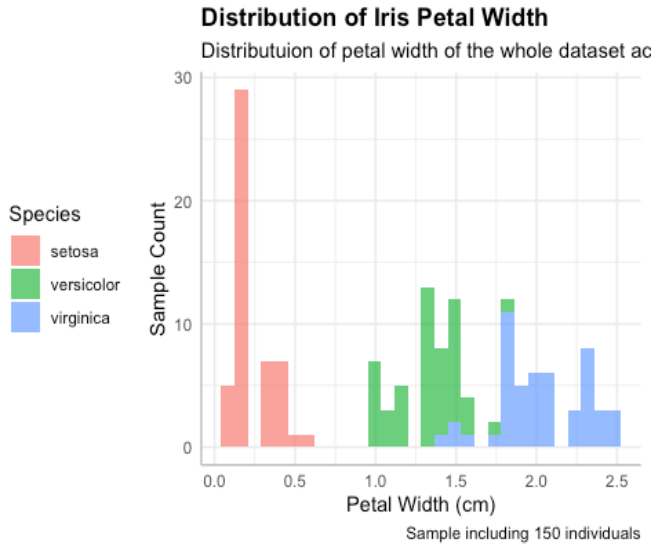

I see that there is a large overlapping part between the three species when it comes to sepal length, indicating that the sepal lengths are rather similar across all the three species.




Here again, I see that there is a large overlapping part between the three species when it comes to sepal width, indicating that the sepal widths are rather similar across all the three species.




In the case of petal length, I see that the length of Iris Setosa is much shorter than the other two species.




Here also, the petal width of Iris Setosa is also much narrower than the other two species.

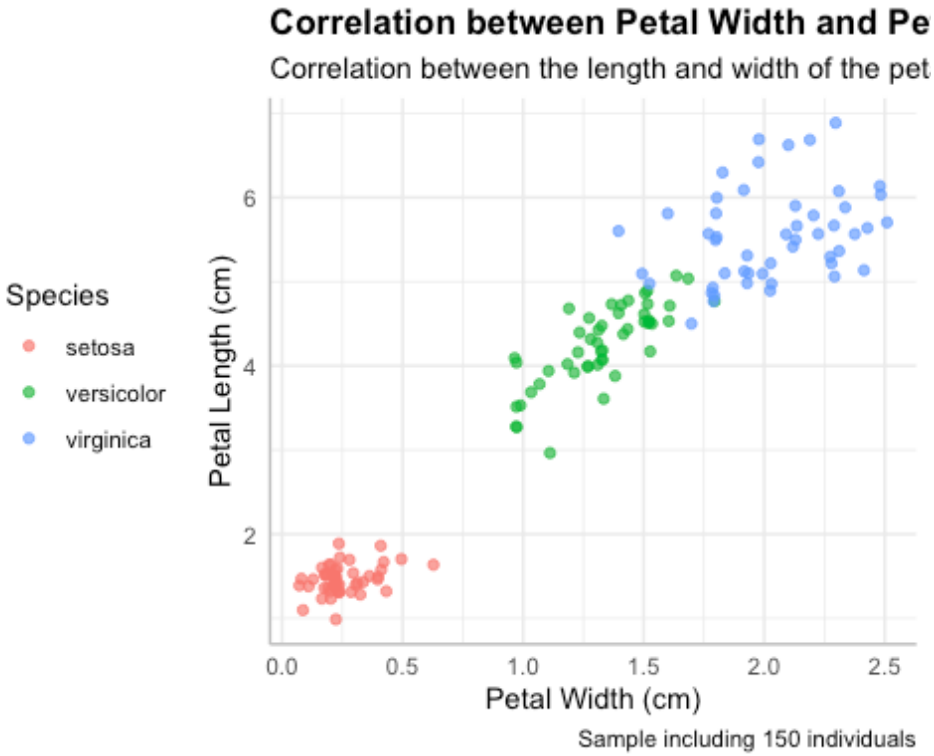
From above four histograms, I can clear see that petal length and width of Iris Setosa clearly stand out in terms of their distribution. I draw the conclusion that petal size is smaller for Iris Setosa and I want to look further into the variables `Petal.Length` and `Petal.Width`.

## A deeper look at the correlation of petal length and petal width

I want to take a deeper look at the correlation between the petal length and petal width of the three iris species to answer two questions:

1. **Is it true that Iris Setosa has the smallest petal, compared with Iris Versicolor and Iris Virginica?**
2. **Is there a positive correlation between petal length and petal width across the three species?**

A scatter plot will show me the answers of above two questions. In a scatter plot we can have `Petal.Length` on the x-axis and `Petal.Width` on the y-axis, then we can color-code the three species to see the relationship between petal length and petal width for each individual data point from all three species in a graph.



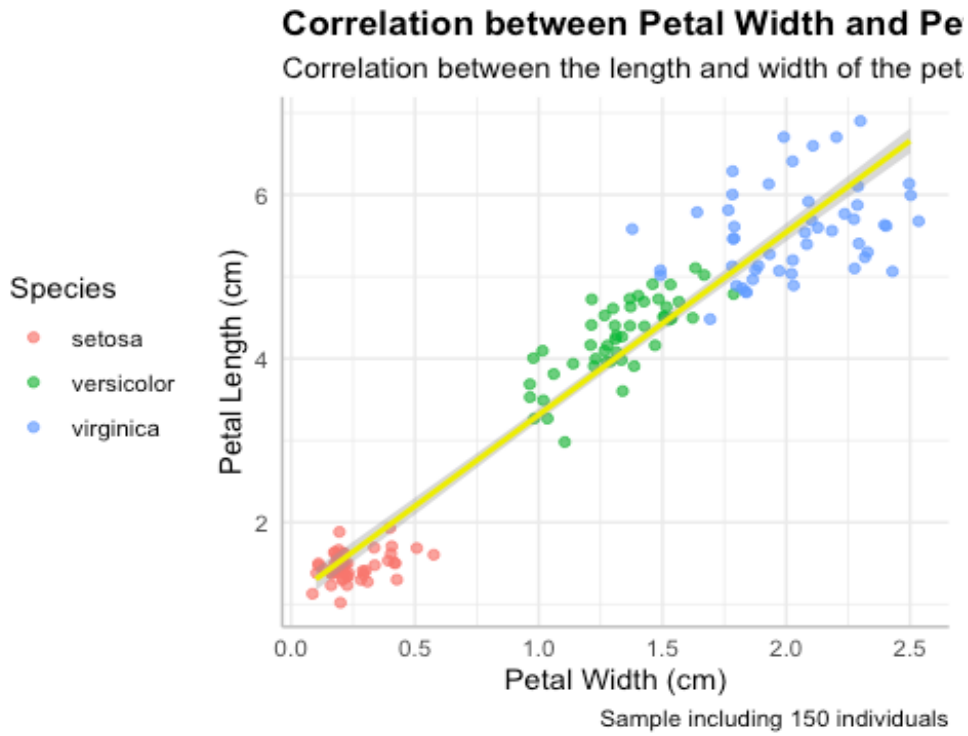

From above scatter plot, it is clear that **Iris Setosa indeed has the smallest petal**, and **there is a clear positive correlation between petal length and petal width** across the three species. That is to say, the longer the petal, generally the wider the petal.

Since there is a clear positive correlation between petal length and petal width, I want to add a trend line to be able to make prediction on the petal length based on the petal width. We can use Least Squares Regression Line (LSRL) to make the prediction.

LSRL is calculated with this general formula: **$\hat{y}=a+bx$**, where $\hat{y}$ is the predicted y-value given x, a is the y **intercept**, b and is the **slope**. For any given x-value, LSRL makes a predicted y-value, **$\hat{y}$**, that is close to the actual y-value, **y**, but not exactly the same y-value. The gap between the predicted y-value and the actual y-value is called residual. The LSRL goes through the data points and create a line which is most fitted, all data points considered, on the scatter plot. LSRL is the line of the smallest sum of the **residual squares**. Any other line than LSRL will have a greater sum of the residual squares. Therefore, LSRL is the most fitted trend line which can be used to make predictions.

Below code plots the same scatter plot as above, showing the correlation between petal length and petal width, but here I have added a LSRL, which is the yellow trend line.

```
ggplot(data = iris) +
  geom_jitter(mapping = aes(x = Petal.Width, y = Petal.Length, col =
Species), alpha = 0.7) +
  # below line of code adds the trend line to the graph
  geom_smooth(mapping = aes(x = Petal.Width, y = Petal.Length), col =
"yellow2", method = lm) +
  labs(title = "Correlation between Petal Width and Petal Length",
       subtitle = "Correlation between the length and width of the petal
across three species",
       caption = "Sample including 150 individuals") +
  ylab(label = "Petal Length (cm)") +
  xlab(label = "Petal Width (cm)") +
  theme_minimal() +
  theme(legend.position = "left",
        axis.line = element_line(color = "grey"),
        axis.line.x = element_line(color = "grey"),
        axis.line.y = element_line(color = "grey"),
        plot.title = element_text(face = "bold"))
```




In the code chunk `geom_smooth(mapping = aes(x = Petal.Width, y = Petal.Length), col = "yellow2", method = lm)` we have `method = lm`, which is the part of the code in ggplot2 that very conveniently adds the LSRL without me doing any manual calculation. To show the calculations behind the plot, we can create a model and get the intercept and the slope with below code:

```r
# modelling with lm()
# 1. we create a model, putting petal length at y-axis, petal width at x-axis
iris_model <- lm(iris$Petal.Length ~ iris$Petal.Width)

# 2. we get the intercept and slope with coef() function
e <- coef(iris_model)[1] # intercept
f <- coef(iris_model)[2] # slope

e # showing the intercept value

## (Intercept)
##    1.083558

f # showing the slope

## iris$Petal.Width
##          2.22994

# summary() provides insight of the model
summary(iris_model)

##
## Call:
## lm(formula = iris$Petal.Length ~ iris$Petal.Width)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33542 -0.30347 -0.02955  0.25776  1.39453
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.08356    0.07297   14.85   <2e-16 ***
## iris$Petal.Width   2.22994    0.05140   43.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4782 on 148 degrees of freedom
## Multiple R-squared:  0.9271, Adjusted R-squared:  0.9266
## F-statistic:  1882 on 1 and 148 DF,  p-value: < 2.2e-16
```

With `summary()` function, we can get more details of the LSRL:

- Residuals of all data points are provided here. The smaller the residuals are, the better fitted the trend line.
- We also see the intercept and slope values from `summary()` function, which are the same values as we got from `coef()` function.

- `Multiple R-squared:  0.9271` is the coefficient of determination in this case. `Adjusted R-squared:  0.9266` is the adjusted coefficient of determination, which gives a better representation, since it is adjusted with bias taken into consideration.

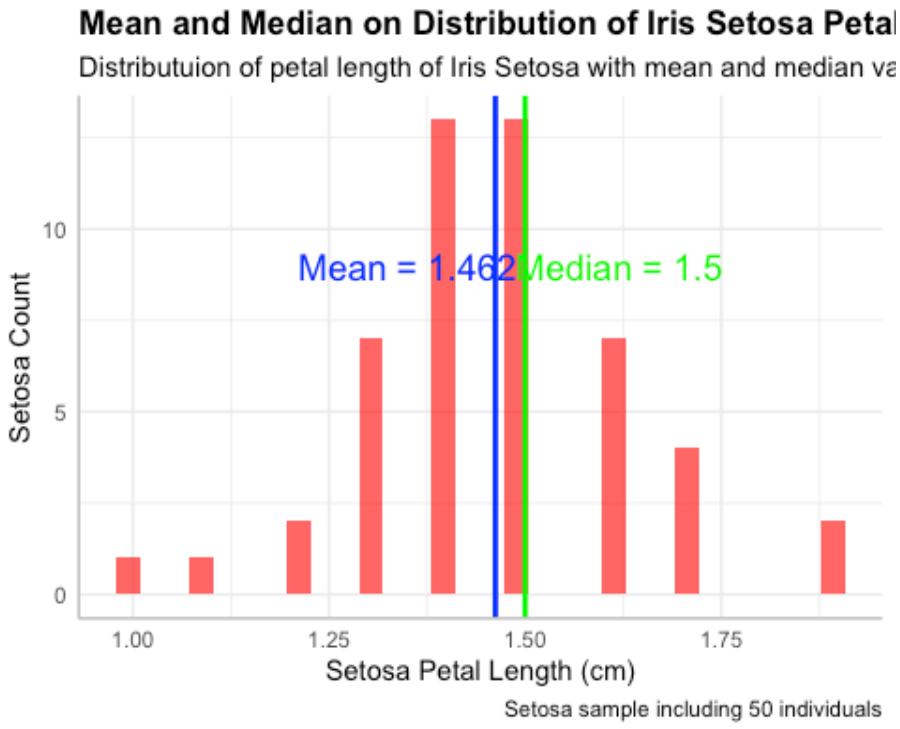
In this case, **R-squared** and **adjusted R-square** values are very close. The value of R-square is always between 0 and 1. The higher the number, the better. In other words, the better a model is at making predictions, the closer its R-square will be to 1 (100%). In this model, the coefficient of determination is **93%**, which means that the model is useful for making good predictions.

## A closer look at Iris Setosa

Based on above explorations, I want to take a closer look at Iris Setosa, which is the flower species with the smallest petals. I want to just look at the individuals that are from specie Setosa, so I use `subset()` to function separate the part of the data I want to look at. And then I find out the mean and median of the petal length for the 50 Setosa individuals.

```
# Separating setosa individuals from the dataset
my_setosa <- subset(iris, Species == "setosa")

# Calculating the mean of petal length
mean(my_setosa$Petal.Length)

## [1] 1.462

# Calculating the median of petal length
median(my_setosa$Petal.Length)

## [1] 1.5
```

After the calculations, I make a plot showing the distribution of setosa individuals' petal length, with the mean and median marked in the graph:



We can see that the length of Setosa petals are **normally distributed**. This is great! I want to use variable `Petal.Length` to do further analysis.

Next, I want to calculate the standard deviation for the graph above. The standard deviation of an observation variable is the square root of its variance. Therefore, I need to first calculate the variance:

```
# Calulation variance
variance <- (my_setosa$Petal.Length - mean(my_setosa$Petal.Length))^2
variance <- sum(variance) / (length(my_setosa$Petal.Length) - 1)
variance

## [1] 0.03015918
```

After getting the variance, I can get the standard deviation by using the `sqrt()` function.

```
# Calulation standard variation
std <- sqrt(variance)
std

## [1] 0.173664
```

A side note: In R, there is already a function for calculating the standard deviation. I use `sd()` function to check if my code to calculate standard deviation above is correct:

```
sd(my_setosa$Petal.Length)

## [1] 0.173664
```

The output from `sd()` function and my code above's output of variable `std` are the same - they both are **0.173664**!

Now that we have the mean for Setosa petal length: **1.1462**, and the standard deviation of Setosa petal length: **0.173664**, we can move on to calculate the z-score for the VG attempt of this assignment.

A z-score is a numerical measurement that describes a value's relationship to the mean of a group of values. z-score is measured in terms of standard deviations from the mean. If a z-score is 0, it indicates that the data point's score is identical to the mean score. A positive z-score indicates the raw score is higher than the mean average. For example, if a z-score is equal to +1, it is 1 standard deviation above the mean. A negative z-score reveals the raw score is below the mean average. For example, if a z-score is equal to -2, it is 2 standard deviations below the mean.

I want to choose the mode of the length of the petal of Setosa as the x-value for the z-score calculation. The questions I want to answer are:

1. **What is the mode value of the above graph?**
2. **Of all the data points in the dataset for Iris Setosa, what is the probability that the mode value is the length of the petal?**

In R, there is no inbuilt function to calculate the mode, so we have to manually write below function to find out the mode of the petal length for Iris Setosa, to answer the first question.

```
# There is no built in function to find mode in R, so we write one
mode <- function(x) {
  u <- unique(x)
  tab <- tabulate(match(x, u))
  u[tab == max(tab)]
}
```

From above histogram `Mean and Median on Distribution of Iris Setosa Petal Length`, we can see that there are two modes of the Satosa petal length: **1.4** and **1.5**. With above `mode()` function, we can check if the modes are indeeed 1.4 and 1.5, just as a control check.

```
mode(my_setosa$Petal.Length)

## [1] 1.4 1.5
```

It is confirmed that there are two mode values, they are **1.4** and **1.5**.

So, I will use both 1.4 and 1.5 as our x-values that we want to convert to a z-score, to answer the second question. The formula to calculate z-score is: **z=(x-mean)/s**.

When **x=1.4**, our calculation is as below:

```
# z-score of setosa petal length when x=1.4
z <- (1.4 - mean(my_setosa$Petal.Length))/sd(my_setosa$Petal.Length)
z

## [1] -0.3570112

# a calculation with the number values
control_z <- (1.4 - 1.462)/0.173664
control_z

## [1] -0.3570112
```

When x=1.4, the z-score is **-0.3570112**. Negative z-scores are below the mean. I use **the Area To The Left of Z-Score Calculator** online to check the probability of 1.4 (one of the two modes) being the length of the petal. The online calculator gives **36.05%** of the probability that the petal length would be 1.4 cm, when we enter the negative z-score -0.3570112.

Let us quickly calculate when x=1.5, which is the other mode value for above graph:

```
# z-score of setosa petal length when x=1.4
z <- (1.5 - mean(my_setosa$Petal.Length))/sd(my_setosa$Petal.Length)
z

## [1] 0.2188133
```

```r
# a calculation with the number values
control_z <- (1.5 - 1.462)/0.173664
control_z
```

```
## [1] 0.2188133
```

When x=1.5, the z-score is **0.2188133**. Positive z-scores are above the mean. I use **the Area To The Left of Z-Score Calculator** online to check the probability of 1.5 (one of the two modes) being the length of the petal. The online calculator gives **58.66%** of the probability that the petal length would be 1.5 cm, when we enter the positive z-score 0.2188133.

In both cases of x=1.4 and x=1.5, the z-scores are close to zero. This means that the z-score is about the same as the mean, which is correct. The mean of the length of petal for Setosa is **1.462**, which is between our both x-values **1.4** and **1.5**.

## Final comments

I started the assignment with exploring different datasets without having any specific question in mind. When I decided on using the `iris` dataset, I still did not have a question in mind. In fact, I found it very difficult to come up any question, without getting to know the dataset first.

After exploring the dataset, I let the graphs lead me by finding what I found interesting and focus on it. Slowly I could come up with questions which are the things I was curious about. I introduced the dataset and gave brief information on the variables.Then I started plotting different graphs and from them I proposed the initial questions I wanted to answer. Along the process of answering the questions, I provided my thinking process as best I could to show how I analyzed and answer the questions.

Along the way, more questions came up. This was actually really fun and I feel I can still do more, but due to the page limit, my analysis stopped here. Thank you for reading my report :)