DAY 4 ASSIGNMENT PART 2

1. Suppose that the data for analysis includes the attribute age. The age values for the data

tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33,

33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70. What is the median?

ANS:

To find the median of the given data set, we need to find the middle value. Since there are 26 values in the set, the middle two values will be used to calculate the median.

First, we need to arrange the data in ascending order:

13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70

The middle two values are 25 and 30, so we need to take the average of these two values to find the median:

Median = (25 + 30) / 2 = 27.5

Therefore, the median of the given data set is 27.5.

2. Suppose that the data for analysis includes the attribute age. The age values for the data

tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33,

33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

ANS:

2)To further analyze the data set with the attribute age, we can calculate some basic statistical measures such as the mean, mode, variance, and standard deviation.

The mean (average) can be calculated by summing all the values and dividing by the total number of values:

Mean = (13 + 15 + 16 + 16 + 19 + 20 + 20 + 21 + 22 + 22 + 25 + 25 + 25 + 25 + 30 + 33 + 33 + 35 + 35 + 35 + 35 + 36 + 40 + 45 + 46 + 52 + 70) / 26

= 28.769

Therefore, the mean age of the data set is approximately 28.769.

The mode is the value that appears most frequently in the data set. In this case, the value 25 appears four times, which is more than any other value. Therefore, the mode is 25.

The variance is a measure of how spread out the data is. It can be calculated by taking the average of the squared differences between each value and the mean:

Variance = [(13-28.769)^2 + (15-28.769)^2 + (16-28.769)^2 + ... + (70-28.769)^2] / 26

= 338.351

The standard deviation is the square root of the variance:

Standard deviation = sqrt(338.351)

= 18.395

Therefore, the standard deviation of the data set is approximately 18.395.

These measures can provide useful insights into the distribution of the data and help to identify any potential outliers or patterns.

3.Load iris Dataset which is inbuilt in R .explore the dataset in terms of dimension and

summary statistics

ANS:

```
head(iris)

str(iris)

summary(iris)

df <- iris[, 1:4]

boxplot(df)

pairs(df)

stars(df)

PL <- df$Petal.Length
```

barplot(PL)

hist(PL)

SP <- iris$Species

pie(table(SP))

boxplot(PL ~ SP)

summary(aov(PL ~ SP))

PW <- df$Petal.Width

plot(PL, PW, col = SP)

abline(lm(PW ~ PL))

4.Find the categorical column data and convert that to factor form, also find the number of

rows for each factors in dataset.

ANS:

Factors are the final major data structure we will introduce in our R genomics lessons. Factors can be thought of as vectors which are specialized for categorical data. Given R's specialization for statistics, this make sense since categorial and continuous variables are usually treated differently. Sometimes you may want to have data treated as a factor, but in other cases, this may be undesirable.

Let's see the value of treating some of which are categorical in nature as factors. Let's take a look at just the alternate alleles

```
## extract the "ALT" column to a new object

alt_alleles <- subset$ALT
```

Let's look at the first few items in our factor using `head()`:

```
head(alt_alleles)

[1] "G"         "T"         "T"         "CTTTTTTTT" "CCGCGC"    "T"
```

There are 801 alleles (one for each row). To simplify, lets look at just the single-nuleotide alleles (SNPs). We can use some of the vector indexing skills from the last episode.

```
snps <- c(alt_alleles[alt_alleles=="A"],
  alt_alleles[alt_alleles=="T"],
  alt_alleles[alt_alleles=="G"],
  alt_alleles[alt_alleles=="C"])
```

This leaves us with a vector of the 701 alternative alleles which were single nucleotides. Right now, they are being treated a characters, but we could treat

them as categories of SNP. Doing this will enable some nice features. For example, we can try to generate a plot of this character vector as it is right now:

```
plot(snps)
```

```
Warning in xy.coords(x, y, xlabel, ylabel, log): NAs introduced by coercion
```

```
Warning in min(x): no non-missing arguments to min; returning Inf
```

```
Warning in max(x): no non-missing arguments to max; returning -Inf
```

```
Error in plot.window(...): need finite 'ylim' values
```

Whoops! Though the `plot()` function will do its best to give us a quick plot, it is unable to do so here. One way to fix this it to tell R to treat the SNPs as categories (i.e. a factor vector); we will create a new object to avoid confusion using the `factor()` function:

```
factor_snps <- factor(snps)
```

Let's learn a little more about this new type of vector:

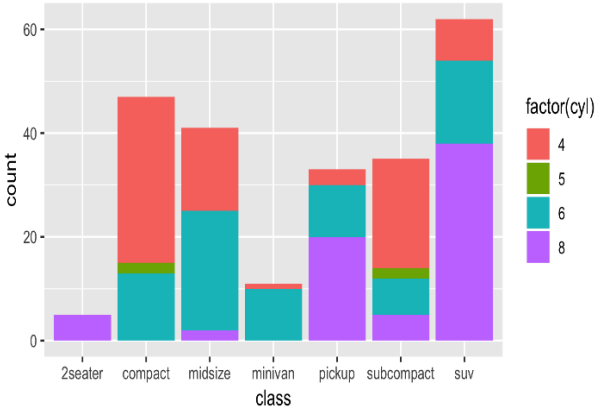
```
str(factor_snps)
```

```
 Factor w/ 4 levels "A","C","G","T": 1 1 1 1 1 1 1 1 1 1 ...
```

5. Find mean of numeric data in dataset based on Species group. and plot Bar chart (use

ggplot) to interpret same

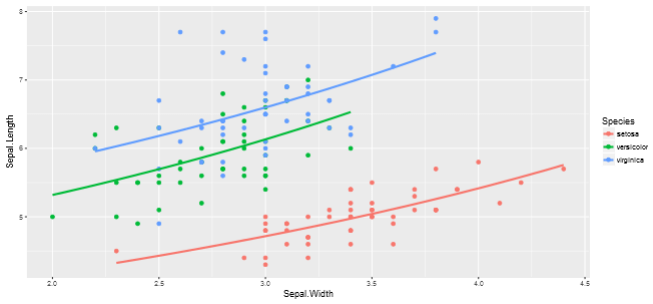
Sample output



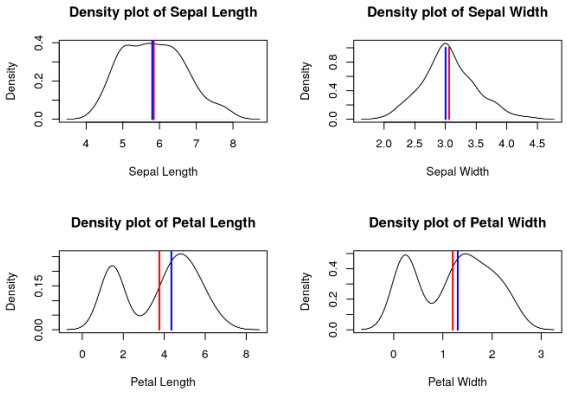
6.Draw a suitable plot which summaries statistical parameter of Sepal.Width based on

Species group

ANS:



7. Draw a suitable plot to find the skewness of the data for Sepal.Width and print the comment about skewness.



8.Draw ggplot2 scatterplot showing the variables Sepal.Length and Petal.Length grouped by the three-level factor “Species”.