Interactive Review of Basic Statistics Using R

ACTIVITY 24

As has been discussed briefly, R is statistical software that can provide statistical analysis, whether in basic descriptive statistic computations or in advanced methods involving both non-parametric and parametric analytical methods. This activity revisits basic statistical analysis using R. It also re-introduces normal distribution and components of analysis regarding the basic assumptions of Gaussian distributions.

Purpose

The purpose of this activity is to introduce the basic statistical analysis package of R

Learning Objective

Review descriptive statistics, Gaussian distributions, and the coding methods and packages for R.

Required Resources

- R, R Studio
- Chapter 4, Dalgaard
- R Packages: RODBC, moments
- ODBC connection
- R script from class website

Time Allocated

40 minutes in class

Tasks

A. Data setup

The first order of business is to obtain the weigh-in-motion data from the class database for Type 11 Trucks (5-axle) recorded on the 8th and 9th of August, 2009.

```
qry <- “SELECT * FROM wim.wimdata WHERE timestamp >= ‘08-08-2009’ AND
timestamp < ‘08-09-2009’ AND type=’11’”

wim <- sqlQuery(channel, qry)
```

As you may noticed, the script provided has variable of *datatoexplore*. This generic term is defined as the variable given to you at the start of this activity. The script reads”

```
datatoexplore <- wim$spc2

label <- “Space 2”
```

Where `wim$spc2` is the spacing between the second and third axle. To facilitate exploration of the assigned variable, please define *datatoexplore* as your assigned variable.

Now we can start our exploration of the basic statistical functions that R provides. Let’s plot the data.

```
plot(datatoexplore)
```

1. What would be your best guess of what an observation would be for your assigned variable (just by looking at the plot)?

B. Measures of Central Tendency

Unsurprisingly, R has the ability to generate histograms which can provide quick insight into the shape of the data. The function `hist()` analyzes the data and generates a histogram to provide a measure of central tendency. The primary formal argument within `hist()` is the argument `breaks=`, which determines the breakpoints between the histogram cells. We will explore more on the histogram in subsequent activities.

R can also calculate the `mean()` and `median()`, but there is not a function to determine the mode, or the value that occurs the most often in the data. However, the mode is relatively easy to determine with a few lines of code in R.

```
table(datatoexplore)
max(table(datatoexplore))
```

The `table()` function counts the number of occurrences for values within the dataset. The `max()` function returns the maximum number of occurrences counted by the `table()` function. From this, the mode can be extrapolated by visual inspection of the table in R or determined by plotting a histogram using the `plot()` function.

```
plot(table(datatoexplore), type=”h”)
```

The `stripchart()` function can be used for small sample sizes. In the case of large sample sizes the detail is lost, especially if the data is highly centered around a small range of values. To see two comparative examples run the following R code.

```
par(mfrow=c(1,2))
stripchart(wim$spc3, pch=21, method=”stack”)
stripchart(wim$spc3, pch=21, method=”jitter”)
```

Another quick method to visually inspect the central tendency of a data set is to use the `stem()` function which generates a stem and leaf plot within the console of R Studio, but does not print to the plots tab in the respective pane. As with other functions in this section, a large sample size and non-descriptive breakpoints does not provide the necessary detail that can be obtained from using a histogram or other method of distribution description.

C. Measures of Relative Standing

The idea behind descriptive statistics is to describe the distribution of the data in an effort to assist in describing the relativity between data points within the same set. Establishing the quantiles of a data set assists in determining the percentiles of data that fall within a certain range. An easy way of determining the range and quantiles of a data set is to request the information using the `summary()` function.

2. Generate the summary statistics for your assigned data set and annotate your code with the results.

There exist other ways to generate percentile information to be used to described keys points of distribution within a data set, such as `quantile()` and for the 50 percent percentile or Q2, `median()`. The `quantile()` function can calculate the 50th percentile as demonstrated;

```
quantile(datatoexplore,probs=(0.5))
```

As would be expected the formal argument, `probs=` can be calculated for any probability value between 0 and 1.

3. Calculate the 25, 50, and 75 percentiles using the `quantile()` function and compare Q2 with the `median()` function.

Another way of discussing the relationship of data is by plotting empirical cumulative distribution function, or `plot.ecdf()`. Similarly, the call can be submitted as the following code:

```
plot(ecdf(datatoexplore), col=”dodgerblue”, xlab=label, ylab=”Cum %”,
xlim=range(datatoexplore))
```

4. Plot an empirical cumulative distribution function overlaid by a normal cumulative distribution function.

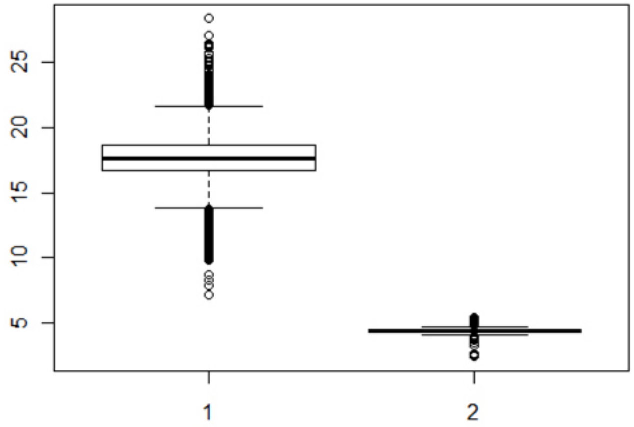

Figure 45 Side by Side box plots of axle space 1(left) and 2(right)

Boxplots are a common method to quickly provide the median, interquartile range, and any outliers in a data set. The function for a boxplot is `boxplot()`. The great thing about `boxplot()` is the acceptance of a formula to quickly separate categories of data. Let's see this in action (see Figure 45 for the graphic).

```
boxplot (wim$spc1, wim$spc2)
```

By calling *wim$spc1* and *wim$spc2* both box plots for the corresponding data will be plotted in the same graphic. Multiple values `boxplot(wim$spc1, wim$spc2, wim$spc3, wim$spc4,...)`can be coded to generate a comparison of many variables. A faster way to generate a series of box plots is described below (see Figure 45 for the graphical representation).

```
boxplot (wim$spc2 ~ wim$station)
```

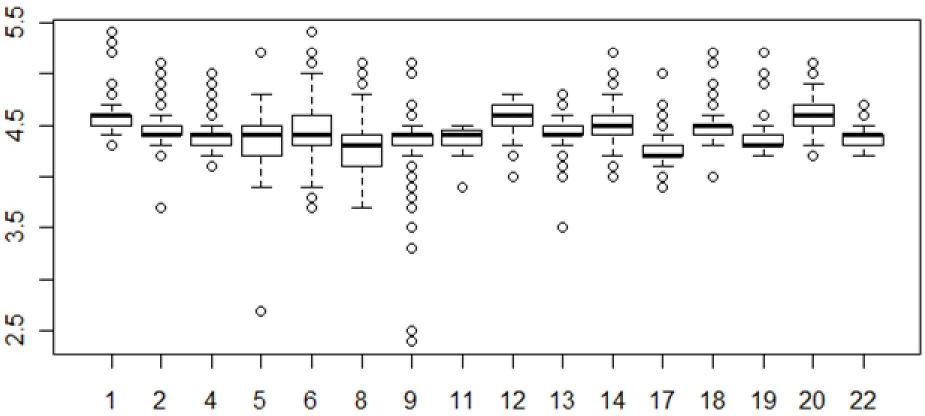

Figure 46 Box plots of axle space 2 by station

This method allows for multiple box plots to be generated based on a categorical variable like the stations for the weigh-in motion data as described by Figure 46.

D. Measures of Variability

Common variability measures are standard deviation or `sd()`, variance or `var()`, and the interquartile range which can be generated using the function `IRQ()` using `library(stats)` if not already active. Or since the interquartile range is the range between Q3 and Q1, the interquartile range can be calculated using the `quantile()` function.

5. Calculate the interquartile range by calculating the value from the `quantile()` function and check the value against `IQR()` function for your assigned variable.
6. Find the range(minimum value, maximum value) using the `range()` function and verify the result.
7. What is the coefficient of variation? Is there a function for coefficient of variation (CV)? If so, what package? Generate a quick function that can calculate the coefficient of variation.

E. Skewness and Kurtosis

As was discussed in the lecture portion of this activity, *skewness* is a numerical value demonstrating the asymmetrical behavior of a variable in a given data set. Upon visual inspection of a plot of counts like a histogram, skewness measures if predominance exists to the left or right of an assumed asymmetrical center.

Kurtosis is the measure of the peaked nature of the data, or the measure of the predominance of data points located near the mean.

R has the capability of calculating the both measures of shape with the addition of the moments package.

8. Using the `skewness()` and `kurtosis()` functions determine the skewness and kurtosis of your assigned data and briefly discuss the results.

Deliverable

Submit a PDF with your output, interpretations, and answers to the questions. Include your R code as an appendix in the PDF document. Submit in the course dropbox.

Assessment

Activity 24 Grading Rubric

	Excellent (10)	Good (8)	Poor (6)	NONE
Script	Organized, complete, accurate and executes.	Missing minor parts, but executes and is otherwise organized and accurate.	Missing significant portions of the activity, unorganized, inaccurate, but executes.	Code does not execute
Annotation	Annotations are complete and describe what the code is accomplishing.	Some annotations are incomplete or do not describe what the code is accomplishing.	No annotations were provided.	Code does not execute
Discussion/ Commentary	Insightful discussion or commentary relating to the question at hand demonstrating student understanding of the task.	Discussion or commentary was incomplete.	Minimal to no discussion or commentary.	Code does not execute