Nonparametric Tests

Taylor Shrode

4/26/2020

MSDS 660

A nonparametric test, which is often called a distribution free test, is based on fewer assumptions than parametric tests (Sullivan, n.d.). Parametric tests involve assumptions about the underlying distribution and involves making an assumption about the population's parameters (mean, median, standard deviation, etc.) (Sullivan, n.d.). While a nonparametric test can be used where distributions and assumptions are not required, they are generally less powerful than parametric tests. In other words, a parametric test is more likely to find a significant effect (Sullivan, n.d.). Nonparametric tests are generally used for nominal (category) or ordinal (order) scales, when one or more assumptions of a parametric test have been violated, sample size is small, data has outliers that cannot be removed, or you want to test the median rather than the mean (Stephanie, 2014).

For this assignment, we will be using the 1-sample Sign Test, which is the nonparametric alternative to the 1-sample t-test, and the 2-sample Wilcoxon Signed Rank Test, which is the nonparametric alternative to the 2-sample t-test. First, we will be using the 1-sample Sign Test. This nonparametric test allows us to compute a significance test of a hypothesized median (One sample sign test, 2017). The median is used instead of the mean because the median is a better measure for the central point of non-normal data (One sample sign test, 2017).

To begin, we need to upload our data into R. The data we will be analyzing includes the ratings of a new chain restaurant, Souperb. Fifteen people were asked to rate the food on a 5-point scale, where 1 is terrible and 5 is excellent. After loading our data into R, we can compute the median of the dataset.

```
> ratings <- c(5,3,2,1,4,3,5,1,5,2,3,4,2,1,3)
> particpants <- c(1:15)
> souperb_ratings <- data.frame(particpants, ratings)
> View(souperb_ratings)
> median(souperb_ratings$ratings)
[1] 3
> |
```

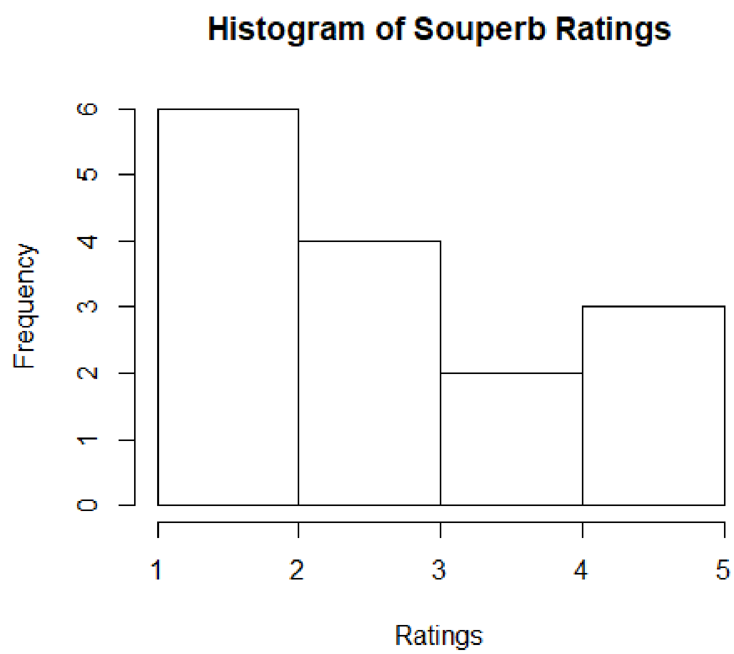
We will use the Sign Test to test if the median rating is at least 3, given a 95% confidence interval. The output below shows that our data has been loaded successfully.

	particpants	ratings
1	1	5
2	2	3
3	3	2
4	4	1
5	5	4
6	6	3
7	7	5
8	8	1
9	9	5
10	10	2
11	11	3
12	12	4
13	13	2
14	14	1
15	15	3

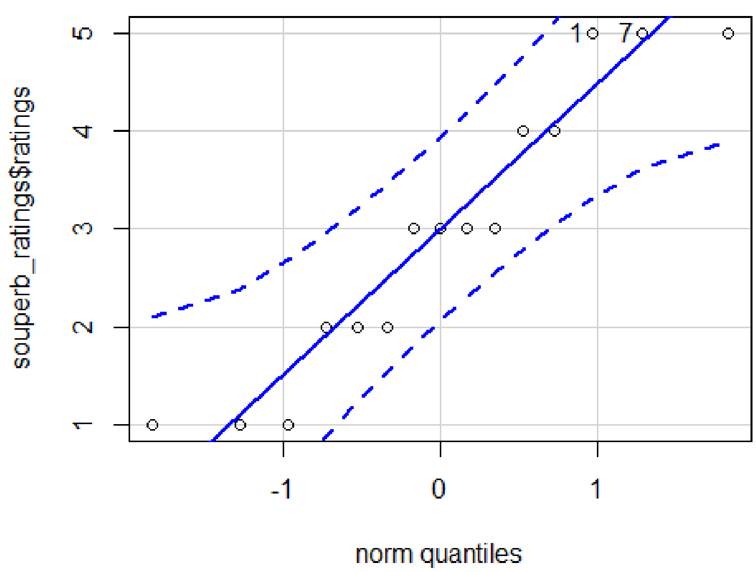
Recall that nonparametric tests are used when data is non-normally distributed. Using the command below,

```
> hist(souperb_ratings$ratings, xlab = "Ratings", main = "Histogram of Souperb Ratings")
```

we can create a histogram to check the distribution of the data.



As seen above, our data is non-normally distributed. We can also create a QQplot using the **qqPlot()** function in the **car** package (Qq-plots, n.d.). From the QQplot below, we can see that there are outliers in our dataset. The outliers are in row 1 and 7 in our dataset.



To conduct the Sign Test for a population median, we can either use binomial probabilities (**binom.test()**) or use the **SIGN.test()** function in R (WorldClass FTE). First, we will use the

binom.test() function in R (One sample sign test, n.d.). Recall that we are testing whether the median rating is at least 3. Thus, our null and alternative hypotheses are defined as

$$H_0\text{: The median rating from the population sample is at least 3; } median \geq 3$$
$$H_A\text{: The median rating from the population sample is less than 3; } median < 3.$$

The **binom.test()** function takes in the following arguments (One sample sign test, n.d.):

1. x = B: Number of data points that are greater than the median (successes).
2. n = N: Sample size after excluding the samples that are exact ties with the median (trials).
3. p = 0.5: Hypothesized probability of success.
4. conf.level: Confidence level.
5. alternative: Indicates the alternative hypothesis; must be “two.sided”, “greater”, or “less”.

To find the values for **B** and **N**, we can use the **sum()** function.

```
> sort(ratings)
 [1] 1 1 1 2 2 2 3 3 3 3 4 4 5 5 5
> sum(ratings>3)
[1] 5
> sum(ratings !=3)
[1] 11
```

Our value for **B** is 5 and our value for **N** is 11. According to our hypotheses above, we set the **alternative** argument to “less”, which is a one-sided test. Thus,

```
> binom.test(5, n = 11, p = 0.5, alternative = 'less', conf.level = 0.95)

        Exact binomial test

data:  5 and 11
number of successes = 5, number of trials = 11, p-value = 0.5
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.7287501
sample estimates:
probability of success
             0.4545455
```

At the 0.05 significance level, we cannot reject our null hypothesis that the median is at least 3 because our p-value is 0.50. In other words, our p-value is greater than our level of significance and cannot reject the null hypothesis.

We can also use the **SIGN.test()** function found in the **BSDA** package (Mangiafico, 2016). Our null and alternative hypotheses are the same as above,

$$H_0: The\ median\ rating\ from\ the\ population\ sample\ is\ at\ least\ 3;\ median\ \geq 3$$
$$H_A: The\ median\ rating\ from\ the\ population\ sample\ is\ less\ than\ 3;\ median\ < 3.$$

The **SIGN.test()** function takes in the following arguments (Arnholt, n.d.):

1. x: data to be tested.
2. md: A single number representing the value of the population median which is specified by the null hypothesis.
3. alternative: Specification of the alternative hypothesis; “greater”, “less”, or “two.sided”.
4. conf.level: Confidence level.

Our one-sample data is the **ratings** column in the **souperb_ratings** dataset, thus we use **souperb_ratings$ratings** for our data and **md** is equal to 3. We are using a one-sided test, thus,

```
> SIGN.test(souperb_ratings$ratings, md = 3, conf.level = 0.95, alternative = "less")

        One-sample Sign-Test

data:  souperb_ratings$ratings
s = 5, p-value = 0.5
alternative hypothesis: true median is less than 3
95 percent confidence interval:
 -Inf    4
sample estimates:
median of x
          3

Achieved and Interpolated Confidence Intervals:

                  Conf.Level L.E.pt U.E.pt
Lower Achieved CI     0.9408   -Inf      4
Interpolated CI       0.9500   -Inf      4
Upper Achieved CI     0.9824   -Inf      4
```

At a 0.05 significance level, we cannot reject the null hypothesis because our p-value is 0.50 which is greater than our level of significance. This implies that there is no evidence that the population median is less than 3. Also, notice that our sign test computed that the median of our sample is 3. There are also three different confidence intervals with varying levels of precision given (Interpret all statistics and graphs for 1-Sample Sign, n.d.). The confidence levels are defined as (Interpret all statistics and graphs for 1-Sample Sign, n.d.):

1. Lower Achieved CI: Highest achievable confidence level that is less than the specified confidence level.
    a. In the output above, we see that with 94.08% confidence, the population median rating is between -Inf ($-\infty$) and 4.
2. Interpolated CI: Specified confidence level.

a. In our output above, with 95%, the population median rating is between -Inf ($-\infty$) and 4.

3. Upper Achieved CI: Highest achievable confidence level that is greater than the level specified.
   a. In our output above, with 98%, the population median rating is between -Inf ($-\infty$) and 4.

Therefore, according to our binomial test and our sign test, there is no evidence that the population sample median is less than 3.

Now, we will be using the 2-sample Wilcoxon Signed Rank Test. This test is used when there are two nominal variables and one measurement variable, and the data is non-normally distributed. We will be analyzing two different operating systems, M and W. These systems have been rated on a scale of 1 to 10, where the higher the number the better the rate. We will be using the 2-sample Wilcoxon Signed Rank test to determine if the two operating systems have the same distribution. Note, we are assuming that the data is paired.

```
> M <- c(9,8,5,3,6,10,4,2,8,7)
> W <- c(7,6,8,2,9,5,1,4,7,10)
> operating_ratings <- data.frame(operating_system = rep(c("M", "W"), each = 10), rating = c
(M, W))
> View(operating_ratings)
> 
```

Loading the data into R and the using the **View()** command, we can view our data.

	operating_system	rating
1	M	9
2	M	8
3	M	5
4	M	3
5	M	6
6	M	10
7	M	4
8	M	2
9	M	8
10	M	7
11	W	7
12	W	6
13	W	8
14	W	2
15	W	9
16	W	5
17	W	1
18	W	4
19	W	7
20	W	10

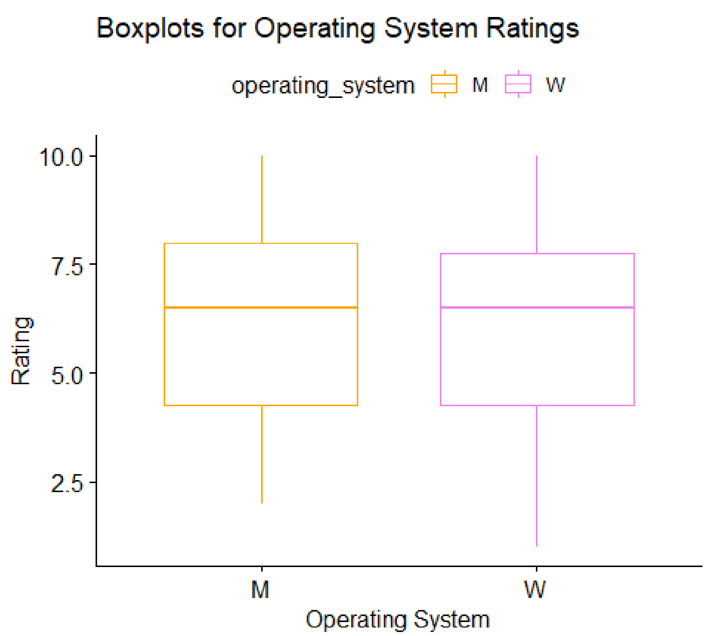
To compute summary statistics of the data above by groups, we can use the **dplyr** package (Unpaired two-samples wilcoxon test in r, n.d.). Below, we compute the median and interquartile range (IQR).

```
> library(dplyr)
> group_by(operating_ratings, operating_system) %>% summarise(count = n(), median = median(r
ating, na.rm = TRUE), IQR = IQR(rating, na.rm = TRUE))
# A tibble: 2 x 4
  operating_system count median   IQR
  <fct>            <int>  <dbl> <dbl>
1 M                   10    6.5  3.75
2 W                   10    6.5  3.5
> 
```

This output indicates that the medians for the operating systems are the same. Now, using the **ggpubr** package, we can create a boxplot of our data by groups (Unpaired two-samples wilcoxon test in r, n.d.).

```
> library(ggpubr)
Loading required package: ggplot2
Loading required package: magrittr
> ggboxplot(operating_ratings, x = "operating_system", y = "rating", color = "operating_sy
stem", palette = c("orange", "violet"), ylab = "Rating", xlab = "Operating System", main =
 "Boxplots for Operating System Ratings")
```

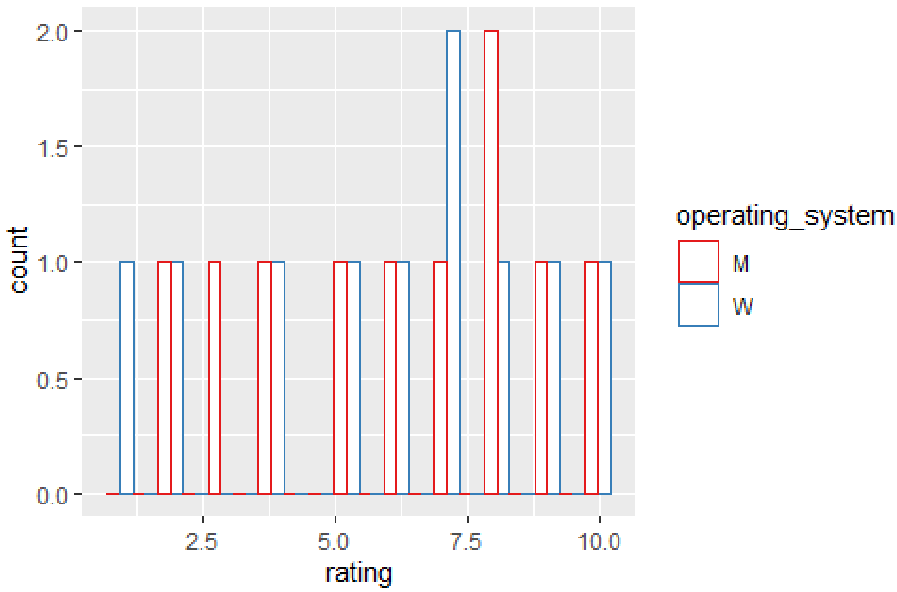
From the boxplot below, we can also see that the median of the operating systems is the same and the range of ratings for each operating system is close to the same.



Now. we can create a histogram of our data, using the **ggplot2** package, to view the distribution of our data (Ggplot2 histogram plot, n.d.).

```
> ggplot(operating_ratings, aes(x=rating, color=operating_system)) + geom_histogram(fill
="white", bins = 20, position = 'dodge') + scale_color_brewer(palette="Set1")
```

The histogram below indicates that our data is skewed and is non-normally distributed.



Now, we can conduct our Wilcoxon test. First, we need to define our null and alternative hypotheses. Recall we are testing whether the operating systems have the same distribution. Thus, our hypotheses are (WorldClass FTE):

$$H_0: The\ median\ difference\ between\ the\ operating\ system\ pairs\ is\ equal\ to\ 0$$
$$H_A: The\ median\ difference\ between\ the\ operating\ system\ pairs\ is\ not\ equal\ to\ 0.$$

This set of hypotheses indicate we will be using a two-sided test. We will use the **wilcox.exact()** function in the **exactRankTests** package. This function takes in the following arguments (Hothorn & Hormik, 2019):

1. x: Numeric vector of data values.
2. y: Numeric vector of data values (optional).
3. alternative: Alternative hypothesis; “two.sided”, “greater”, or “less”.
4. paired: A logical indicating whether you want a paired test.
5. conf.level: Confidence level.

Thus,

```
> library(exactRankTests)
 Package ‘exactRankTests’ is no longer under development.
 Please consider using package ‘coin’ instead.

> wilcox.exact(M, W, alternative = 'two.sided', paired  = TRUE, conf.level = 0.95)

        Exact Wilcoxon signed rank test

data:  M and W
V = 28.5, p-value = 0.9414
alternative hypothesis: true mu is not equal to 0
```

The p-value is greater than our level of significance ($0.9414 > 0.05$).Thus, we can not reject the null hypothesis. This indicates that the median difference between the ratings from operating system M and the operating system W is 0.

We performed two nonparametric tests: a 1-sample Sign Test and a 2-sample, paired Wilcoxon Sign Rank Test. After preparing various types of visuals to confirm that nonparametric tests were appropriate, we found that both the data sets being analyzed were non-normally distributed and contained outliers. Thus, making nonparametric tests appropriate. After conducting the nonparametric tests, we found that neither test indicated that we should reject the null hypotheses. Thus, there were no significant differences within our population samples.

Resources

Arnholt, A. (n.d.). *SIGN.test*. Retrieved April 26, 2020, from https://www.rdocumentation.org/packages/BSDA/versions/1.2.0/topics/SIGN.test

Ggplot2 histogram plot. (n.d.). Retrieved April 26, 2020, from http://www.sthda.com/english/wiki/ggplot2-histogram-plot-quick-start-guide-r-software-and-data-visualization

Hothorn, T., & Hormik, K. (2019). *Exact Distributions for Rank and Permutation Tests*. https://cran.r-project.org/web/packages/exactRankTests/exactRankTests.pdf

Interpret all statistics and graphs for 1-Sample Sign. (n.d.). Retrieved April 26, 2020, from https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/basic-statistics/inference/how-to/one-sample/1-sample-sign/interpret-the-results/all-statistics/

Mangiafico, S. S. (2016). *R handbook: Sign test for one-sample data*. https://rcompanion.org/handbook/F_03.html

One sample sign test . (2017, June 29). BPI Consulting. https://www.spcforexcel.com/help/statistical-tools/nonparametric/one-sign

One sample sign tests. (n.d.). Retrieved April 26, 2020, from https://cran.r-project.org/web/packages/distributions3/vignettes/one-sample-sign-tests.html

Qq-plots. (n.d.). Retrieved April 26, 2020, from http://www.sthda.com/english/wiki/qq-plots-quantile-quantile-plots-r-base-graphs

Stephanie. (2014, January 21). *Nonparametric data and tests(Distribution free tests)*. Statistics How To. https://www.statisticshowto.com/parametric-and-non-parametric-data/

Sullivan, L. (n.d.). *Nonparametric tests*. Retrieved April 26, 2020, from http://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/BS704_Nonparametric/BS704_Nonparametric_print.html

Unpaired two-samples wilcoxon test in r . (n.d.). Retrieved April 26, 2020, from http://www.sthda.com/english/wiki/unpaired-two-samples-wilcoxon-test-in-r