Introduction to R and RStudio: Answer sheet

Please respond using short answer and code based answers.

###### Q1) What is the contig name and position of the last row in samtools.depth?

Chr4\_group5 in position 57,299

Code: > tail(samtools.depth)

###### Q2) What is the length of samtools.depth?

3

Code: “length(samtools.depth)”

###### Q3) What class is samtools.depth overall?

Data.frame

Code: class(samtools.depth)

###### Q4) What class is each column of samtools.depth?

Column 1: Character, Code: class(samtools.depth$V1)

Column 2: Integer, Code: class(samtools.depth[[2]])

Column 3: Integer, Code: class(samtools.depth[,"V3"])

###### Q5) What about the answer to the previous question makes it obvious that this object is a data.frame and not a matrix despite both objects being 2D tables?

It must be a data.frame because samtools.depth object has characters and integers, and matrices can only have one type of data.

Q6) Write code based on the example code above to validate that each column in the data.frame is the same length by combining an extraction method and the length command.

Each column has length of 250,000

Code:

> col1 <- samtools.depth$V1, then length(x = col1)

> col2 <- samtools.depth$V2, then length(x = col2)

> col3 <- samtools.depth$V3, then length(x = col3)

###### Q7) If you coerce (convert) row13 to become a vector using the unlist function, what is the class of the resulting vector?

Character

Code:

> row13 <- samtools.depth[13,]

> unlist(row13)

> x <- unlist(row13)

> class(x)

###### Q8) What property of vectors and row13’s contents prevent it from being an integer vector after using unlist?

Vectors can only have one type of data. However, row13 has characters and integers. So, it treats the integers in column 2 and 3 as characters and not as integers separate from the column 1 characters.

###### Q9) How many unique contigs are in this file?

5

Code: unique(samtools.depth$V1)

###### Q10) Why did “V1” get the list of contigs in this line of code?

The contigs are only under column 1. So, if we want to know each unique contig, we have to search for it only under column 1 (which is V1).

###### Q11) Write code to subset the data.frame to a single contig that is not chr4\_group5. Hint: I suggest that you use head and tail to check the result; it should be a data.frame still! Just shorter.

Wrote code to subset chr4\_group1 contig.

Code:

> chr4\_group1\_logic <- samtools.depth$V1 == "chr4\_group1"

> chr4\_group1 <- samtools.depth[chr4\_group1\_logic, ]

> head(x = chr4\_group1)

> tail(chr4\_group1)

> unique(chr4\_group5$V1)

###### Q12) Fill out statistics table

|  | Mean (V3) | Standard Deviation (V3) |
| --- | --- | --- |
| Whole data file | 10.801 | 6.153 |
| Chr4\_group5 | 9.384 | 4.758 |
| Your favorite contig (group1) | 9.997 | 4.827 |

Table 1. Fill in the table below with the calculation results for the indicated subset.

###### Whole data file: Used the “col3” code from question 6 which contained the values in column 3. Then used mean and sd code of col3.

###### Code: > mean(col3) and > sd(col3)

Chr4\_group5: First created a subset of the values in column 3 that were identified as the contig chr4\_group5. Used head and tail functions to make sure the subset was isolated correctly. Then used mean and sd codes.

Code:

> group5col3 <- samtools.depth$V3[samtools.depth$V1 == "chr4\_group5"]

> head(group5col3)

> tail(group5col3)

> mean(group5col3)

> sd(group5col3)

Chr4\_group1: Like chr4\_group5, first created a subset of column 3 values that were the chr4\_group1 contig. Used head and tail functions again. Then used mean and sd codes.

Code:

> group1col3 <- samtools.depth$V3[samtools.depth$V1 == "chr4\_group1"]

> head(group1col3)

> tail(group1col3)

> mean(group1col3)

> sd(group1col3)

###### Q13) Add a picture of the output plot (See “Saving Visualizations” below)

Code:

> xlim=c(-5,max(chr4\_group5$V3))

> hist(x = group5col3, breaks =

150)

> abline(v = mean(group5col3), col

= "red")

###### Q14) Add a picture of the random data plot.

Code:

> random.data <- rnorm(n = 1000000, mean = mean(group5col3), sd = sd(group5col3))

> hist(x = random.data, breaks = 150)

> abline(v = mean(random.data), col = "red")

###### Q15) Compare and contrast the shape of the plots. Describe how the real data deviated from the distribution of truly normally distributed data. This discussion should demonstrate a comprehension of what the values being visualized represent.

The real data is discrete, while the normally distributed data is continuous. This is evident by looking at the graphs and seeing if there are gaps between each bar. The histogram of real data clearly has gaps between the values because it only contains integers, whereas the histogram of normally distributed data has no gaps and is continuous because there are integers and rational numbers between each consecutive integer. In a biological sense, the real data can only contain integers because the values we are looking at are the number of base pairs that overlap at a certain position along aligned sequencing reads. Since we are counting base pairs, values must be integers (discrete).

The real data is not symmetrical; it is right skewed. In order for the real data to be symmetrical, the sequencing reads need to be aligned with equal spacing between each subsequent read. However, DNA is not aligned in such fashion. A right skewed plot means that it was more common to have base pair overlaps that were less than the mean amount than base pair overlaps greater than the mean amount.

Also, using the mean and median functions, the random data plot has near identical mean and median, both at 9.39, while the real data had a mean of 9.38 and median of 9.00.

The read depth values will always fail to meet the requirement of being continuous. Since the values can only be integers, the plot will always be discrete. At low read depths, it will also tend to fail the requirement of being symmetrical. It will tend to be right skewed because the readings will tend to be less than the mean.

###### EC (10%)

* Repeat the analysis above for group 2.
* Add the two histograms.
* Add a QQ plot by modifying the guide found at <http://www.sthda.com/english/wiki/qq-plots-quantile-quantile-plots-r-base-graphs>
* Compare and contrast the randomly generated normal data to the actual data.



Code:

> xlim=c(-5, max(chr4\_group2$V3))

> group2col3 <- samtools.depth$V3[samtools.depth$V1 == "chr4\_group2"]

> hist(x = group2col3, breaks = 150)

> abline(v = mean(group2col3), col = "red")



Code:

> rnorm(n = 1000000, mean = mean(group2col3), sd = sd(group2col3))

> random\_data <- rnorm(n = 1000000, mean = mean(group2col3), sd = sd(group2col3))

> hist(x = random\_data, breaks = 120)

> abline(v = mean(random\_data), col = "red")



Code:

> qqnorm(y = group2col3)

> qqline(y = group2col3, col = "red", lwd = 2)

Similar to group 5, group 2 also displayed data that was discrete whereas the normally distributed histogram was continuous. Again, the reason the real data is discrete is because the read depth values can only be integers as base pair overlaps only occur in integers.

The real data for group 2 was also not symmetrical. It was also right skewed.

For the real data, the mean was 14.33, and the median was 12.00. For the normally distributed data, the mean was 14.34, and the median was 14.34. Since the real data does not have equal mean and median values, it deviates from normally distributed values.

The qqplot also shows how the real data is not normally distributed because it clearly deviates from the qqline.