Traditional Nonparametric Tests

Introduction to Traditional Nonparametric Tests

Packages used in this chapter

The packages used in this chapter include:

- rcompanion

The following commands will install these packages if they are not already installed:

```
if(!require(rcompanion)){install.packages("rcompanion")}
```

Introduction

The traditional nonparametric tests presented in this book are primarily rank-based tests. Instead of using the numeric values of the dependent variable, the dependent variable is converted into relative ranks.

For example, imagine we have the heights of eight students in centimeters.

```
Height = c(110, 132, 137, 139, 140, 142, 142, 145)

names(Height) = letters[1:8]

Height

      a   b   c   d   e   f   g   h
    110 132 137 139 140 142 142 145


rank(Height)

      a   b   c   d   e   f   g   h
    1.0 2.0 3.0 4.0 5.0 6.5 6.5 8.0
```

a has the smallest height and so is ranked 1. *b* has the next smallest height and so is ranked 2. And so on. Note that *f* and *g* are tied for spots 6 and 7, and so share a rank of 6.5.

Also note that the value of *a* is quite a bit smaller than the others, but its rank is simply 1. Information about the absolute height values is lost, and only the relative ranking is retained in the ranks. That is, if the value of *a* were changed to 100 or 5 or –10, its rank would remain 1 in this data set.

The advantage of using these rank-based tests is that they don't make many assumptions about the distribution of the data. Instead, their conclusions are based on the relative ranks of values in the groups being tested.

Advantages of nonparametric tests

- Most of the traditional nonparametric tests presented in this section are relatively common, and your audience is relatively likely to be familiar with them.

- They are appropriate for interval/ratio or, often, ordinal dependent variables.

- Their nonparametric nature makes them appropriate for data that don't meet the assumptions of parametric analyses. These include data that are skewed, non-normal, contain outliers, or, possibly, are censored. (Censored data is data where there is an upper or lower limit to values. For example, if ages under 5 are reported as "under 5".)

Disadvantages of nonparametric tests

- These tests are typically named after their authors, with names like Mann–Whitney, Kruskal–Wallis, and Wilcoxon signed-rank. It may be difficult to remember these names, or to remember which test is used in which situation.

- Most of the traditional nonparametric tests presented here are limited by the types of experimental designs they can address. They are typically limited to a one-way comparison of independent groups (e.g. Kruskal–Wallis), or to unreplicated complete block design for paired samples (e.g. Friedman).

- There may be more flexible approaches that can cover more complex designs. The aligned ranks transformation is one nonparametric approach. Ordinal regression is appropriate when there is an ordinal dependent variable. Permutation tests may be applicable in some cases.

- Readers are likely to find a lot of contradictory information in different sources about the hypotheses and assumptions of these tests. In particular, authors will often treat the hypotheses of some tests as corresponding to tests of medians, and then list the assumptions of the test as corresponding to these hypotheses. However, if this is not explicitly explained, the result is that different sources list different assumptions that the underlying populations must meet in order for the test to be valid. This creates unnecessary confusion in the mind of students trying to correctly employ these tests.

Interpretation of nonparametric tests

In general, these tests determine if there is a *systematic* difference among groups. This may be due to a difference in location (e.g. median) or in the shape or spread of the distribution of the data. With the Mann–Whitney and Kruskal–Wallis tests, the difference among groups that is of interest is the probability of an observation from one group being larger than an observation from another group. If this probability is 0.50, this is termed “stochastic equality”, and when this probability is far from 0.50, it is sometimes called “stochastic dominance”.

Optional technical note: Without additional assumptions about the distribution of the data, the Mann–Whitney and Kruskal–Wallis tests do not test hypotheses about the group medians. Mangiafico (2015) and McDonald (2014) in the “References” section provide an example of a significant Kruskal–Wallis test where the groups have identical medians, but differ in their stochastic dominance.

Effect size statistics

Effect size statistics for traditional nonparametric tests include Cliff’s *delta* and Vargha and Delaney’s *A* for Mann–Whitney, and *epsilon*-squared and Freeman’s “coefficient of determination” (Freeman’s *theta*) (Freeman, 1965) for Kruskal–Wallis. Rank biserial correlation is appropriate for Mann–Whitney and the paired signed-rank test. Kendall’s *W* can be used for Friedman’s test.

A couple of accessible resources on effect sizes for these tests are Tomczak and Tomczak (2014) and King and Rosopa (2010).

Some effect size statistics included here determine the degree to which one group has data with higher ranks than other groups. They tend to vary from 0 (groups have data that are stochastically equal) to 1 (one group, the first, stochastically dominates) or –1 (the other, second, group stochastically dominates). They are related to the probability that a value from one group will be greater than a value from another group.

As rank-based measures, these effect size statistics do not indicate the difference in absolute values between groups. That is, if you were to replace the *5*’s in the second example below with *100*’s, the value of the effect size statistics would not change, because in either case the *5*’s or *100*’s are the highest-ranked numbers. For a practical interpretation of results, it is usually important to consider the absolute values of data such as with descriptive statistics.

```
library(rcompanion)

A = c(1,1,1, 2,2,2, 3,3,3, 4,4,4)
B = c(1,1,1, 2,2,2, 3,3,3, 4,4,4)

cliffDelta(x=A, y=B)

   Cliff.delta
             0

      ### This corresponds to a VDA of 0.5,
      ###  the probability of an observation in B being larger than
      ###  an observation in A.


A = c(1,1,1, 2,2,2, 3,3,3, 4,4,4)
B = c(2,2,2, 3,3,3, 4,4,4, 5,5,5)

cliffDelta(x=A, y=B)
```

```
   Cliff.delta
        -0.438

      ### Note that a negative Cliff’s delta suggests that the values in
      ###  B tend to be larger than those in A.

      ### This corresponds to a VDA of 0.281,
      ###  the probability of an observation in A being larger than
      ###  an observation in B.


A = c(1,1,1, 2,2,2, 3,3,3, 4,4,4)
B = c(3,3,3, 4,4,4, 5,5,5, 6,6,6)

cliffDelta(x=A, y=B)

   Cliff.delta
         -0.75

      ### This corresponds to a VDA of 0.125,
      ###  the probability of an observation in A being larger than
      ###  an observation in B.


A = c(1,1,1, 2,2,2, 3,3,3, 4,4,4)
B = c(5,5,5, 6,6,6, 7,7,7, 8,8,8)

cliffDelta(x=A, y=B)

   Cliff.delta
            -1

      ### This corresponds to a VDA of 0,
      ###  the probability of an observation in A being larger than
      ###  an observation in B.
```

Optional: Appropriate use of traditional nonparametric tests

Using traditional nonparametric tests with ordinal data

Some authors caution against using traditional nonparametric tests with ordinal dependent variables since many of them were developed for use with continuous (interval/ratio) data. Some authors have further concerns about situations where are likely to be many ties in ranks, such as Likert data.

Other authors argue that, since these tests rank-transform data before analysis and have adjustments for tied ranks, that they are appropriate for ordinal data.

Simulations comparing traditionally nonparametric tests to ordinal regression are presented in the “Optional: Simulated comparisons of traditional nonparametric tests and ordinal regression” in the *Introduction to Likert Data* chapter. And in Mangiafico (2019).

Using traditional nonparametric tests with interval/ratio data

These nonparametric tests are commonly used for interval/ratio data when the data fail to meet the assumptions of parametric analysis.

Some authors discourage using common nonparametric tests for interval/ratio data in some circumstances.

- One issue is the interpretation of the results mentioned above. That is, often results are incorrectly interpreted as a difference in medians when they are really describing a stochastic difference in distributions.

- Another problem is the lack of flexibility in designs that these test can handle.

- Finally, these tests may lack power relative to their parametric equivalents.

Given these considerations and the fact that that parametric statistics are often relatively robust to minor deviations in their assumptions, some authors argue that it is often better to stick with parametric analyses for interval/ratio data if it's possible to make them work. Often, with a parametric approach, a *generalized* linear model would be appropriate where *general* linear models aren't appropriate.

References

Freeman, L.C. 1965. *Elementary Applied Statitics: For Students in Behavioral Science*. John Wiley & Sons. New York.

King, B.M., P.J. Rosopa, and E.W. Minium. 2018. Some (Almost) Assumption-Free Tests. In *Statistical Reasoning in the Behavioral Sciences*, 7th ed. Wiley.

"Kruskal–Wallis Test" in Mangiafico, S.S. 2015. *An R Companion for the Handbook of Biological Statistics*, version 1.09. rcompanion.org/rcompanion/d_06.html.

"Kruskal–Wallis Test" in McDonald, J.H. 2014. *Handbook of Biological Statistics*. www.biostathandbook.com/kruskalwallis.html.

Mangiafico, S.S. 2019. How Should We Analyze Likert Item Data? *Journal of the National Association of County Agricultural Agents* 12(2). www.nacaa.com/journal/index.php?jid=1001.

Tomczak, M. and Tomczak, E. 2014. The need to report effect size estimates revisited. An overview of some recommended measures of effect size. Trends in Sports Sciences 1(21):1–25. www.tss.awf.poznan.pl/files/3_Trends_Vol21_2014__no1_20.pdf.

One-sample Wilcoxon Signed-rank Test

One-sample tests are useful to compare a set of values to a given default value. For example, one might ask if a set of five-point Likert scores are significantly different from a "default" or "neutral" score of 3. Another use might be to compare a current set of values to a previously published value.

The one-sample Wilcoxon test is a rank-based test that begins with calculating the difference between the observed values and the default value. Because of this subtraction operation in the calculations, the data are assumed to be interval. That is, with Likert item data with this test, the data are assumed to be numeric. For purely ordinal data, the one-sample sign test could be used instead.

The null hypothesis for the test is that the data are symmetric about the default value (except that the test is done on ranks after the distances of observations from default value are determined).

A significant result suggests either that the (ranked) data are symmetric about another value or are sufficiently skewed in one direction. In either case, this suggests that the location of the data is different from the chosen default value.

Without further assumptions about the distribution of the data, the test is not a test of the median.

Appropriate data

- One-sample data
- Data are interval or ratio

Hypotheses

- Null hypothesis (simplified): The population from which the data are sampled is symmetric about the default value.
- Alternative hypothesis (simplified, two-sided): The population from which the data are sampled is not symmetric about the default value.

Interpretation

Reporting significant results as e.g. "Likert scores were significantly different from a neutral value of 3" is acceptable.

Notes on name of test

The names used for the one-sample Wilcoxon signed-rank test and similar tests can be confusing. "Sign test" may be used, although properly the sign test is a different test. Both "signed-rank test" and "sign test" are sometimes used to refer to either one-sample or two-sample tests.

The best advice is to use a name specific to the test being used.

Other notes and alternative tests

Some authors recommend this test only in cases where the data are symmetric. It is my understanding that this requirement is only for the test to be considered a test of the median.

For ordinal data or for a test specifically about the median, the sign test can be used.

Packages used in this chapter

The packages used in this chapter include:

- psych
- FSA
- rcompanion
- coin
- exactRankTests

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(coin)){install.packages("coin")}
if(!require(exactRankTests)){install.packages("exactRankTests")}
```

One-sample Wilcoxon signed-rank test example

This example will re-visit the Maggie Simpson data from the *Descriptive Statistics for Likert Data* chapter.

The example answers the question, “Are Maggie’s scores significantly different from a ‘neutral’ score of 3?”

The test will be conducted with the *wilcox.test* function, which produces a *p*-value for the hypothesis, as well a pseudo-median and confidence interval.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

  Speaker          Rater  Likert
 'Maggie Simpson'   1        3
 'Maggie Simpson'   2        4
 'Maggie Simpson'   3        5
 'Maggie Simpson'   4        4
 'Maggie Simpson'   5        4
 'Maggie Simpson'   6        4
 'Maggie Simpson'   7        4
 'Maggie Simpson'   8        3
 'Maggie Simpson'   9        2
 'Maggie Simpson'  10        5
")


### Create a new variable which is the likert scores as an ordered factor

Data$Likert.f = factor(Data$Likert,
                       ordered = TRUE)
```

```
###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

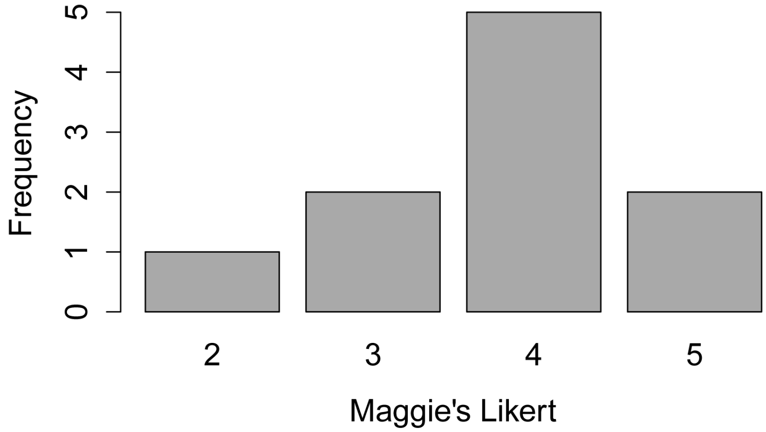
Summarize data treating Likert scores as factors

Note that the variable we want to count is *Likert.f*, which is a factor variable. Counts for *Likert.f* are cross tabulated over values of *Speaker*. The *prop.table* function translates a table into proportions. The *margin=1* option indicates that the proportions are calculated for each row.

```
xtabs( ~ Speaker + Likert.f,
       data = Data)

                  Likert.f
   Speaker         2 3 4 5
     Maggie Simpson 1 2 5 2


XT = xtabs( ~ Speaker + Likert.f,
           data = Data)

prop.table(XT,
           margin = 1)

                  Likert.f
   Speaker           2   3   4   5
     Maggie Simpson 0.1 0.2 0.5 0.2
```

Bar plot

```
XT = xtabs(~ Likert.f,
          data=Data)

barplot(XT,
        col="dark gray",
        xlab="Maggie's Likert",
        ylab="Frequency")
```



Summarize data treating Likert scores as numeric

```
library(FSA)

Summarize(Likert ~ Speaker,
          data=Data,
          digits=3)

           Speaker  n mean    sd min   Q1 median Q3 max percZero
1 Maggie Simpson 10  3.8 0.919   2 3.25      4  4   5        0
```

One-sample Wilcoxon signed-rank test

wilcox.test function

In the *wilcox.test* function, the *mu* option indicates the value of the default value to compare to. In this example *Data$Likert* is the one-sample set of values on which to conduct the test. For the meaning of other options, see *?wilcox.test*.

```
wilcox.test(Data$Likert,
            mu=3,
            conf.int=TRUE,
            conf.level=0.95)

   Wilcoxon signed rank test with continuity correction

   V = 32.5, p-value = 0.04007

   alternative hypothesis: true location is not equal to 3

      ### Note p-value in the output above

      ### You will get the "cannot compute exact p-value with ties" error
      ###   You can ignore this, or use the exact=FALSE option.

   95 percent confidence interval:
```

```
 3.000044 4.500083

sample estimates:

(pseudo)median
      4.000032

   ### Note that the output will also produce a pseudo-median value
   ###   and a confidence interval if the conf.int=TRUE option is used.
```

wilcox.exact function

```
wilcox.exact(Data$Likert,
             mu=3,
             exact=TRUE,
             conf.int=TRUE,
             conf.level=0.95)
```

```
Exact Wilcoxon signed rank test

V = 32.5, p-value = 0.05469

alternative hypothesis: true mu is not equal to 3

95 percent confidence interval:
 2 5

sample estimates:
(pseudo)median
          3.75
```

Effect size

I am not aware of any established effect size statistic for the one-sample Wilcoxon signed-rank test. However, a version of the rank biserial correlation coefficient (*rc*) can be used. This is my recommendation. It is included for the matched pairs case in King, Rosopa, and Minimum (2000).

As an alternative, using a statistic analogous to the *r* used in the Mann–Whitney test may make sense.

The following interpretation is based on my personal intuition. It is not intended to be universal.

	small	medium	Large
rc	0.10 – < 0.30	0.30 – < 0.50	≥ 0.50
r	0.10 – < 0.40	0.40 – < 0.60	≥ 0.60

Rank biserial correlation coefficient

```
library(rcompanion)
```

```
wilcoxonOneSampleRC(Data$Likert, mu=3)

      rc
   0.806


library(rcompanion)

wilcoxonOneSampleRC(Data$Likert, mu=3, ci=TRUE)

         rc lower.ci upper.ci
      0.806    0.333        1
```

r

```
library(rcompanion)

wilcoxonOneSampleR(Data$Likert, mu=3)

       r
   0.674


wilcoxonOneSampleR(Data$Likert, mu=3, ci=TRUE)

          r lower.ci upper.ci
      0.674      0.2    0.914
```

References

King, B.M., P.J. Rosopa, E.W. and Minium. 2000. *Statistical Reasoning in the Behavioral Sciences*, 6th. Wiley.

Exercises I

1. Considering Maggie Simpson's data,

a. What was her median score?

b. What were the first and third quartiles for her scores?

c. According to the one-sample Wilcoxon signed-rank test, are her scores significantly different from a neutral score of 3?

d. Is the confidence interval output from the test useful in answering the previous question?

e. Overall, how would you summarize her results? Be sure to address the practical implication of her scores compared with a neutral score of 3.

f. Do these results reflect what you would expect from looking at the bar plot?

2. Brian Griffin wants to assess the education level of students in his course on creative writing for adults. He wants to know the median education level of his class, and if the education level of his class is different from the typical Bachelor's level.

Brian used the following table to code his data.

Code	Abbreviation	Level
1	< HS	Less than high school
2	HS	High school
3	BA	Bachelor's
4	MA	Master's
5	PhD	Doctorate

The following are his course data.

```
Instructor       Student  Education
'Brian Griffin'  a        3
'Brian Griffin'  b        2
'Brian Griffin'  c        3
'Brian Griffin'  d        3
'Brian Griffin'  e        3
'Brian Griffin'  f        3
'Brian Griffin'  g        4
'Brian Griffin'  h        5
'Brian Griffin'  i        3
'Brian Griffin'  j        4
'Brian Griffin'  k        3
'Brian Griffin'  l        2
```

For each of the following, answer the question, and ***show the output from the analyses you used to answer the question***.

a. What was the median education level? (Be sure to report the education level, not just the numeric code!)

b. What were the first and third quartiles for education level?

b. According to the one-sample Wilcoxon signed-rank test, are the education levels significantly different from a typical level of Bachelor's?

e. Is the confidence interval output from the test useful in answering the previous question?

f. Overall, how would you summarize the results? Be sure to address the practical implications.

g. Plot Brian's data in a way that helps you visualize the data.

h. Do the results reflect what you would expect from looking at the plot?

Sign Test for One-sample Data

The one-sample sign test compares the number of observations greater than or less than the default value without accounting for the magnitude of the difference between each observation and the default value. The test is similar in purpose to the one-sample Wilcoxon signed-rank test, but looks specifically at the median value, and is not affected by the distribution of the data.

The test is conducted with functions in the *DescTools* package, the *nonpar* package, or the *BSDA* package. These functions produce a *p*-value for the hypothesis, as well as the median and confidence interval of the median for the data.

Appropriate data

- One-sample data
- Data are ordinal, interval, or ratio

Hypotheses

- Null hypothesis: The median of the population from which the sample was drawn is equal to the default value.
- Alternative hypothesis (two-sided): The median of the population from which the sample was drawn is not equal to the default value.

Interpretation

Reporting significant results as e.g. "Likert scores were significantly different from a default value of 3" is acceptable. As is e.g. "Median Likert scores were significantly different from a default value of 3"

Packages used in this chapter

The packages used in this chapter include:

- BSDA
- DescTools
- rcompanion
- nonpar

The following commands will install these packages if they are not already installed:

```
if(!require(BSDA)){install.packages("BSDA")}
if(!require(DescTools)){install.packages("DescTools")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(nonpar)){install.packages("nonpar")}
```

One-sample sign test example

For appropriate plots and summary statistics, see the *One-sample Wilcoxon Signed-rank Test* chapter.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

  Speaker           Rater     Likert
 'Maggie Simpson'   1         3
 'Maggie Simpson'   2         4
 'Maggie Simpson'   3         5
 'Maggie Simpson'   4         4
 'Maggie Simpson'   5         4
 'Maggie Simpson'   6         4
 'Maggie Simpson'   7         4
 'Maggie Simpson'   8         3
 'Maggie Simpson'   9         2
 'Maggie Simpson'  10         5
")

###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

Sign test with the DescTools package

Note that *Data$Likert* is the one-sample data, and *mu=3* indicates the default value to compare to.

```
library(DescTools)

SignTest(Data$Likert,
         mu = 3)

   One-sample Sign-Test

   S = 7, number of differences = 8, p-value = 0.07031

      ### Note the p-value in the output above

   alternative hypothesis: true median is not equal to 3

   97.9 percent confidence interval:
    3 5

   sample estimates:
   median of the differences
```

```
                          4

      ### Median value and confidence interval
```

Sign test with the nonpar package

Note that *Data$Likert* is the one-sample data, and *m=3* indicates the default value to compare to. At the time of writing, it appears that the *exact=FALSE* option actually produces the exact test.

```
library(nonpar)

signtest(Data$Likert, m=3, conf.level=0.95, exact=FALSE)

   Exact Sign Test

   The p-value is  0.07032

   The  95 % confidence interval is [ 2 ,  4 ].
```

Sign test with the BSDA package

Note that *Data$Likert* is the one-sample data, and *md=3* indicates the default value to compare to.

```
library(BSDA)

SIGN.test(Data$Likert,
          md = 3)

   One-sample Sign-Test

   s = 7, p-value = 0.07031

   alternative hypothesis: true median is not equal to 3

      ### Note the p-value in the output above

   95 percent confidence interval:
    3.000000 4.675556

   sample estimates:
   median of x
             4

      ### Median value and confidence interval
```

Effect size statistics

One way to assess the effect size after a one-sample sign test is to use a dominance statistic. This statistic simply looks at the proportion of observations greater than the default median value minus the proportion of observations less than the default median value. A value of 1 would indicate that all observations are greater than the default median, and a value of –1 would indicate that all observations are less than the default median. A value of 0 indicates that the number of observations greater than the default median are equal to the number that are less than the default median.

A VDA-like statistic can be calculated as *Dominance / 2 + 0.5*. This statistic varies from 0 to 1, with 0.5 being equivalent to a dominance value of 0.

Note that neither of these statistics take into account values tied to the default median value.

```
library(rcompanion)

oneSampleDominance(Data$Likert, mu=3)

     n Median mu Less Equal Greater Dominance VDA
   1 10      4  3  0.1   0.2     0.7       0.6 0.8


oneSampleDominance(Data$Likert, mu=3, ci=TRUE)

     n Median mu Less Equal Greater Dominance lower.ci upper.ci VDA lower.vda.ci upper.vda.ci
   1 10      4  3  0.1   0.2     0.7       0.6      0.2      0.9 0.8          0.6         0.95
```

Manual calculations

```
Likert = c(3, 4, 5, 4, 4, 4, 4, 3, 2, 5)

Greater = sum(Likert > 3)

NotMedian = sum(Likert != 3)

binom.test(Greater, NotMedian)

   Exact binomial test

   number of successes = 7, number of trials = 8, p-value = 0.07031


MU = 3

N = length(Likert)

GreaterProp = sum(Likert > MU) / N

GreaterProp

   0.7


LesserProp = sum(Likert < MU) / N

LesserProp

   0.1
```

```
EqualProp = sum(Likert == MU) / N

EqualProp

   0.2
```

Two-sample Mann–Whitney U Test

The two-sample Mann–Whitney U test is a rank-based test that compares values for two groups. A significant result suggests that the values for the two groups are different. It is equivalent to a two-sample Wilcoxon rank-sum test.

Without further assumptions about the distribution of the data, the Mann–Whitney test does not address hypotheses about the medians of the groups. Instead, the test addresses if it is likely that an observation in one group is greater than an observation in the other. This is sometimes stated as testing if one sample has stochastic dominance compared with the other.

The test assumes that the observations are independent. That is, it is not appropriate for paired observations or repeated measures data.

The test is performed with the *wilcox.test* function in the native *stats* package.

Appropriate effect size statistics include Vargha and Delaney's *A*, Cliff's *delta*, and the Glass rank biserial coefficient.

Appropriate data

- Two-sample data. That is, one-way data with two groups only
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with two levels. That is, two groups
- Observations between groups are independent. That is, not paired or repeated measures data
- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread. Otherwise, the test is typically a test of stochastic equality.

Hypotheses

- Null hypothesis: The two groups are sampled from populations with identical distributions. Typically, that the sampled populations exhibit stochastic equality.
- Alternative hypothesis (two-sided): The two groups are sampled from populations with different distributions. Typically, that one sampled population exhibits stochastic dominance.

Interpretation
Significant results can be reported as e.g. "Values for group A were significantly different from those for group B."

Other notes and alternative tests
The Mann–Whitney U test can be considered equivalent to the Kruskal–Wallis test with only two groups. Mood's median test compares the medians of two groups. Aligned ranks transformation anova (ART anova) provides nonparametric analysis for a variety of designs. For ordinal data, an alternative is to use cumulative link models, which are described later in this book.

Optional technical note on hypotheses for Mann–Whitney test
See the *Kruskal–Wallis Test* chapter for more information.

Packages used in this chapter

The packages used in this chapter include:

- psych
- FSA
- lattice
- rcompanion
- coin
- DescTools
- effsize
- exactRankTests

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(lattice)){install.packages("lattice")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(coin)){install.packages("coin")}
if(!require(DescTools)){install.packages("DescTools")}
if(!require(effsize)){install.packages("effsize")}
if(!require(exactRankTests)){install.packages("exactRankTests")}
```

Two-sample Mann–Whitney U test example

This example re-visits the Pooh and Piglet data from the *Descriptive Statistics with the likert Package* chapter.

It answers the question, "Are Pooh's scores significantly different from those of Piglet?"

The Mann–Whitney U test is conducted with the *wilcox.test* function in the native *stats* package, which produces a *p*-value for the hypothesis. First the data are summarized and examined using bar plots for each group.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Likert
 Pooh      3
 Pooh      5
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      5
 Pooh      5
 Piglet    2
 Piglet    4
 Piglet    2
 Piglet    2
 Piglet    1
 Piglet    2
 Piglet    3
 Piglet    2
 Piglet    2
 Piglet    3
")


### Create a new variable which is the Likert scores as an ordered factor

Data$Likert.f = factor(Data$Likert,
                       ordered = TRUE)


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

Summarize data treating Likert scores as factors

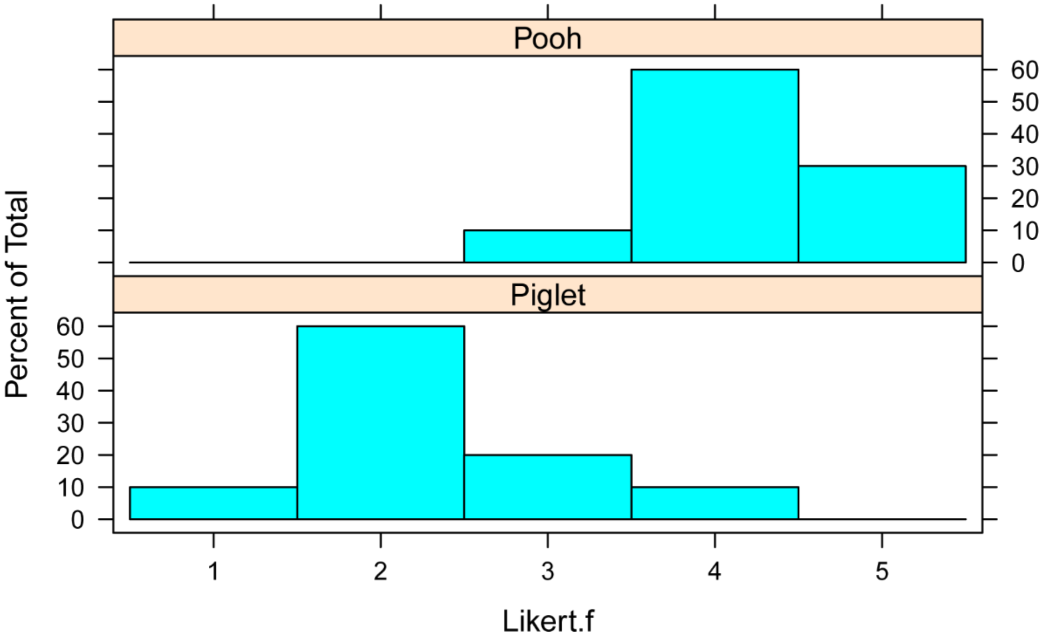
Note that the variable we want to count is *Likert.f*, which is a factor variable. Counts for *Likert.f* are cross tabulated over values of *Speaker*. The *prop.table* function translates a table into proportions. The *margin=1* option indicates that the proportions are calculated for each row.

```
xtabs( ~ Speaker + Likert.f,
       data = Data)

          Likert.f
```

```
   Speaker  1 2 3 4 5
     Piglet 1 6 2 1 0
     Pooh   0 0 1 6 3


XT = xtabs( ~ Speaker + Likert.f,
            data = Data)

prop.table(XT,
           margin = 1)

           Likert.f
   Speaker    1   2   3   4   5
     Piglet 0.1 0.6 0.2 0.1 0.0
     Pooh   0.0 0.0 0.1 0.6 0.3
```

Bar plots of data by group

```
library(lattice)

histogram(~ Likert.f | Speaker,
          data=Data,
          layout=c(1,2)      #  columns and rows of individual plots
          )
```



Summarize data treating Likert scores as numeric

```
library(FSA)

Summarize(Likert ~ Speaker,
          data=Data,
          digits=3)

     Speaker  n mean    sd min Q1 median   Q3 max percZero
   1  Piglet 10  2.3 0.823   1  2      2 2.75   4        0
   2    Pooh 10  4.2 0.632   3  4      4 4.75   5        0
```

Two-sample Mann–Whitney U test example

This example uses the formula notation indicating that *Likert* is the dependent variable and *Speaker* is the independent variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?wilcox.test*.

```
wilcox.test(Likert ~ Speaker,
            data=Data)

   Wilcoxon rank sum test with continuity correction

   W = 5, p-value = 0.0004713

   ### You may get a "cannot compute exact p-value with ties" error.
   ###    You can ignore this or use the exact=FALSE option.
```

As an alternative, the Mann–Whitney test can be conducted by exact test or Monte Carlo simulation with the *coin* package.

```
library(coin)

wilcox_test(Likert ~ Speaker, data=Data, distribution = "exact")

   Exact Wilcoxon-Mann-Whitney Test

   data:  Likert by Speaker (Piglet, Pooh)
   Z = -3.5358, p-value = 0.0002382


library(coin)

wilcox_test(Likert ~ Speaker, data=Data, distribution = "approximate")

   Approximative Wilcoxon-Mann-Whitney Test

   Z = -3.5358, p-value = 2e-04
```

Another approach is to use the exact test from the *exactRankTests* package.

```
library(exactRankTests)

wilcox.exact(Likert ~ Speaker, data=Data, exact=TRUE)

   Exact Wilcoxon rank sum test
```

```
W = 5, p-value = 0.0002382
```

Effect size

Statistics of effect size for the Mann–Whitney test report the degree to which one group has data with higher ranks than the other group. They are related to the probability that a value from one group will be greater than a value from the other group. Unlike *p*-values, they are not affected by sample size.

Vargha and Delaney's *A* is relatively easy to understand. It reports the probability that a value from one group will be greater than a value from the other group. A value of 0.50 indicates that the two groups are stochastically equal. A value of 1 indicates that the first group shows complete stochastic domination over the other group, and a value of 0 indicates the complete stochastic domination by the second group.

Cliff's *delta* is linearly related to Vargha and Delaney's *A*. It ranges from –1 to 1, with 0 indicating stochastic equality of the two groups. 1 indicates that one group shows complete stochastic dominance over the other group, and a value of –1 indicates the complete stochastic domination of the other group. Its absolute value will be numerically equal to Freeman's *theta*.

The Glass rank biserial coefficient (*rg*) is a recommended effect size statistic, and, as far as I can tell, is equivalent to Cliff's *delta*. It is included in King, Rosopa, and Minimum (2000).

A common effect size statistic for the Mann–Whitney test is *r*, which is the *z* value from the test divided by the square root of the total number of observations. This statistic has some drawbacks. Under usual circumstances, it will not range all the way from –1 to 1. It is also affected by sample size. These problems appear to get worse when there are unequal sample sizes between the groups.

Kendall's *tau*-b is sometimes used and varies from approximately –1 to 1.

Freeman's *theta* and *epsilon*-squared are usually used when there are more than two groups, with the Kruskal–Wallis test, but can also be employed in the case of two groups.

Interpretation of effect sizes necessarily varies by discipline and the expectations of the experiment, but for behavioral studies, the guidelines proposed by Cohen (1988) are sometimes followed. The following guidelines are based on the literature values and my personal intuition. They should not be considered universal.

Optional technical note: The interpretation values for *r* below are found commonly in published literature and on the internet. I suspect that this interpretation stems from the adoption of Cohen's interpretation of values for Pearson's *r*. This may not be justified, but it turns out that this interpretation for the *r* used here is relatively reasonable. The interpretation for *tau*-b, Freeman's *theta*, and *epsilon*-squared here are based on their values relative to those for *r*, based on simulated data (5-point Likert items, *n* per group between 4 and 25). Plots for some of these simulations are shown below.

Interpretations for Vargha and Delaney's *A* and Cliff's *delta* come from Vargha and Delaney (2000).

	small	medium	large
r	0.10 – < 0.30	0.30 – < 0.50	≥ 0.50
***tau*-b**	0.10 – < 0.30	0.30 – < 0.50	≥ 0.50
Cliff's *delta* or *rg*	0.11 – < 0.28	0.28 – < 0.43	≥ 0.43
Vargha and Delaney's *A*	0.56 – < 0.64	0.64 – < 0.71	≥ 0.71
	> 0.34 – 0.44	> 0.29 – 0.34	≤ 0.29
Freeman's *theta*	0.11 – < 0.34	0.34 – < 0.58	≥ 0.58
***epsilon*-squared**	0.01 – < 0.08	0.08 – < 0.26	≥ 0.26

Vargha and Delaney's *A*

```
library(effsize)

VD.A(d = Data$Likert,
     f = Data$Speaker)

   Vargha and Delaney A

   A estimate: 0.05 (large)


library(rcompanion)

vda(Likert ~ Speaker, data=Data)

    VDA
   0.05


library(rcompanion)

vda(Likert ~ Speaker, data=Data, ci=TRUE)

      VDA lower.ci upper.ci
   1 0.05        0    0.162

      ### Note: Bootstrapped confidence interval may vary.
```

Glass rank biserial correlation coefficient

```
library(rcompanion)

wilcoxonRG(x = Data$Likert,
           g = Data$Speaker )

     rg
   -0.9
```

```
library(rcompanion)

wilcoxonRG(x = Data$Likert,
           g = Data$Speaker,
           ci = TRUE)

       rg lower.ci upper.ci
   1 -0.9       -1   -0.697

      ### Note: Bootstrapped confidence interval may vary.
```

Cliff's *delta*

```
library(effsize)

cliff.delta(d = Data$Likert,
            f = Data$Speaker)

   Cliff's Delta

   delta estimate: -0.9 (large)

   95 percent confidence interval:
        lower      upper
   -0.9801533 -0.5669338
```

```
library(rcompanion)

cliffDelta(Likert ~ Speaker, data=Data)

   Cliff.delta
          -0.9
```

```
library(rcompanion)

cliffDelta(Likert ~ Speaker, data=Data, ci=TRUE)

     Cliff.delta lower.ci upper.ci
   1        -0.9       -1    -0.67

      ### Note: Bootstrapped confidence interval may vary.
```

r

```
library(rcompanion)
```

```
wilcoxonR(x = Data$Likert,
          g = Data$Speaker)

       r
   0.791


library(rcompanion)

wilcoxonR(x  = Data$Likert,
          g  = Data$Speaker,
          ci = TRUE)

         r lower.ci upper.ci
   1 0.791    0.602    0.897

      ### Note: Bootstrapped confidence interval may vary.
```

Agresti's Generalized Odds Ratio for Stochastic Dominance

```
library(rcompanion)

wilcoxonOR(Likert ~ Speaker, data=Data)

        OR
   1 0.011


wilcoxonOR(Likert ~ Speaker, data=Data, ci=TRUE)

        OR lower.ci upper.ci
   1 0.011        0    0.073

      ### Note: Bootstrapped confidence interval may vary.
```

Grissom and Kim's Probability of Superiority

```
library(rcompanion)

wilcoxonPS(Likert ~ Speaker, data=Data)

       PS
   1 0.01


wilcoxonPS(Likert ~ Speaker, data=Data, ci=TRUE)

   1 0.01       0     0.06

      ### Note: Bootstrapped confidence interval may vary.
```

tau-b

```
library(DescTools)

KendallTauB(x = Data$Likert,
            y = as.numeric(Data$Speaker))

   [1] 0.7397954


library(DescTools)

KendallTauB(x = Data$Likert,
            y = as.numeric(Data$Speaker),
            conf.level = 0.95)

       tau_b    lwr.ci    upr.ci
   0.7397954 0.6074611 0.8721298
```

Freeman’s *theta*

```
library(rcompanion)

freemanTheta(x = Data$Likert,
             g = Data$Speaker)

   Freeman.theta
             0.9


library(rcompanion)

freemanTheta(x  = Data$Likert,
             g  = Data$Speaker,
             ci = TRUE)

     Freeman.theta lower.ci upper.ci
   1           0.9    0.688        1


      ### Note: Bootstrapped confidence interval may vary.
```

epsilon-squared

```
library(rcompanion)

epsilonSquared(x = Data$Likert,
               g = Data$Speaker)

   epsilon.squared
             0.658
```

```
library(rcompanion)

epsilonSquared(x  = Data$Likert,
               g  = Data$Speaker,
               ci = TRUE)

     epsilon.squared lower.ci upper.ci
   1           0.658    0.383    0.842


      ### Note: Bootstrapped confidence interval may vary.
```

Optional: extracting the z value

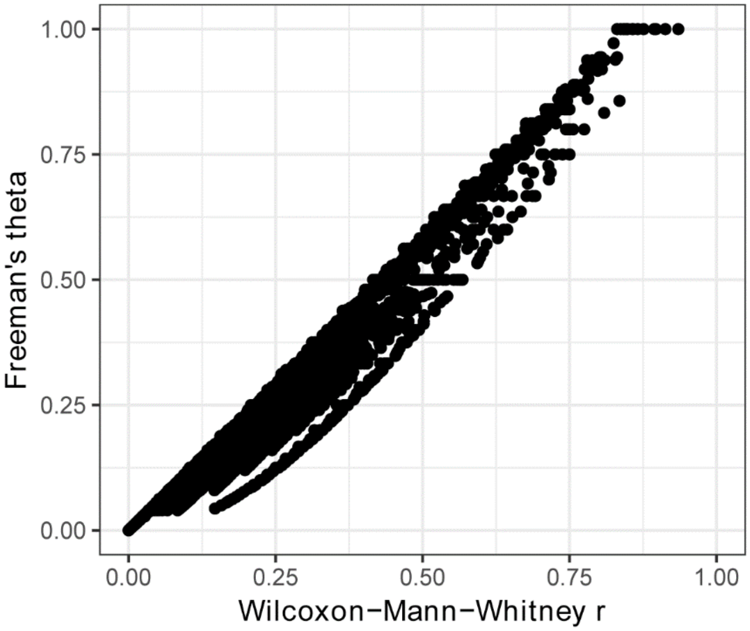
The *wilcox.test* function calculates the *z* value but doesn't report it in the output. It is sometimes more useful to report a *z* value than the *U* statistic or the *W* statistic that R reports. There are a couple of different ways to extract the *z* value.

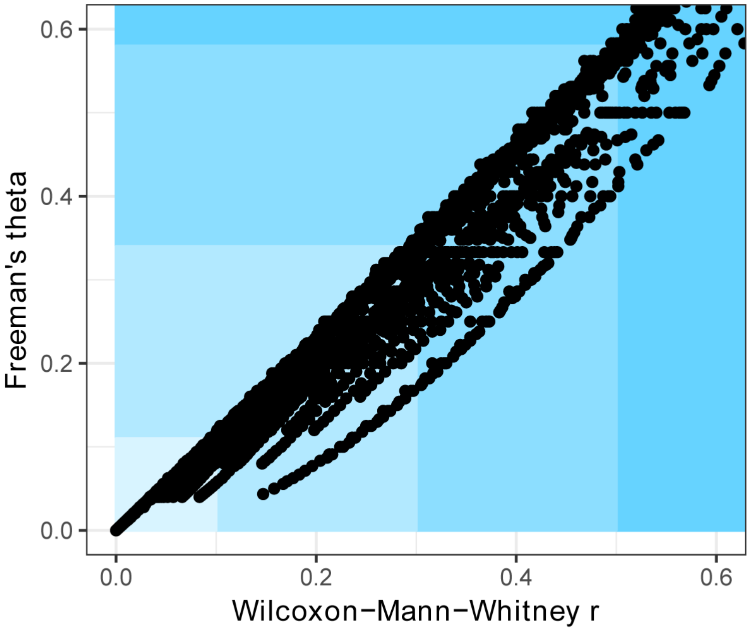
```
A = c(2, 4,  6,  8, 10, 12)
B = c(7, 9, 11, 13, 15, 17)


library(rcompanion)

wilcoxonZ(A, B)

       z
   -1.92


Y = c(A, B)

Group = factor(c(rep("A", length(A)), rep("B", length(B))))


library(coin)

wilcox_test(Y ~ Group)

   Asymptotic Wilcoxon-Mann-Whitney Test

   Z = -1.9215, p-value = 0.05466
```

Optional: Comparison among effect size statistics

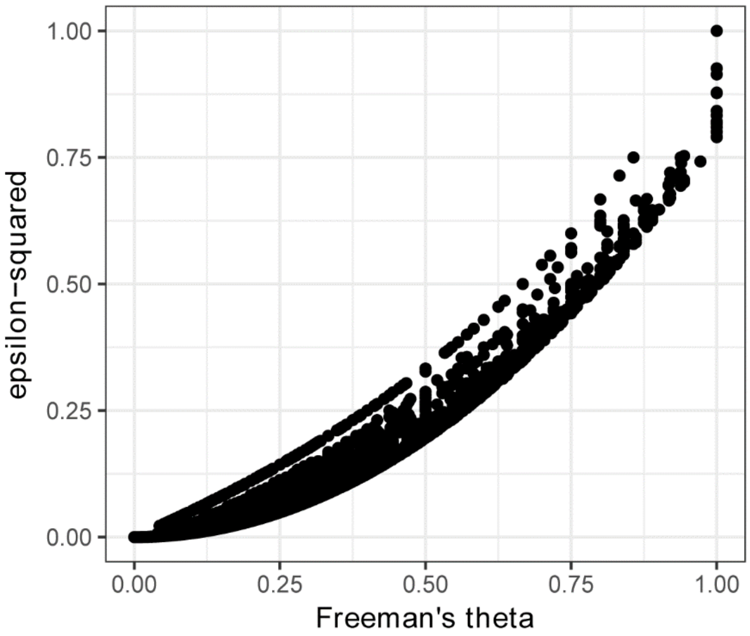
The follow plots show the relationship among some effect size statistics discussed in this chapter. Data were 5-point Likert item responses, with *n* per group between 4 and 25.

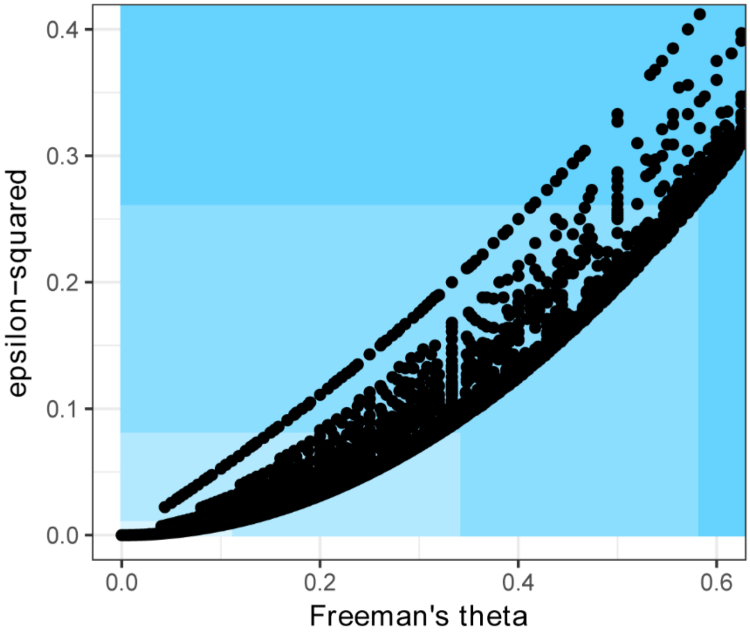
Freeman's *theta* was mostly linearly related to *r*, with variation depending on sample size and data values. In the second figure below, the colors indicate interpretation of less-than-small, small, medium, and large as the blue becomes darker.



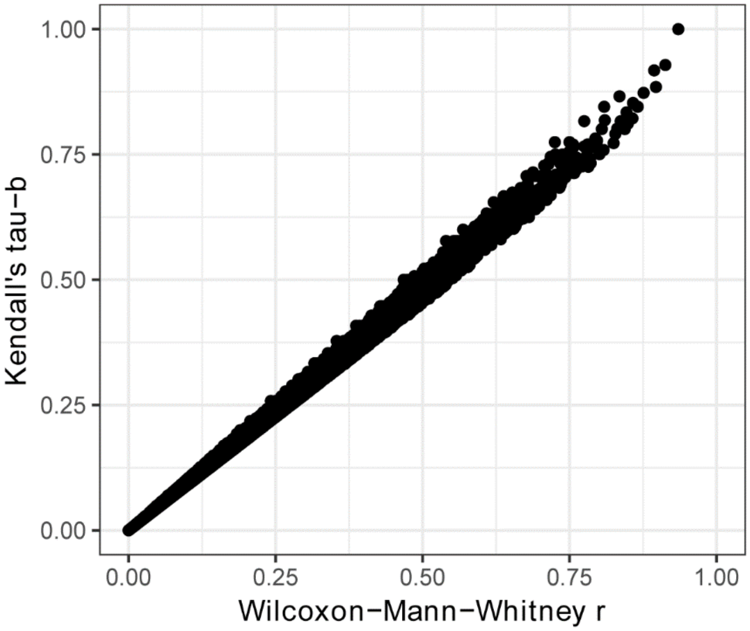


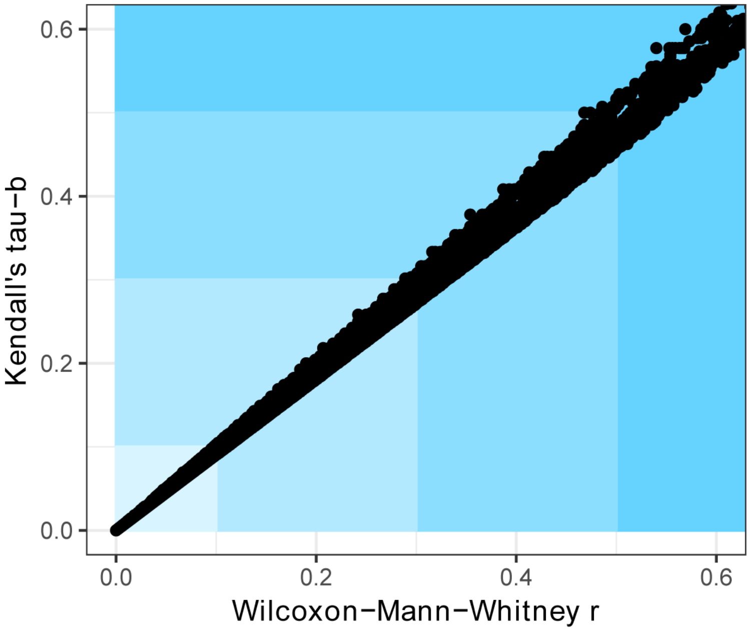
The relationship of *epsilon*-squared and Freeman's *theta* was curvilineal, with variation depending on sample size and data values. In the second figure below, the colors indicate interpretation of less-than-small, small, medium, and large as the blue becomes darker





Kendall's *tau*-b was relatively closely linearly related to *r*, up to a value of about 0.88. In second figure below, the colors indicate interpretation of less-than-small, small, medium, and large as the blue becomes darker.





References

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*, 2nd Edition. Routledge.

King, B.M., P.J. Rosopa, and E.W. Minium. 2000. *Statistical Reasoning in the Behavioral Sciences*, 6th. Wiley.

Vargha, A. and H.D. Delaney. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. 2000. *Journal of Educational and Behavioral Statistics* 25(2):101–132.

Exercises J

1. Considering Pooh and Piglet's data,

a. What was the median score for each instructor?

b. What were the first and third quartiles for each instructor's scores?

c. According to the Mann–Whitney test, is there a difference in scores between the instructors?

d. What was the value of Vargha and Delaney's *A* for the effect size for these data?

e. How do you interpret this value? (What does it mean? And is the standard interpretation in terms of "small", "medium", or "large"?)

f. How would you summarize the results of the descriptive statistics and tests? Include practical considerations of any differences.

2. Brian and Stewie Griffin want to assess the education level of students in their courses on creative writing for adults. They want to know the median education level for each class, and if the education level of the classes were different between instructors.

They used the following table to code his data.

```
Code   Abbreviation   Level

1      < HS           Less than high school
2        HS           High school
3        BA           Bachelor's
4        MA           Master's
5        PhD          Doctorate
```

The following are the course data.

```
Instructor        Student  Education
'Brian Griffin'   a        3
'Brian Griffin'   b        2
'Brian Griffin'   c        3
'Brian Griffin'   d        3
```

```
'Brian Griffin'   e        3
'Brian Griffin'   f        3
'Brian Griffin'   g        4
'Brian Griffin'   h        5
'Brian Griffin'   i        3
'Brian Griffin'   j        4
'Brian Griffin'   k        3
'Brian Griffin'   l        2
'Stewie Griffin'  m        4
'Stewie Griffin'  n        5
'Stewie Griffin'  o        4
'Stewie Griffin'  p        4
'Stewie Griffin'  q        4
'Stewie Griffin'  r        4
'Stewie Griffin'  s        3
'Stewie Griffin'  t        5
'Stewie Griffin'  u        4
'Stewie Griffin'  v        4
'Stewie Griffin'  w        3
'Stewie Griffin'  x        2
```

For each of the following, answer the question, and ***show the output from the analyses you used to answer the question***.

a. What was the median education level for each instructor? (Be sure to report the education level, not just the numeric code!)

b. What were the first and third quartiles for education level for each instructor?

c. According to the Mann–Whitney test, is there a difference in scores between the instructors?

d. What was the value of Vargha and Delaney's *A* for the effect size for these data?

e. How do you interpret this value? (What does it mean? And is the standard interpretation in terms of “small”, “medium”, or “large”?)

f. Plot Brian and Stewie's data in a way that helps you visualize the data. Do the results reflect what you would expect from looking at the plot?

g. How would you summarize the results of the descriptive statistics and tests? Include your practical interpretation.

Mood's Median Test for Two-sample Data

Mood's median test compares the medians of two or more groups. The test can be conducted with the *mood.medtest* function in the *RVAideMemoire* package or with the *median_test* function in the *coin* package or with the *nonpar* package.

Appropriate data

- One-way data with two or more groups
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with levels indicating groups
- Observations between groups are independent. That is, not paired or repeated measures data

Hypotheses

- Null hypothesis: The medians of the populations from which the groups were sampled are equal.
- Alternative hypothesis (two-sided): The medians of the populations from which the groups were sampled are not equal.

Interpretation

Significant results can be reported as "The median value of group A was significantly different from group B."

Packages used in this chapter

The packages used in this chapter include:

- RVAideMemoire
- coin
- nonpar
- FSA
- rcompanion

The following commands will install these packages if they are not already installed:

```
if(!require(RVAideMemoire)){install.packages("RVAideMemoire")}
if(!require(coin)){install.packages("coin")}
if(!require(nonpar)){install.packages("nonpar")}
if(!require(FSA)){install.packages("FSA")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

Mood's median test example

This example uses the formula notation indicating that *Likert* is the dependent variable and *Speaker* is the independent variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?mood.medtest* or the documentation for the employed function.

For appropriate plots and summary statistics, see the *Two-sample Mann–Whitney U Test* chapter.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Likert
 Pooh      3
 Pooh      5
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      5
 Pooh      5
 Piglet    2
 Piglet    4
 Piglet    2
 Piglet    2
 Piglet    1
 Piglet    2
 Piglet    3
 Piglet    2
 Piglet    2
 Piglet    3
")


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

RVAideMemoire package

```
library(RVAideMemoire)

mood.medtest(Likert ~ Speaker,
             data  = Data,
             exact = FALSE)

   Mood's median test

   X-squared = 9.8, df = 1, p-value = 0.001745
```

Coin package

```
### Median test
```

```
library(coin)

median_test(Likert ~ Speaker,
            data = Data)

   Asymptotic Two-Sample Brown-Mood Median Test

   Z = -3.4871, p-value = 0.0004883


### Exact median test

median_test(Likert ~ Speaker,
             data = Data,
             distribution="exact")

   Exact Two-Sample Brown-Mood Median Test

   Z = -3.4871, p-value = 0.001093


### Median test by Monte Carlo simulation

library(coin)

median_test(Likert ~ Speaker,
            data = Data,
            distribution = approximate(nresample = 10000))

   Approximative Two-Sample Brown-Mood Median Test

   Z = -3.4871, p-value = 0.0011
```

nonpar package

```
X = Data$Likert[Data$Speaker=="Pooh"]

Y = Data$Likert[Data$Speaker=="Piglet"]

library(nonpar)

mediantest(x = X, y = Y, exact=TRUE)

   Exact Median Test

   The p-value is  0.0010825088224469
```

Effect size measurements

A simple effect size measurement for Mood’s median test is is to compare the medians of the groups.

In addition, the whole 5-number summary could be used, including the minimum, 1st quartile, median, 3rd quartile, and the maximum.

```
library(FSA)

Summarize(Likert ~ Speaker, data=Data)

     Speaker  n mean        sd min Q1 median   Q3 max
   1  Piglet 10  2.3 0.8232726   1  2      2 2.75   4
   2    Pooh 10  4.2 0.6324555   3  4      4 4.75   5
```

Examining the medians and confidence intervals would be a somewhat different approach. Here, be cautious that confidence intervals by bootstrap may not be appropriate for the median for ordinal data with may ties, such as with Likert item data, or with small samples.

```
library(rcompanion)

groupwiseMedian(Likert ~ Speaker, data=Data, bca=FALSE, perc=TRUE)

     Speaker  n Median Conf.level Percentile.lower Percentile.upper
   1  Piglet 10      2       0.95                2                3
   2    Pooh 10      4       0.95                4                5

      ### Note that confidence intervals by bootstrap may vary.
```

In addition, looking at a statistic of stochastic dominance, like Vargha and Delaney's *A*, may be useful in this case.

```
library(rcompanion)

vda(Likert ~ Speaker, data=Data, verbose=TRUE)

             Statistic Value
   1 Proportion Ya > Yb  0.01
   2 Proportion Ya < Yb  0.91
   3    Proportion ties  0.08

    VDA

   0.05
```

Finally, we can divide the difference in medians from two groups by their pooled median absolute deviation (*mad*). Unless I find another reference for this statistic, I've termed it *Mangiafico's d*. It's somewhat analogous to a nonparametric version of Cohen's *d*. Note that this statistic assumes the data are at least interval in nature, as so may not be appropriate for Likert item data.

```
A     = c(1,2,2,2,2,3,4,5)
B     = c(2,3,4,4,4,5,5,5)
Y     = c(A, B)
Group = c(rep("A", length(A)), rep("B", length(B)))
Data2 = data.frame(Group, Y)
```

```
library(rcompanion)

mangiaficoD(Y ~ Group, data=Data2, verbose=TRUE)

     Group  Statistic  Value
   1     A     Median  2.000
   2     B     Median  4.000
   3         Difference -2.000
   4     A        MAD  0.741
   5     B        MAD  1.480
   6          Pooled MAD  1.170

       d
   -1.71
```

Manual calculation

```
MU  = median(Data$Likert)

A1  = sum(Data$Likert[Data$Speaker=="Pooh"]   <  MU)
B1  = sum(Data$Likert[Data$Speaker=="Piglet"] <  MU)
A2  = sum(Data$Likert[Data$Speaker=="Pooh"]   >= MU)
B2  = sum(Data$Likert[Data$Speaker=="Piglet"] >= MU)

Matrix = matrix(c(A1, B1, A2, B2), byrow=FALSE, ncol=2)

rownames(Matrix) = c("Pooh", "Piglet")

colnames(Matrix) = c("LessThanMu", "GreaterThanEqualMu")

Matrix

             LessThanMu  GreaterThanEqualMu
   Pooh               1                   9
   Piglet             9                   1
```

Monte Carlo simulation

```
chisq.test(Matrix, simulate.p.value=TRUE, B=10000)

   Pearson's Chi-squared test with simulated p-value (based on 10000 replicates)

   X-squared = 12.8, df = NA, p-value = 0.0009999
```

Two-sample Paired Signed-rank Test

The two-sample signed-rank test for paired data is used to compare values for two groups where each observation in one group is paired with one observation in the other group.

The test is useful to compare scores on a pre-test vs. scores on a post-test, or scores or ratings from two speakers, two different presentations, or two groups of audiences when there is a reason to pair observations, such as being done by the same rater.

A discussion of paired data can be found in the *Independent and Paired Values* chapter of this book.

Because the first step in the calculations is the subtraction of the paired values, one from the other, the data must be at least ordinal in nature.

The test is equivalent to using a one-sample signed-rank test on the difference of the paired values.

In base R, the test is performed with the *wilcox.test* function with the *paired=TRUE* option. However, the *wilcoxsign_test* in the *coin* package has the advantage of using the Pratt method to handle zero differences, which may be preferable in some cases.

Appropriate data

- Two-sample paired data. That is, one-way data with two groups only, where the observations are paired between groups.
- Dependent variable is interval, or ratio
- Independent variable is a factor with two levels. That is, two groups
- For the test to be a test of the median of the differences, the distribution of differences in paired samples needs to be symmetric

Hypotheses

- Null hypothesis: The population of the differences of paired values is symmetric around zero.
- Alternate hypothesis: (two-sided): The population of the differences of paired values is not symmetric around zero.

Interpretation

Significant results can be reported as e.g. “Values for group A were significantly different from those of group B.”

Other notes and alternative tests

Some authors recommend this test only in cases where the distribution of the differences is symmetric. It is my understanding that this requirement is only for the test to be considered a test of the median of the differences. For a little more discussion on this point, see the *One-sample Wilcoxon Signed-rank Test* chapter.

If the median is the statistic of interest, the two-sample sign test for paired data can be used.

Packages used in this chapter

The packages used in this chapter include:

- psych
- coin

- rcompanion
- exactRankTests

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(coin)){install.packages("coin")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(exactRankTests)){install.packages("exactRankTests")}
```

Two-sample paired signed-rank test example

For this example, imagine we want to compare scores for Pooh between Time 1 and Time 2. Here, we've recorded the identity of the student raters, and Pooh's score for each rater. This allows us to focus on the changes for each rater between Time 1 and Time 2. This makes for a more powerful test than would the Mann–Whitney U test in cases like this where one rater might tend to rate high and another rater might tend to rate low, but there is an overall trend in how raters change their scores between Time 1 and Time 2.

Note in this example we needed to record the identity of the student rater so that a rater's score from Time 1 can be paired with their score from Time 2. If we cannot pair data in this way—for example, if we did not record the identity of the raters—the data would have to be treated as unpaired, independent samples, for example like those in the *Two-sample Mann–Whitney U Test* chapter.

Also note that the data is arranged in long form. In this form, for this test, the data must be ordered so that the first observation where *Time* = 1 is paired to the first observation where *Time* = 2, and so on.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Time  Student  Likert
 Pooh      1      a        1
 Pooh      1      b        4
 Pooh      1      c        3
 Pooh      1      d        3
 Pooh      1      e        3
 Pooh      1      f        3
 Pooh      1      g        4
 Pooh      1      h        3
 Pooh      1      i        3
 Pooh      1      j        3
 Pooh      2      a        4
 Pooh      2      b        5
 Pooh      2      c        4
 Pooh      2      d        5
 Pooh      2      e        4
 Pooh      2      f        5
 Pooh      2      g        3
 Pooh      2      h        4
 Pooh      2      i        3
 Pooh      2      j        4
")


###  Order data by Time and Student if not already ordered

Data = Data[order(Data$Time, Data$Student),]


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

Number of observations per group
It is helpful to check the data to be sure there is one observation per *student* per *time*.

```
xtabs( ~ Student + Time,
       data = Data)
```

```
       Time
Student 1 2
      a 1 1
      b 1 1
      c 1 1
      d 1 1
      e 1 1
      f 1 1
      g 1 1
      h 1 1
      i 1 1
      j 1 1
```

Plot the paired data

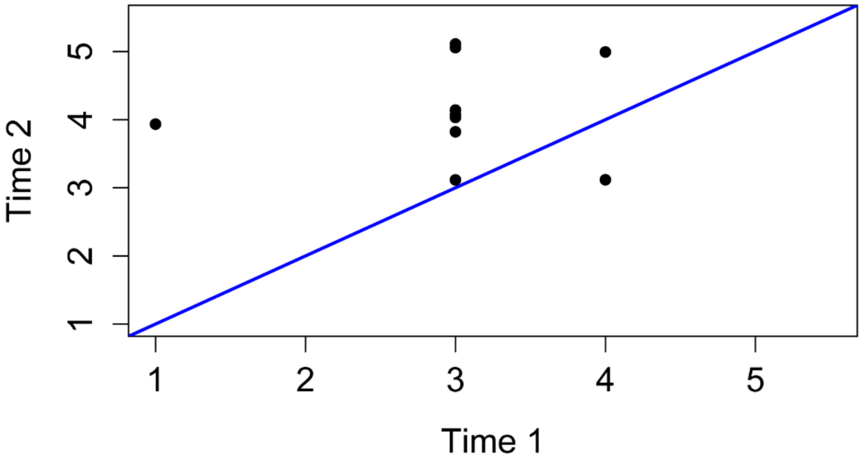
<u>Scatter plot with one-to-one line</u>
Paired data can be visualized with a scatter plot of the paired cases. In the plot below, points that fall above and to the left of the blue line indicate cases for which the value for Time 2 was greater than for Time 1.

Note that the points in the plot are jittered slightly so that points which would fall directly on top of one another can be seen.

First, two new variables, *Time.1* and *Time.2*, are created by extracting the values of *Likert* for observations with the *Time* variable equal to 1 or 2, respectively, and then the plot is produced.

Note that for this to work correctly, the data must be ordered so that the first observation where *Time* = 1 is paired to the first observation where *Time* = 2, and so on.

```
Time.1 = Data$Likert[Data$Time==1]
Time.2 = Data$Likert[Data$Time==2]


plot(Time.1, jitter(Time.2),   # jitter offsets points so you can see them all
     pch = 16,                 # shape of points
     cex = 1.0,                # size of points
      xlim=c(1, 5.5),          # limits of x axis
      ylim=c(1, 5.5),          # limits of y axis
     xlab="Time 1",
     ylab="Time 2"
     )
abline(0,1, col="blue", lwd=2) # line with intercept of 0 and slope of 1
```



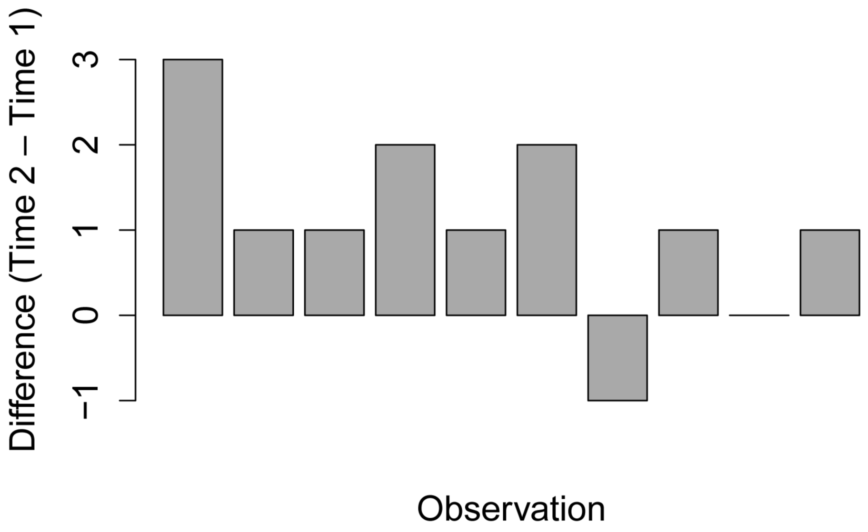
Bar plot of differences

Paired data can also be visualized with a bar chart of differences. In the plot below, bars with a value greater than zero indicate cases for which values for Time 2 are greater than for Time 1.

New variables are first created for *Time.1*, *Time.2*, and their *Difference*. And then the plot is produced.

Note that for this to work correctly, the data must be ordered so that the first observation where *Time* = 1 is paired to the first observation where *Time* = 2, and so on.

```
Time.1 = Data$Likert[Data$Time==1]
Time.2 = Data$Likert[Data$Time==2]

Difference = Time.2 - Time.1

barplot(Difference,                             # variable to plot
        col="dark gray",                        # color of bars
        xlab="Observation",                # x-axis label
        ylab="Difference (Time 2 - Time 1)")    # y-axis label
```



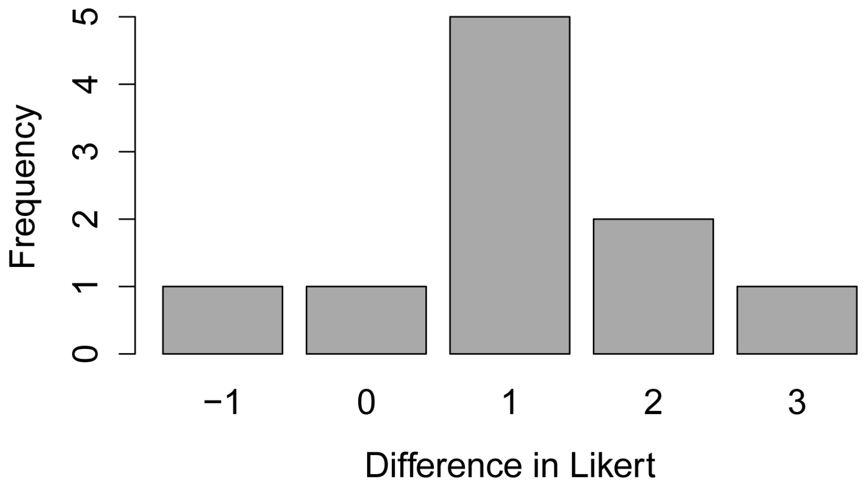
Bar plot of differences

A bar plot of differences in paired data can be used to examine the distribution of the differences.

Here, new variables are created: *Time.1*, *Time.2*, *Difference*, and *Diff.f*, which has the same values as *Difference* but as a factor variable. The *xtabs* function is used to create a count of values of *Diff.f*. The *barplot* function then uses these counts.

Note that for this to work correctly, the data must be ordered so that the first observation where *Time* = 1 is paired to the first observation where *Time* = 2, and so on.

```
Time.1 = Data$Likert[Data$Time==1]
Time.2 = Data$Likert[Data$Time==2]

Difference = Time.2 - Time.1

Diff.f = factor(Difference)

XT = xtabs(~ Diff.f)

barplot(XT,
        col="dark gray",
        xlab="Difference in Likert",
        ylab="Frequency")
```



Descriptive statistics

It is helpful to look at medians for each group and the median difference between groups in order to determine the practical importance of the differences.

```
library(FSA)

Summarize(Likert ~ Time,
          data = Data)
```

```
     Time  n mean        sd min Q1 median   Q3 max
   1    1 10  3.0 0.8164966   1  3      3 3.00   4
   2    2 10  4.1 0.7378648   3  4      4 4.75   5
```

Note that for the following to work correctly, the data must be ordered so that the first observation where *Time* = 1 is paired to the first observation where *Time* = 2, and so on.

```
Time.1 = Data$Likert[Data$Time==1]

Time.2 = Data$Likert[Data$Time==2]

Difference = Time.2 - Time.1

median(Difference)
```

```
   [1] 1
```

Two-sample paired signed-rank test

Note that if data are in long format, the data must be ordered so that the first observation of Time 1 is paired to the first observation of Time 2, and so on, because the *wilcox.test* function will take the observations in order.

This example uses the formula notation indicating that *Likert* is the dependent variable and *Time* is the independent variable. The *data=* option indicates the data frame that contains the variables, and *paired=TRUE* indicates that the test for paired data should be used. For the meaning of other options, see *?wilcox.test*.

```
wilcox.test(Likert ~ Time,
            data = Data,
            paired = TRUE,
            conf.int = TRUE,
            conf.level = 0.95)

   Wilcoxon signed rank test with continuity correction
   V = 3.5, p-value = 0.02355
   alternative hypothesis: true location shift is not equal to 0

   ### Note the p-value given in the above results

   95 percent confidence interval:
    -2.000051e+00 -1.458002e-05

   ### Confidence interval for the median or the location of differences

   ### You may get a "cannot compute exact p-value with ties" error.
   ###    You can ignore this or use the exact=FALSE option.
```

Coin package
By default, the *wilcoxsign_test* function uses the Pratt method to handle zero differences.

```
library(coin)

Time.1 = Data$Likert[Data$Time==1]

Time.2 = Data$Likert[Data$Time==2]

wilcoxsign_test(Time.1 ~ Time.2)

   Asymptotic Wilcoxon-Pratt Signed-Rank Test

   Z = -2.3522, p-value = 0.01866

   alternative hypothesis: true mu is not equal to 0
```

exactRankTests package

```
library(exactRankTests)

wilcox.exact(Likert ~ Time,
            data = Data,
            paired = TRUE,
            exact = TRUE,
            conf.int = TRUE,
            conf.level = 0.95)

   Exact Wilcoxon signed rank test

   V = 3.5, p-value = 0.02734

   95 percent confidence interval:
    -2.5 -0.5

   sample estimates:
   (pseudo)median
            -1.25
```

Effect size

The matched-pairs rank biserial correlation coefficient (*rc*) is a recommended effect size statistic. It is included in King, Rosopa, and Minimum (2000).

As an alternative, using a statistic analogous to the *r* used in the Mann–Whitney test may make sense.

The following interpretation is based on my personal intuition. It is not intended to be universal.

	small	medium	large
rc	0.10 – < 0.30	0.30 – < 0.50	≥ 0.50
r	0.10 – < 0.40	0.40 – < 0.60	≥ 0.60

rc

```
   library(rcompanion)

   wilcoxonPairedRC(x = Data$Likert,
                    g = Data$Time)

       rc
   -0.844

   ### Note that a negative rc value indicates that the second group tends to have
   ###  larger values than the first group.


   wilcoxonPairedRC(x = Data$Likert, g = Data$Time, ci=TRUE)

         rc lower.ci upper.ci
     -0.844       -1   -0.418

   ### Note that the confidence interval endpoints may vary.
```

r

```
library(rcompanion)

wilcoxonPairedR(x = Data$Likert,
                g = Data$Time)

      r
 -0.737

 ### Note that a negative r value indicates that the second group tends to have
 ###  larger values than the first group.


wilcoxonPairedR(x = Data$Likert, g = Data$Time, ci=TRUE)

          r lower.ci upper.ci
   -0.737   -0.939   -0.364

 ### Note that the confidence interval endpoints may vary.
```

References

King, B.M., P.J. Rosopa, E.W. and Minium. 2000. Statistical Reasoning in the Behavioral Sciences, 6th. Wiley.

Exercises K

1. Considering Pooh's data for Time 1 and Time 2,

a. What do the plots suggest about the relative value of the scores for Time 1 and Time 2? That is, do they suggest that scores increased, decreased, or stayed the same between Time 1 and Time 2?

b. Is the distribution of the differences between paired samples relatively symmetrical?

c. Does the two-sample paired signed-rank test indicate that there is a significant difference between Time 1 and Time 2?

d. Practically speaking, what do you conclude? If significant, is the difference between Time 1 and Time 2 of practical importance?

2. Lois Griffin gave proficiency scores to her students in her course on piano playing for adults. She gave a score for each student for their left hand playing and right hand playing. She wants to know if students in her class are more proficient in the right hand, left hand, or if there is no difference in hands.

```
Instructor       Student    Hand    Score
'Lois Griffin'   a          left    8
```

```
'Lois Griffin'   a          right  9
'Lois Griffin'   b          left   6
'Lois Griffin'   b          right  5
'Lois Griffin'   c          left   7
'Lois Griffin'   c          right  9
'Lois Griffin'   d          left   6
'Lois Griffin'   d          right  7
'Lois Griffin'   e          left   7
'Lois Griffin'   e          right  7
'Lois Griffin'   f          left   9
'Lois Griffin'   f          right  9
'Lois Griffin'   g          left   4
'Lois Griffin'   g          right  6
'Lois Griffin'   h          left   5
'Lois Griffin'   h          right  8
'Lois Griffin'   i          left   5
'Lois Griffin'   i          right  6
'Lois Griffin'   j          left   7
'Lois Griffin'   j          right  8
```

For each of the following, answer the question, and ***show the output from the analyses you used to answer the question***.

a. Is the distribution of the differences between paired samples relatively symmetrical?

b. Does the two-sample paired signed-rank test indicate that there is a difference between hands? If so, which hand received higher scores?

c. What can you conclude about the results of the plots, summary statistics, effect size, and statistical test? Practically speaking, what do you conclude? If significant, is the difference between hands of practical importance?

d. What if Lois wanted to change the design of the experiment so that she could determine if each student were more proficient in one hand or the other? That is, is student *a* more proficient in left hand or right? Is student *b* more proficient in left hand or right? How should she change what data she's collecting to determine this?

Sign Test for Two-sample Paired Data

The two-sample sign test assesses the number of observations in one group that are greater than paired observations in the other group without accounting for the magnitude of the difference. The test is similar in purpose to the two-sample Wilcoxon signed-rank test but looks specifically at the median value of differences (if the values are numeric), and is not affected by the distribution of the data.

The *SIGN.test* function in the *BSDA* package requires the data to be separated into two variables, each of which is ordered so that the first observation of each is paired, and so on. Information on options for the function can be viewed with *?SIGN.test*. The *SignTest* function in the *DescTools* package is similar.

For appropriate plots and summary statistics, see the *Two-sample Paired Signed-rank Test* chapter.

The test is equivalent to the one-sample sign test on the differences of the pairs.

Appropriate data

- Two-sample paired data. That is, one-way data with two groups only, where the observations are paired between groups.
- Dependent variable is interval or ratio. Note that because the first step in the process is to find the differences in the pairs, that the test is not appropriate for truly ordinal data.
- Independent variable is a factor with two levels. That is, two groups.

Hypotheses

- Null hypothesis: For numeric data, the median of the paired differences in the population from which the sample was drawn is equal to zero.
- Alternative hypothesis (two-sided): For numeric data, the median of the paired differences in the population from which the sample was drawn is not equal to zero.

Interpretation

Significant results can be reported as “There was a significant difference in values between group A and group B.”

Packages used in this chapter

The packages used in this chapter include:

- psych
- BSDA
- DescTools
- rcompanion

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(BSDA)){install.packages("BSDA")}
if(!require(DescTools)){install.packages("DescTools")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

Sign test for paired two-sample data example

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Time  Student  Likert
 Pooh      1     a         1
 Pooh      1     b        4
 Pooh      1     c        3
 Pooh      1     d        3
 Pooh      1     e        3
 Pooh      1     f        3
 Pooh      1     g        4
 Pooh      1     h        3
 Pooh      1     i        3
 Pooh      1     j        3
 Pooh      2     a        4
 Pooh      2     b        5
 Pooh      2     c        4
 Pooh      2     d        5
 Pooh      2     e        4
 Pooh      2     f        5
 Pooh      2     g        3
 Pooh      2     h        4
 Pooh      2     i        3
 Pooh      2     j        4
")

###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

Two-sample sign test with **DescTools** *package*

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

library(DescTools)

SignTest(x = Time.1,
         y = Time.2)

   Dependent-samples Sign-Test

   S = 1, number of differences = 9, p-value = 0.03906

      ### p-value reported above

   alternative hypothesis: true median difference is not equal to 0

   97.9 percent confidence interval:
    -2  0

   sample estimates:
   median of the differences
```

```
                          -1

      ### median of differences and confidence interval of differences
```

Two-sample sign test with BSDA *package*

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

library(BSDA)

SIGN.test(x = Time.1,
          y = Time.2,
          alternative = "two.sided",
          conf.level = 0.95)

   Dependent-samples Sign-Test

   S = 1, p-value = 0.03906

      ### p-value reported above

   95 percent confidence interval:
    -2.0000000 -0.3244444

   sample estimates:
   median of x-y
              -1

      ### median of differences and confidence interval of differences
```

Two-sample sign test with nonpar *package*

Note that the paired differences between the two groups is calculated manually, and *m=3* indicates the default value to compare to. At the time of writing, it appears that the *exact=FALSE* option actually produces the exact test.

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

Diff = Time.1 - Time.2

library(nonpar)

signtest(Diff, m=0, conf.level=0.95, exact=FALSE)

   Exact Sign Test

    The p-value is  0.03906

    The  95 % confidence interval is [ -2 ,  -1 ].
```

Effect size measurement

One effect size statistic that can be used for the paired sign test is a dominance statistic. For more information on this statistic, see the *Sign Test for One-sample Data* chapter. Note that *median* represents the median of the paired differences.

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

Diff = Time.1 - Time.2

library(rcompanion)

oneSampleDominance(Diff, mu=0)

     n Median mu Less Equal Greater Dominance  VDA
   1 10     -1  0  0.8   0.1     0.1      -0.7 0.15

   ### Note that a negative median of the diffrences, as conducted here, indicates
   that Time.2 has larger values that Time.1.  This also applies to the dominance
   statistic.


oneSampleDominance(Diff, mu=0, ci=TRUE)

     n Median mu Less Equal Greater Dominance lower.ci upper.ci  VDA lower.vda.ci upper.vda.ci
   1 10     -1  0  0.8   0.1     0.1      -0.7       -1     -0.2 0.15            0         0.45
```

It is helpful to look at the difference between medians of the two paired groups. The functions for the sign test above report the difference in medians and a confidence interval for that difference. Or this difference can be assessed manually.

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

median(Time.1)

   3


median(Time.2)

   4


Diff = Time.1 - Time.2

Median(Diff)

   -1
```

The confidence interval for the median of the differences by bootstrap can be assessed with the *groupwiseMedian* function, with the caveat that the bootstrap procedure may not be appropriate with discrete data or a small sample size.

```
Time.1 = Data$Likert [Data$Time == 1]
Time.2 = Data$Likert [Data$Time == 2]

Diff = Time.1 - Time.2

Sum = data.frame(Time.1, Time.2, Diff)

library(rcompanion)

groupwiseMedian(Diff ~ 1, data=Sum, bca=FALSE, perc=TRUE)

      .id  n Median Conf.level Percentile.lower Percentile.upper
   1 <NA> 10     -1       0.95               -2             -0.5


library(DescTools)

MedianCI(Diff, method="exact")

   median lwr.ci upr.ci
       -1     -2      0

   attr(,"conf.level")
   [1] 0.9785156
```

Manual calculations

```
Time1 = c(1, 4, 3, 3, 3, 3, 4, 3, 3, 3)

Time2 = c(4, 5, 4, 5, 4, 5, 3, 4, 3, 4)

Time1Greater = sum(Time1 > Time2)

DifferentPairs = sum(Time1 != Time2)

binom.test(Time2Greater, DifferentPairs)

   Exact binomial test

   number of successes = 8, number of trials = 9, p-value = 0.03906


N = length(Likert)

Time2GreaterProp = sum(Time2 > Time1) / N

Time2GreaterProp

   0.8
```

```
Time2LesserProp = sum(Time2 < Time1) / N

Time2LesserProp

   0.1


EqualProp = sum(Time2 == Time1) / N

EqualProp

   0.1
```

Kruskal–Wallis Test

The Kruskal–Wallis test is a rank-based test that is similar to the Mann–Whitney U test, but can be applied to one-way data with more than two groups.

Without further assumptions about the distribution of the data, the Kruskal–Wallis test does not address hypotheses about the medians of the groups. Instead, the test addresses if it is likely that an observation in one group is greater than an observation in the other. This is sometimes stated as testing if one sample has stochastic dominance compared with the other.

The test assumes that the observations are independent. That is, it is not appropriate for paired observations or repeated measures data.

It is performed with the *kruskal.test* function in the native *stats* package.

Appropriate effect size statistics include maximum Vargha and Delaney's *A*, maximum Cliff's *delta*, Freeman's *theta*, and *epsilon*-squared.

<u>Post-hoc tests</u>
The outcome of the Kruskal–Wallis test tells you if there are differences among the groups, but doesn't tell you *which* groups are different from other groups. In order to determine which groups are different from others, post-hoc testing can be conducted. Probably the most common post-hoc test for the Kruskal–Wallis test is the Dunn test (1964). Also presented are the Conover test and Nemenyi test.

<u>Appropriate data</u>

- One-way data
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with two or more levels. That is, two or more groups
- Observations between groups are independent. That is, not paired or repeated measures data

- In order to be a test of medians, the distributions of values for each group need to be of similar shape and spread. Otherwise, the test is typically a test of stochastic equality.

Hypotheses

- Null hypothesis: The groups are sampled from populations with identical distributions. Typically, that the sampled populations exhibit stochastic equality.
- Alternative hypothesis (two-sided): The groups are sampled from populations with different distributions. Typically, that one sampled population exhibits stochastic dominance.

Interpretation

Significant results can be reported as “There was a significant difference in values among groups.”

Post-hoc analysis allows you to say “There was a significant difference in values between groups A and B.”, and so on.

Other notes and alternative tests

Mood’s median test compares the medians of groups. Aligned ranks transformation anova (ART anova) provides nonparametric analysis for a variety of designs. For ordinal data, an alternative is to use cumulative link models, which are described later in this book.

Optional technical note on hypotheses for Kruskal–Wallis test

There is a lot of conflicting information on the null and alternate hypotheses for Mann–Whitney and Kruskal–Wallis tests. Some authors will state that they test medians, usually adding an assumption that the distributions of the groups need to be of the same shape and spread. If this assumption holds, then, yes, these tests can be thought of as tests of location such as the median.

Without this assumption, these tests compare the stochastic dominance of the groups. Once a rank transformation is applied, stochastic dominance is exhibited simply by the groups with higher values.

Conover (1999) adds the following assumption:

> Either the *k* population distribution functions are identical, or else some of the populations tend to yield larger values than other populations do.

My understanding is that these tests may yield significant results in the case of stochastic equality and different spread among groups. Conover’s assumption bypasses this difficulty by fiat. The upshot here is one should be careful using these tests when there is stochastic equality and different spreads among groups.

Packages used in this chapter

The packages used in this chapter include:

- psych
- FSA
- lattice
- coin

- multcompView
- rcompanion
- PMCMRplus

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(lattice)){install.packages("lattice")}
if(!require(coin)){install.packages("coin")}
if(!require(multcompView)){install.packages("multcompView")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(PMCMRplus)){install.packages("PMCMRplus")}
```

Kruskal–Wallis test example

This example re-visits the Pooh, Piglet, and Tigger data from the *Descriptive Statistics with the likert Package* chapter.

It answers the question, "Are the scores significantly different among the three speakers?"

The Kruskal–Wallis test is conducted with the *kruskal.test* function, which produces a *p*-value for the hypothesis. First the data are summarized and examined using bar plots for each group.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Likert
 Pooh      3
 Pooh      5
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      5
 Pooh      5
 Piglet    2
 Piglet    4
 Piglet    2
 Piglet    2
 Piglet    1
 Piglet    2
 Piglet    3
 Piglet    2
 Piglet    2
 Piglet    3
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    5
 Tigger    3
 Tigger    5
 Tigger    4
 Tigger    4
 Tigger    3
")


### Order levels of the factor; otherwise R will alphabetize them

Data$Speaker = factor(Data$Speaker,
                      levels=unique(Data$Speaker))


### Create a new variable which is the likert scores as an ordered factor

Data$Likert.f = factor(Data$Likert,
                       ordered = TRUE)


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

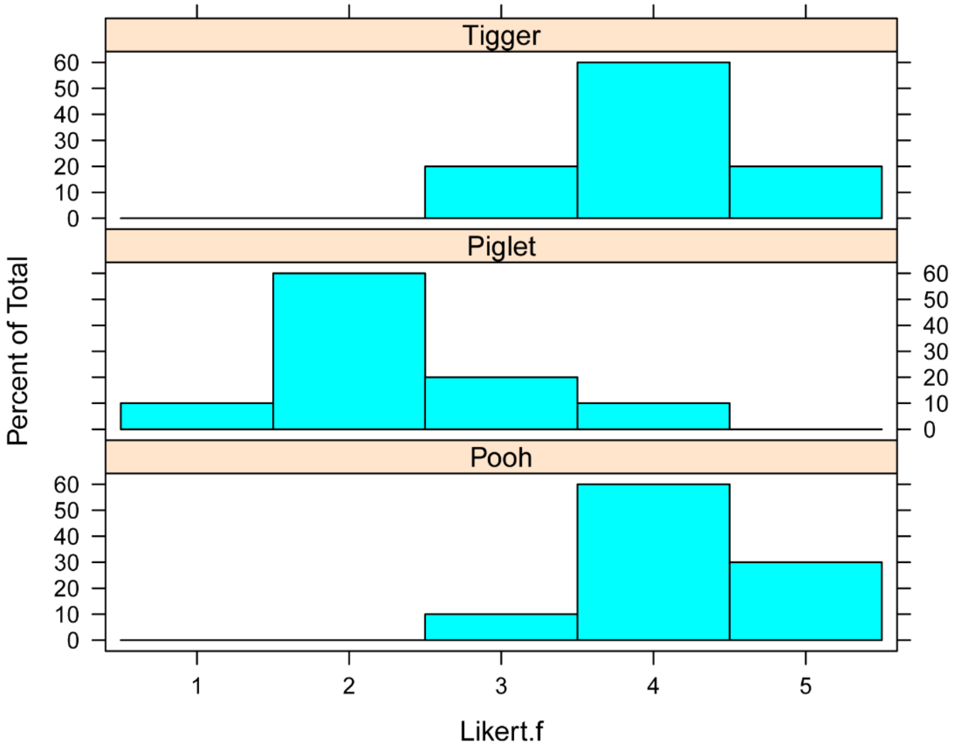
Summarize data treating Likert scores as factors

```
xtabs( ~ Speaker + Likert.f,
       data = Data)

           Likert.f
   Speaker  1 2 3 4 5
     Pooh   0 0 1 6 3
     Piglet 1 6 2 1 0
     Tigger 0 0 2 6 2


XT = xtabs( ~ Speaker + Likert.f,
            data = Data)

prop.table(XT,
           margin = 1)

           Likert.f
   Speaker    1   2   3   4   5
     Pooh   0.0 0.0 0.1 0.6 0.3
     Piglet 0.1 0.6 0.2 0.1 0.0
```

```
      Tigger 0.0 0.0 0.2 0.6 0.2
```

Bar plots of data by group

```
library(lattice)

histogram(~ Likert.f | Speaker,
          data=Data,
          layout=c(1,3)      #  columns and rows of individual plots
          )
```



Summarize data treating Likert scores as numeric

```
library(FSA)

Summarize(Likert ~ Speaker,
          data=Data,
          digits=3)

    Speaker  n mean    sd min Q1 median   Q3 max percZero
  1    Pooh 10  4.2 0.632   3  4      4 4.75   5        0
  2  Piglet 10  2.3 0.823   1  2      2 2.75   4        0
  3  Tigger 10  4.0 0.667   3  4      4 4.00   5        0
```

Kruskal–Wallis test example

This example uses the formula notation indicating that *Likert* is the dependent variable and *Speaker* is the independent variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?kruskal.test*.

```
kruskal.test(Likert ~ Speaker,
             data = Data)

   Kruskal-Wallis rank sum test

   Kruskal-Wallis chi-squared = 16.842, df = 2, p-value = 0.0002202
```

As an alternative, the Kruskal–Wallis test can be conducted by Monte Carlo simulation with the *coin* package.

```
library(coin)

kruskal_test(Likert ~ Speaker, data = Data,
             distribution = approximate(nresample = 10000))

   Approximative Kruskal-Wallis Test

   chi-squared = 16.842, p-value < 1e-04
```

Effect size statistics

Common effect size statistics for the Kruskal–Wallis test include *epsilon*-squared and *eta*-squared, with *epsilon*-squared being probably the most common. These two statistics often produce similar results. It's important to note that both *epsilon*-squared and *eta*-squared have versions for ANOVA that are distinct from those versions for Kruskal–Wallis. It is therefore important to be sure that an appropriate implementation is chosen. Formulae for these statistics can be found in King and others (2018) and Cohen (2013), respectively.

Technical note: Interestingly, it appears that the *epsilon*-squared statistic for Kruskal–Wallis corresponds to the *eta*-squared statistic for ANOVA on ranks. However, the definitions presented here appear to be well-established.

For Freeman's *theta*, an effect size of 1 indicates that the measurements for each group are entirely greater or entirely less than some other group, and an effect size of 0 indicates that there is no effect; that is, that the groups are absolutely stochastically equal.

Another option is to use the maximum Cliff's *delta* or Vargha and Delaney's *A* (VDA) from pairwise comparisons of all groups. VDA is the probability that an observation from one group is greater than an observation from the other group. Because of this interpretation, VDA is an effect size statistic that is relatively easy to understand.

Interpretation of effect sizes necessarily varies by discipline and the expectations of the experiment. The following guidelines are based on my personal intuition or published values. They should not be considered universal.

Technical note: The values for the interpretations for Freeman's *theta* to *epsilon*-squared below were derived by keeping the interpretation for *epsilon*-squared constant and equal to that for the Mann–Whitney test. Interpretation values for Freeman's *theta* were determined through comparing Freeman's *theta* to *epsilon*-squared for simulated data (5-point Likert items, *n* per group between 4 and 25).

Interpretations for Vargha and Delaney's *A* and Cliff's *delta* come from Vargha and Delaney (2000).

	small	medium	large
***epsilon*-squared**	0.01 – < 0.08	0.08 – < 0.26	≥ 0.26
Freeman's *theta*, *k* = 2	0.11 – < 0.34	0.34 – < 0.58	≥ 0.58
Freeman's *theta*, *k* = 3	0.05 – < 0.26	0.26 – < 0.46	≥ 0.46
Freeman's *theta*, *k* = 5	0.05 – < 0.21	0.21 – < 0.40	≥ 0.40
Freeman's *theta*, *k* = 7	0.05 – < 0.20	0.20 – < 0.38	≥ 0.38
Freeman's *theta*, *k* = 7	0.05 – < 0.20	0.20 – < 0.38	≥ 0.38
Maximum Cliff's *delta*	0.11 – < 0.28	0.28 – < 0.43	≥ 0.43
Maximum Vargha and Delaney's *A*	0.56 – < 0.64 > 0.34 – 0.44	0.64 – < 0.71 > 0.29 – 0.34	≥ 0.71 ≤ 0.29

Ordinal *epsilon*-squared

```
library(rcompanion)

epsilonSquared(x = Data$Likert,
               g = Data$Speaker)

   epsilon.squared
             0.581


epsilonSquared(x = Data$Likert,
               g = Data$Speaker,
               ci=TRUE)

   ### Confidence interval endpoints may vary

     epsilon.squared lower.ci upper.ci
   1           0.581    0.325    0.805
```

r-squared for anova on ranks

```
summary(lm(rank(Likert) ~ Speaker, data=Data))$r.squared

   [1] 0.5807692
```

Ordinal *eta*-squared

```
library(rcompanion)

ordinalEtaSquared(x = Data$Likert,
                  g = Data$Speaker)

   eta.squared
          0.55


ordinalEtaSquared (x  = Data$Likert,
                   g  = Data$Speaker,
                   ci = TRUE)

     eta.squared lower.ci upper.ci
   1        0.55    0.293    0.794

      ### Confidence interval endpoints may vary
```

adjusted r-squared for anova on ranks

```
summary(lm(rank(Likert) ~ Speaker, data=Data))$adj.r.squared

   [1] 0.5497151
```

Freeman's *theta*

```
library(rcompanion)

freemanTheta(x = Data$Likert,
             g = Data$Speaker)

   Freeman.theta
            0.64


freemanTheta(x = Data$Likert,
             g = Data$Speaker,
             ci = TRUE)

   ### Confidence interval endpoints may vary

     Freeman.theta lower.ci upper.ci
   1          0.64    0.445    0.84
```

Maximum Vargha and Delaney's *A* or Cliff's *delta*

Here, the *multiVDA* function is used to calculate Vargha and Delaney's *A* (VDA), Cliff's *delta* (CD), and the Glass rank biserial correlation coefficient (*rg*) between all pairs of groups. The function identifies the comparison with the most extreme VDA statistic (0.95 for *Pooh* – *Piglet*). That is, it identifies the most disparate groups.

```
library(rcompanion)

multiVDA(x = Data$Likert,
         g = Data$Speaker)

   $pairs
          Comparison  VDA    CD    rg  VDA.m CD.m rg.m
   1   Pooh - Piglet 0.95  0.90  0.90  0.95 0.90 0.90
   2   Pooh - Tigger 0.58  0.16  0.16  0.58 0.16 0.16
   3 Piglet - Tigger 0.07 -0.86 -0.86  0.93 0.86 0.86

   $comparison
        Comparison
   "Pooh - Piglet"

   $statistic
    VDA
   0.95

   $statistic.m
   VDA.m
    0.95
```

Post-hoc test: Dunn test for multiple comparisons of groups

If the Kruskal–Wallis test is significant, a post-hoc analysis can be performed to determine which groups differ from each other group.

Probably the most popular post-hoc test for the Kruskal–Wallis test is the Dunn test. Also presented are the Conover test and Nemenyi test.

Because the post-hoc test will produce multiple *p*-values, adjustments to the *p*-values can be made to avoid inflating the possibility of making a type-I error. There are a variety of methods for controlling the familywise error rate or for controlling the false discovery rate. See *?p.adjust* for details on these methods.

When there are many *p*-values to evaluate, it is useful to condense a table of *p*-values to a compact letter display format. In the output, groups are separated by letters. Groups sharing the same letter are not significantly different. Compact letter displays are a clear and succinct way to present results of multiple comparisons. However, they suffer from presenting “non-differences” instead of differences among groups, and they obscure the actual *p*-values, which are more informative than just a $p \leq 0.05$ cutoff.

```
### Order groups by median

Data$Speaker = factor(Data$Speaker,
                      levels=c("Pooh", "Tigger", "Piglet"))
```

```
levels(Data$Speaker)

   ### At the time of writing, it doesn’t appear that ordering the groups affects
   ###  the dunnTest output.

### Dunn test

library(FSA)

DT = dunnTest(Likert ~ Speaker,
              data=Data,
              method="bh")      # Adjusts p-values for multiple comparisons;
                                # See ?dunnTest for options

DT

   Dunn (1964) Kruskal-Wallis multiple comparison
     p-values adjusted with the Benjamini-Hochberg method.

          Comparison          Z      P.unadj        P.adj
   1   Piglet - Pooh -3.7702412 0.0001630898 0.0004892695
   2 Piglet - Tigger -3.2889338 0.0010056766 0.0015085149
   3   Pooh - Tigger  0.4813074 0.6302980448 0.6302980448


### Compact letter display

PT = DT$res

PT

library(rcompanion)

cldList(P.adj ~ Comparison,
        data = PT,
        threshold = 0.05)

      Group Letter MonoLetter
   1 Piglet      a          a
   2   Pooh      b           b
   3 Tigger      b           b

   ### Groups sharing a letter not signficantly different (alpha = 0.05).
```

Post-hoc tests: Dunn test, Conover test, and Nemenyi test

```
### Order groups by median

Data$Speaker = factor(Data$Speaker,
                      levels=c("Pooh", "Tigger", "Piglet"))

levels(Data$Speaker)
```

Dunn test

```
library(PMCMRplus)

DT = kwAllPairsDunnTest(Likert ~ Speaker, data=Data, method="bh")

DT

   Pairwise comparisons using Dunn's all-pairs test

          Pooh    Tigger
   Tigger 0.63030 -
   Piglet 0.00049 0.00201

   P value adjustment method: holm


library(rcompanion)

DTT =PMCMRTable(DT)

DTT

              Comparison  p.value
   1   Tigger - Pooh = 0     0.63
   2   Piglet - Pooh = 0 0.000489
   3 Piglet - Tigger = 0  0.00201


cldList(p.value ~ Comparison, data=DTT)

      Group Letter MonoLetter
   1 Tigger      a         a
   2 Piglet      b          b
   3   Pooh      a         a
```

Conover test

```
library(PMCMRplus)

CT = kwAllPairsConoverTest(Likert ~ Speaker, data=Data)

CT

   Pairwise comparisons using Conover's all-pairs test

          Pooh    Tigger
   Tigger 0.75549 -
   Piglet 1.7e-05 0.00011

   P value adjustment method: single-step
```

```
library(rcompanion)

CTT =PMCMRTable(CT)

CTT

             Comparison  p.value
1   Tigger - Pooh = 0    0.755
2   Piglet - Pooh = 0 1.69e-05
3 Piglet - Tigger = 0 0.000115


cldList(p.value ~ Comparison, data=CTT)

   Group Letter MonoLetter
1 Tigger      a          a
2 Piglet      b           b
3   Pooh      a          a
```

Nemenyi test

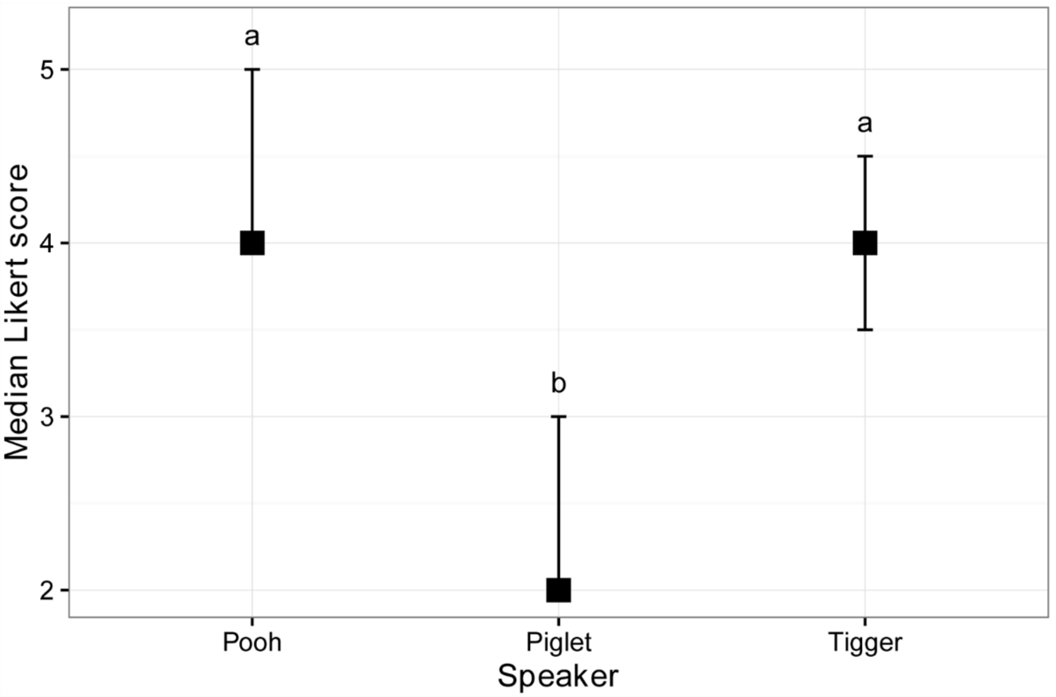
```
library(PMCMRplus)

NT = kwAllPairsNemenyiTest(Likert ~ Speaker, data=Data)

NT

   Pairwise comparisons using Tukey-Kramer-Nemenyi all-pairs test with Tukey-Dist
   approximation

          Pooh   Tigger
   Tigger 0.8912 -
   Piglet 0.0010 0.0051

   P value adjustment method: single-step


library(rcompanion)

NTT =PMCMRTable(NT)

NTT
             Comparison  p.value
1   Tigger - Pooh = 0    0.891
2   Piglet - Pooh = 0 0.000996
3 Piglet - Tigger = 0  0.00507


cldList(p.value ~ Comparison, data=NTT)

   Group Letter MonoLetter
1 Tigger      a          a
2 Piglet      b           b
3   Pooh      a          a
```

Plot of medians and confidence intervals

The following code uses the *groupwiseMedian* function to produce a data frame of medians for each speaker along with the 95% confidence intervals for each median with the percentile method. These medians are then plotted, with their confidence intervals shown as error bars. The grouping letters from the multiple comparisons (Dunn test) are added.

Note that bootstrapped confidence intervals may not be reliable for discreet data, such as the ordinal Likert data used in these examples, especially for small samples.

```
library(rcompanion)

Sum = groupwiseMedian(Likert ~ Speaker,
                      data       = Data,
                      conf       = 0.95,
                      R          = 5000,
                      percentile = TRUE,
                      bca        = FALSE,
                      digits     = 3)

Sum

     Speaker  n Median Conf.level Percentile.lower Percentile.upper
   1    Pooh 10      4       0.95              4.0              5.0
   2  Piglet 10      2       0.95              2.0              3.0
   3  Tigger 10      4       0.95              3.5              4.5


X     = 1:3
Y     = Sum$Percentile.upper + 0.2
Label = c("a", "b", "a")


library(ggplot2)

ggplot(Sum,                ### The data frame to use.
       aes(x = Speaker,
           y = Median)) +
   geom_errorbar(aes(ymin = Percentile.lower,
                     ymax = Percentile.upper),
                     width = 0.05,
                     size  = 0.5) +
   geom_point(shape = 15,
              size  = 4) +
   theme_bw() +
   theme(axis.title   = element_text(face  = "bold")) +

   ylab("Median Likert score") +

annotate("text",
         x = X,
         y = Y,
         label = Label)
```



Plot of median Likert score versus Speaker. Error bars indicate the 95% confidence intervals for the median with the percentile method.

Optional technical note: Schwenk dice and pairwise Wilcoxon–Mann–Whitney tests

When choosing a post-hoc test, it is often tempting to use pairwise tests, usually with a *p*-value adjustment to control the familywise error rate or the false discovery rate. One issue with pairwise tests is that they ignore all the data that isn't included in that pair of treatments. It is often better to use a post-hoc test designed to assess pairs of treatments based on the totality of the data.

In the case of the Kruskal–Wallis test, it is often desirable to use e.g. the Dunn (1964) test that preserves the rankings from all of the original data, rather than to use pairwise Mann–Whitney tests where each test would re-rank the data based only on that pair of treatments being compared.

An interesting example of the difference between these two approaches are non-transitive dice. One example is Schwenk's dice, which are described in Futility Closet (2018) and originally by Schwenk (2000).

Imagine three six-sided dice, which have the following numbers marked on each of the sides. We want to use rank-based statistics to determine which die tends to have higher numbers.

```
Red    = c(2,2,2,11,11,14)
```

```
Blue  = c(0,3,3,12,12,12)
Green = c(1,1,1,13,13,13)
```

A simple way to determine which die is stochastically larger is to use Cliff's *delta*.

```
library(rcompanion)

cliffDelta(x=Red, y=Blue)

   Cliff.delta
        -0.167

   ### The negative Cliff's delta value suggests the values in Blue tend to be
   ###   greater than those in Red.

   ### Specifically, when rolled, the Blue die beats the Red 7 / 12 times.

   ###  ( -0.167 / 2 + 0.5 = 0.4165 = 5 / 12 )
   ###  vda(x=Red, y=Blue) = 0.4165


cliffDelta(x=Red, y=Green)

   Cliff.delta
        0.167

   ### The positive Cliff's delta value suggests the values in Red tend to be
   ###   greater than those in Green.

   ### Specifically, when rolled, the Red die beats the Green 7 / 12 times.


cliffDelta(x=Blue, y=Green)

   Cliff.delta
        -0.167

   ### The positive Cliff's delta value suggests the values in Green tend to be
   ###   greater than those in Blue.

   ### Specifically, when rolled, the Green die beats the Blue 7 / 12 times.
```

So, in summary, Blue > Red, Red > Green, and Green > Blue. When viewed together, these results would be difficult to interpret, or, honestly, to make sense of.

The same could be shown with pairwise Mann–Whitney tests, if we kept the same data, but increase the sample size. I'll refer to this new data set as representing three "groups".

```
RedBig   = rep(c(2,2,2,11,11,14), 25)
BlueBig  = rep(c(0,3,3,12,12,12), 25)
GreenBig = rep(c(1,1,1,13,13,13), 25)
```

```
wilcox.test(RedBig, BlueBig)

   W = 9375, p-value = 0.01081

wilcoxonZ(RedBig, BlueBig)

   z
   -2.55

   ### Blue > Red


wilcox.test(RedBig, GreenBig)

   W = 13125, p-value = 0.01038

wilcoxonZ(RedBig, GreenBig)

   z
   2.56

   ### Red > Green


wilcox.test(BlueBig, GreenBig)

   W = 9375, p-value = 0.01038

wilcoxonZ(BlueBig, GreenBig)

   z
   -2.56

   ### Green > Blue
```

On the other hand, because the Dunn test retains the ranking from all three groups together, it won't find a difference among any of the groups.

```
Y = c(RedBig, BlueBig, GreenBig)

Group = c(rep("Red", length(RedBig)), rep("Blue", length(BlueBig)), rep("Green",
length(GreenBig)))


library(FSA)

dunnTest(Y ~ Group)

   Dunn (1964) Kruskal-Wallis multiple comparison
     p-values adjusted with the Holm method.
```

```
    Comparison Z P.unadj P.adj
1 Blue - Green 0       1     1
2   Blue - Red 0       1     1
3  Green - Red 0       1     1
```

Of course, in this case, the omnibus Kruskal–Wallis test would also find no stochastic differences among the groups.

```
kruskal.test(Y ~ Group)

   Kruskal-Wallis rank sum test

   Kruskal-Wallis chi-squared = 0, df = 2, p-value = 1
```

References

Cohen, B.H. 2013. Explaining Psychological Statistics, 4th. Wiley.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*, 2nd Edition. Routledge.

Conover, W.J. 1999. Practical Nonparametric Statistics, 3rd. John Wiley & Sons.

Futility Closet. 2018. "Schwenk Dice". https://www.futilitycloset.com/2018/06/27/schwenk-dice/.

King, B.M., P.J. Rosopa, and E.W. Minium. 2018. Some (Almost) Assumption-Free Tests. In *Statistical Reasoning in the Behavioral Sciences*, 7th ed. Wiley.

Schwenk, A.J. 2000. Beware of Geeks Bearing Grifts. *Math Horizons* 7(4): 10–13.

Tomczak, M. and Tomczak, E. 2014. The need to report effect size estimates revisited. An overview of some recommended measures of effect size. Trends in Sports Sciences 1(21):1–25. www.tss.awf.poznan.pl/files/3_Trends_Vol21_2014__no1_20.pdf.

Vargha, A. and H.D. Delaney. A Critique and Improvement of the CL Common Language Effect Size Statistics of McGraw and Wong. 2000. Journal of Educational and Behavioral Statistics 25(2):101–132.

Exercises L

1. Considering Pooh, Piglet, and Tigger's data,

   a. What was the median score for each instructor?

b. According to the Kruskal–Wallis test, is there a statistical difference in scores among the instructors?

c. What is the value of maximum Vargha and Delaney's *A* for these data?

d. How do you interpret this value? (What does it mean? And is the standard interpretation in terms of "small", "medium", or "large"?)

e. Looking at the post-hoc analysis, which speakers' scores are statistically different from which others? Who had the statistically highest scores?

f. How would you summarize the results of the descriptive statistics and tests? Include practical considerations of any differences.

2. Brian, Stewie, and Meg want to assess the education level of students in their courses on creative writing for adults. They want to know the median education level for each class, and if the education level of the classes were different among instructors.

They used the following table to code his data.

```
Code   Abbreviation   Level

1      < HS           Less than high school
2        HS           High school
3        BA           Bachelor's
4        MA           Master's
5        PhD          Doctorate
```

The following are the course data.

```
Instructor        Student  Education
'Brian Griffin'   a        3
'Brian Griffin'   b        2
'Brian Griffin'   c        3
'Brian Griffin'   d        3
'Brian Griffin'   e        3
'Brian Griffin'   f        3
'Brian Griffin'   g        4
'Brian Griffin'   h        5
'Brian Griffin'   i        3
'Brian Griffin'   j        4
'Brian Griffin'   k        3
'Brian Griffin'   l        2
'Stewie Griffin'  m        4
'Stewie Griffin'  n        5
'Stewie Griffin'  o        4
'Stewie Griffin'  p        4
'Stewie Griffin'  q        4
'Stewie Griffin'  r        4
'Stewie Griffin'  s        3
```

```
'Stewie Griffin'  t        5
'Stewie Griffin'  u        4
'Stewie Griffin'  v        4
'Stewie Griffin'  w        3
'Stewie Griffin'  x        2
'Meg Griffin'     y        3
'Meg Griffin'     z        4
'Meg Griffin'     aa       3
'Meg Griffin'     ab       3
'Meg Griffin'     ac       3
'Meg Griffin'     ad       2
'Meg Griffin'     ae       3
'Meg Griffin'     af       4
'Meg Griffin'     ag       2
'Meg Griffin'     ah       3
'Meg Griffin'     ai       2
'Meg Griffin'     aj       1
```

For each of the following, answer the question, and ***show the output from the analyses you used to answer the question***.

a. What was the median education level for each instructor's class? (Be sure to report the education level, not just the numeric code!)

b. According to the Kruskal–Wallis test, is there a difference in the education level of students among the instructors?

c. What is the value of maximum Vargha and Delaney's *A* for these data?

d. How do you interpret this value? (What does it mean? And is the standard interpretation in terms of "small", "medium", or "large"?)

e. Looking at the post-hoc analysis, which classes education levels are statistically different from which others? Who had the statistically highest education level?

f. Plot Brian, Stewie, and Meg's data in a way that helps you visualize the data. Do the results reflect what you would expect from looking at the plot?

g. How would you summarize the results of the descriptive statistics and tests? What do you conclude practically?

Mood's Median Test

Mood's median test compares the medians of two or more groups. The test can be conducted with the *mood.medtest* function in the *RVAideMemoire* package or with the *median_test* function in the *coin* package.

Post-hoc tests

The outcome of Mood's median test tells you if there are differences among the groups, but doesn't tell you *which* groups are different from other groups. In order to determine which groups are different from others, post-hoc testing can be conducted. The function *pairwiseMedianTest* in the *rcompanion* package can perform the post-hoc tests. It simply passes data for pairs of groups to *coin::median_test* and produces a table of output.

Appropriate data

- One-way data with two or more groups
- Dependent variable is ordinal, interval, or ratio
- Independent variable is a factor with levels indicating groups
- Observations between groups are independent. That is, not paired or repeated measures data

Hypotheses

- Null hypothesis: The medians of the populations from which the groups were sampled are equal.
- Alternative hypothesis (two-sided): The medians of the populations from which the groups were sampled are not all equal.

Interpretation

Significant results can be reported as "There was a significant difference in the median values among groups."

Post-hoc analysis allows you to say "The median for group A was higher than the median for group B", and so on.

Packages used in this chapter

The packages used in this chapter include:

- RVAideMemoire
- coin
- rcompanion

The following commands will install these packages if they are not already installed:

```
if(!require(RVAideMemoire)){install.packages("RVAideMemoire")}
if(!require(coin)){install.packages("coin")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

Mood's median test example

This example uses the formula notation indicating that *Likert* is the dependent variable and *Speaker* is the independent variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?mood.medtest* or the documentation for the employed function.

A significant *p*-value for Mood's median test indicates that not all medians among groups are equal.

For appropriate plots and summary statistics, see the *Kruskal–Wallis Test* chapter.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Likert
 Pooh      3
 Pooh      5
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      5
 Pooh      5
 Piglet    2
 Piglet    4
 Piglet    2
 Piglet    2
 Piglet    1
 Piglet    2
 Piglet    3
 Piglet    2
 Piglet    2
 Piglet    3
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    5
 Tigger    3
 Tigger    5
 Tigger    4
 Tigger    4
 Tigger    3
")


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

RVAideMemoire package

```
   library(RVAideMemoire)

   mood.medtest(Likert ~ Speaker,
                data  = Data,
                exact = FALSE)

      Mood's median test

      X-squared = 3.36, df = 2, p-value = 0.1864
```

An interesting thing happened with the result here. The test counts how many observations in each group are greater than the global median for all groups together, in this case 4. It then tests if there is a significant difference in this proportion among groups. For this data set, however, both Pooh and Tigger have a majority of observations equal to the global median. Because they are *equal* to the global median, they are not *greater than* the global median, and so aren't much different than Piglet's scores on this count. The result in this case is a non-significant *p*-value.

But the test would come out differently if we were counting observation *less than* the global median, because Pooh and Tigger have few of these, and Piglet has relatively many.

This is a quirk with Mood's median test, and isn't common in statistical tests.

One solution would be to re-code the function to count observations less than the global median.

But it is easier to simply invert the scale we are using. This is really an arbitrary change, but for this test, it can make a difference. Imagine if our original scale interpreted 5 to be the best, and 1 to be the worst. When we designed the survey tool, we could just as easily have made 1 the best and 5 the worst. And then instead of ranking "good" with a 4, the respondents would have marked it 2, and so on. By the way the calculations are done, this arbitrary change in scale will change the results of Mood's median test.

For a 5-point scale, we do this inversion by simply by making a new variable equal to 6 minus the original score.

With Mood's median test, I recommend making this kind of inversion in cases where many values are equal to the global median. Then use whichever result has a lower *p*-value.

```
   Data$Likert.inv = 6 - Data$Likert

   library(psych)

   headTail(Data)

          Speaker Likert Likert.inv
      1      Pooh      3          3
   2      Pooh      5         1
   3      Pooh      4         2
   4      Pooh      4         2
   ...    <NA>     ...       ...
   27   Tigger      5         1
   28   Tigger      4         2
   29   Tigger      4         2
   30   Tigger      3         3
```

```
library(RVAideMemoire)

mood.medtest(Likert.inv ~ Speaker,
             data = Data,
             exact = FALSE)

   Mood's median test

   X-squared = 15.833, df = 2, p-value = 0.0003646
```

Coin package

```
### Median test

library(coin)

median_test(Likert.inv ~ Speaker,
            data = Data)

   Asymptotic K-Sample Brown-Mood Median Test

   chi-squared = 15.306, df = 2, p-value = 0.0004747


### Median test by Monte Carlo simulation

library(coin)

median_test(Likert.inv ~ Speaker,
            data = Data,
            distribution = approximate(nresample = 10000))

   Approximative K-Sample Brown-Mood Median Test

   chi-squared = 15.306, p-value = 3e-04
```

Post-hoc: pairwise median tests

If Mood's median test is significant, a post-hoc analysis can be performed to determine which groups differ from each other group.

For this we will use the *pairwiseMedianTest* function in the *rcompanion* package, which conducts Mood's median test on all pairs of groups from one-way data.

Because the post-hoc test will produce multiple *p*-values, adjustments to the *p*-values can be made to avoid inflating the possibility of making a type-I error. There are a variety of methods for controlling the familywise error rate or for controlling the false discovery rate. See *?p.adjust* for details on these methods.

<u>*pairwiseMedianTest* function</u>

```
### Order groups by median

Data$Speaker = factor(Data$Speaker,
                      levels=c("Pooh", "Tigger", "Piglet"))


### Pairwise median tests

library(rcompanion)

PT = pairwiseMedianTest(Likert.inv ~ Speaker,
                        data   = Data,
                        exact  = NULL,
                        method = "fdr")
                              # Adjusts p-values for multiple comparisons;
                              # See ?p.adjust for options

PT

                 Comparison   p.value p.adjust
   1   Pooh - Tigger = 0    0.5416 0.541600
   2   Pooh - Piglet = 0 0.0004883 0.001465
   3 Tigger - Piglet = 0  0.001381 0.002072


### Compact letter display

library(rcompanion)

cldList(p.adjust ~ Comparison,
        data = PT,
        threshold = 0.05)

      Group Letter MonoLetter
   1   Pooh      a          a
   2 Tigger      a          a
   3 Piglet      b           b

   Groups sharing a letter are not significantly different (alpha = 0.05).
```

pairwiseMedianMatrix function

```
### Order groups by median

Data$Speaker = factor(Data$Speaker,
                      levels=c("Pooh", "Tigger", "Piglet"))


### Pairwise median tests

library(rcompanion)

PT = pairwiseMedianMatrix(Likert.inv ~ Speaker,
                          data   = Data,
                          exact  = NULL,
                          method = "fdr")
                                # Adjusts p-values for multiple comparisons;
                                # See ?p.adjust for options

PT

   $Unadjusted
          Pooh Tigger    Piglet
   Pooh     NA 0.5416 0.0004883
   Tigger   NA     NA 0.0013810
   Piglet   NA     NA        NA

   $Method
   [1] "fdr"

   $Adjusted
              Pooh   Tigger   Piglet
   Pooh   1.000000 0.541600 0.001465
   Tigger 0.541600 1.000000 0.002072
   Piglet 0.001465 0.002072 1.000000


### Compact letter display

library(multcompView)

multcompLetters(PT$Adjusted,
                compare="<",
                threshold=0.05,
                Letters=letters)

     Pooh Tigger Piglet
      "a"    "a"    "b"
```

Effect size measurements

A simple effect size measurement for Mood's median test is is to compare the medians of the groups.

In addition, the whole 5-number summary coule be used, including the minimum, 1st quartile, median, 3rd quartile, and the maximum.

```
library(FSA)

Summarize(Likert ~ Speaker, data=Data)

     Speaker  n mean        sd min Q1 median   Q3 max
   1  Piglet 10  2.3 0.8232726   1  2      2 2.75   4
   2    Pooh 10  4.2 0.6324555   3  4      4 4.75   5
   3  Tigger 10  4.0 0.6666667   3  4      4 4.00   5
```

Examining the medians and confidence intervals would be a somewhat different approach. Here, be cautious that confidence intervals by bootstrap may not be appropriate for the median for ordinal data with may ties, such as with Likert item data, or with small samples.

```
library(rcompanion)

groupwiseMedian(Likert ~ Speaker, data=Data, bca=FALSE, perc=TRUE)

     Speaker  n Median Conf.level Percentile.lower Percentile.upper
   1  Piglet 10      2       0.95              2.0              3.0
   2    Pooh 10      4       0.95              4.0              5.0
   3  Tigger 10      4       0.95              3.5              4.5

      ### Note that confidence intervals by bootstrap may vary.
```

In addition, looking at a statistic of stochastic dominance, like Vargha and Delaney's *A*, may be useful in this case.

```
library(rcompanion)

vda(x = Data$Likert[Data$Speaker=="Piglet"],
    y = Data$Likert[Data$Speaker=="Pooh"],
    verbose=TRUE)

   Statistic Value
   1 Proportion Ya > Yb  0.01
   2 Proportion Ya < Yb  0.91
   3    Proportion ties  0.08

   VDA
   0.05


vda(x = Data$Likert[Data$Speaker=="Piglet"],
    y = Data$Likert[Data$Speaker=="Tigger"],
    verbose=TRUE)

   Statistic Value
   1 Proportion Ya > Yb  0.02
```

```
  2 Proportion Ya < Yb  0.88
  3    Proportion ties  0.10

  VDA
  0.07
```

```
vda(x = Data$Likert[Data$Speaker=="Pooh"],
    y = Data$Likert[Data$Speaker=="Tigger"],
    verbose=TRUE)

  Statistic Value
  1 Proportion Ya > Yb  0.36
  2 Proportion Ya < Yb  0.20
  3    Proportion ties  0.44

  VDA
  0.58
```

Finally, we can divide the difference in medians from two groups by their pooled median absolute deviation (*mad*), which I've termed *Mangiafico's d*. It's somewhat analogous to a nonparametric version of Cohen's *d*. Note that this statistic assumes the data are at least interval in nature, as so may not be appropriate for Likert item data.

Here, the largest statistic is for the difference between groups A and C, where *d* = –4.55.

```
A     = c(1,2,2,2,2,3,4,5)
B     = c(2,3,4,4,4,5,5,5)
C     = c(3,4,4,4,5,5,5,5)
Y     = c(A, B, C)
Group = c(rep("A", length(A)), rep("B", length(B)), rep("C", length(C)))
Data3 = data.frame(Group, Y)


library(rcompanion)

multiMangiaficoD(Y ~ Group, data=Data3)

    $pairs
     Comparison Median.1 Median.2 MAD.1 MAD.2 Difference Pooled.MAD Mangiafico.d
  1      A - B        2      4.0 0.741 1.480       -2.0      1.170       -1.460
  2      A - C        2      4.5 0.741 0.741       -2.5      0.741       -4.550
  3      B - C        4      4.5 1.480 0.741       -0.5      1.170       -0.364

  $comparison
  Comparison
     "A - C"

  $statistic.m
  Maximum(abs(d))
             4.55
```

Manual calculation

For this example, the original *Data$Likert* is used. But the test is modified to use "*greater than or equal to mu*" rather than "*less than or equal to mu*".

```
MU  = median(Data$Likert)

A1  = sum(Data$Likert[Data$Speaker=="Pooh"]   >= MU)
B1  = sum(Data$Likert[Data$Speaker=="Piglet"] >= MU)
C1  = sum(Data$Likert[Data$Speaker=="Tigger"] >= MU)
A2  = sum(Data$Likert[Data$Speaker=="Pooh"]   <  MU)
B2  = sum(Data$Likert[Data$Speaker=="Piglet"] <  MU)
C2  = sum(Data$Likert[Data$Speaker=="Tigger"] <  MU)

Matrix = matrix(c(A1, B1, C1, A2, B2, C2), byrow=FALSE, ncol=2)

rownames(Matrix) = c("Pooh", "Piglet", "Tigger")

colnames(Matrix) = c("GreaterThanEqualMu", "LessThanMu")

Matrix

          GreaterThanEqualMu LessThanMu
   Pooh                    9          1
   Piglet                  1          9
   Tigger                  8          2
```

Monte Carlo simulation

```
chisq.test(Matrix, simulate.p.value=TRUE, B=10000)

   Pearson's Chi-squared test with simulated p-value (based on 10000 replicates)

   X-squared = 15.833, df = NA, p-value = 0.0005999
```

Post-hoc test

```
library(rcompanion)

pairwiseNominalIndependence(Matrix, simulate.p.value=TRUE, B=10000,
                            fisher=FALSE, gtest=FALSE)

          Comparison p.Chisq p.adj.Chisq
   1   Pooh : Piglet  0.0014     0.00420
   2   Pooh : Tigger  1.0000     1.00000
   3 Piglet : Tigger  0.0051     0.00765
```

Friedman Test

The Friedman test determines if there are differences among groups for two-way data structured in a specific way, namely in an *unreplicated complete block design*. In this design, one variable serves as the *treatment* or *group* variable, and another variable serves as the *blocking* variable. It is the differences among treatments or groups that we are interested in. We aren't necessarily interested in differences among blocks, but we want our statistics to take into account differences in the blocks. In the unreplicated complete block design, each block has one and only one observation of each treatment.

For an example of this structure, look at the Belcher family data below. *Rater* is considered the blocking variable, and each rater has *one observation* for each *Instructor*. The test will determine if there are differences among values for *Instructor*, taking into account any consistent effect of a *Rater*. For example, if *Rater a* rated consistently low and *Rater g* rated consistently high, the Friedman test can account for this statistically.

In other cases, the blocking variable might be the class where the ratings were done or the school where the ratings were done. If you were testing differences among curricula or other teaching treatments with different instructors, different instructors might be used as blocks.

Some people critique the Friedman test for having low power in detecting differences among groups. It has been suggested, however, that Friedman test may be powerful when there are five or more groups.

In general, you may want to choose a more powerful test. For an ordinal dependent variable, ordinal regression can be used, with the blocking variable being used as a random variable in the model. For a continuous dependent variable, the Quade test is an option, or aligned ranks transformation anova (ART anova) could be used, with the blocking variable being used as a random variable in the model.

Post-hoc tests

The outcome of the Friedman test tells you if there are differences among the groups, but doesn't tell you *which* groups are different from other groups. In order to determine which groups are different from others, post-hoc testing can be conducted. Several are presented here.

Appropriate data

- Two-way data arranged in an unreplicated complete block design
- Dependent variable is ordinal, interval, or ratio
- Treatment or group independent variable is a factor with two or more levels. That is, two or more groups
- Blocking variable is a factor with two or more levels
- Blocks are independent of each other and have no interaction with treatments

Hypotheses

- Null hypothesis: The distributions of values for each group are equal.
- Alternative hypothesis (two-sided): There is systematic difference in the distribution of values for the groups.

Interpretation

Significant results can be reported as "There was a significant difference in values among groups."

Other notes and alternative tests

The Quade test is used for the same kinds of data and hypotheses, but can be more powerful in some cases. It has been suggested that Friedman test may be preferable when there are a larger number of groups (five or more), while the Quade is preferable for fewer groups. The Quade test is described in the next chapter.

Cumulative link models for ordinal data (ordinal regression) are appropriate when the dependent variable is ordinal. Otherwise, aligned ranks transformation anova may be appropriate. Either of these approaches allows for more flexibility in design than the Friedman or Quade tests.

If the unreplicated block design is partially incomplete, the Skillings–Mack test can be used.

Packages used in this chapter

The packages used in this chapter include:

- psych
- FSA
- lattice
- coin
- PMCMRplus
- rcompanion
- DescTools

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(lattice)){install.packages("lattice")}
if(!require(coin)){install.packages("coin")}
if(!require(PMCMRplus)){install.packages("PMCMRplus")}
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(DescTools)){install.packages("DescTools")}
```

Friedman test example

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Instructor        Rater  Likert
 'Bob Belcher'        a      4
 'Bob Belcher'        b      5
 'Bob Belcher'        c      4
 'Bob Belcher'        d      6
 'Bob Belcher'        e      6
 'Bob Belcher'        f      6
 'Bob Belcher'        g     10
 'Bob Belcher'        h      6
 'Linda Belcher'      a      8
 'Linda Belcher'      b      6
 'Linda Belcher'      c      8
 'Linda Belcher'      d      8
 'Linda Belcher'      e      8
 'Linda Belcher'      f      7
 'Linda Belcher'      g     10
 'Linda Belcher'      h      9
 'Tina Belcher'       a      7
 'Tina Belcher'       b      5
 'Tina Belcher'       c      7
 'Tina Belcher'       d      8
 'Tina Belcher'       e      8
 'Tina Belcher'       f      9
 'Tina Belcher'       g     10
 'Tina Belcher'       h      9
 'Gene Belcher'       a      6
 'Gene Belcher'       b      4
 'Gene Belcher'       c      5
 'Gene Belcher'       d      5
 'Gene Belcher'       e      6
 'Gene Belcher'       f      6
 'Gene Belcher'       g      5
 'Gene Belcher'       h      5
 'Louise Belcher'     a      8
 'Louise Belcher'     b      7
 'Louise Belcher'     c      8
 'Louise Belcher'     d      8
 'Louise Belcher'     e      9
 'Louise Belcher'     f      9
 'Louise Belcher'     g      8
 'Louise Belcher'     h     10
")

### Order levels of the factor; otherwise R will alphabetize them

Data$Instructor = factor(Data$Instructor,
                         levels=unique(Data$Instructor))

### Create a new variable which is the likert scores as an ordered factor

Data$Likert.f = factor(Data$Likert,
                           ordered=TRUE)


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

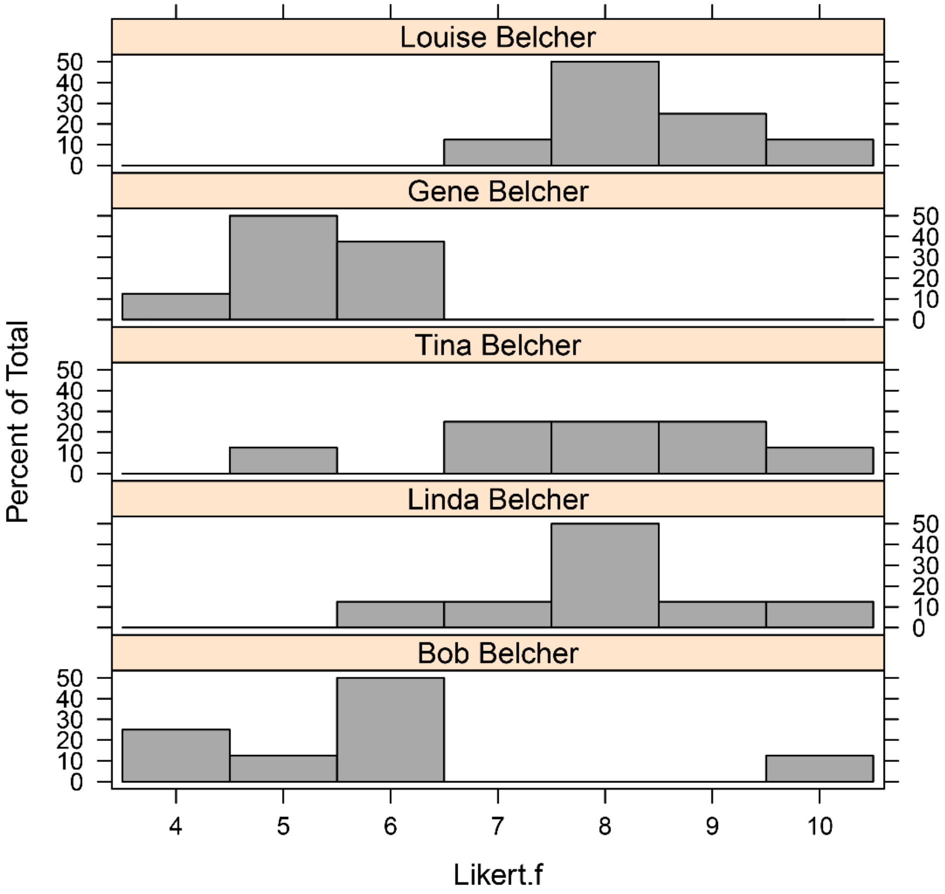
Summarize data treating Likert scores as factors

```
xtabs( ~ Instructor + Likert.f,
      data = Data)

                   Likert.f
   Instructor       4 5 6 7 8 9 10
     Bob Belcher    2 1 4 0 0 0  1
     Linda Belcher  0 0 1 1 4 1  1
     Tina Belcher   0 1 0 2 2 2  1
     Gene Belcher   1 4 3 0 0 0  0
     Louise Belcher 0 0 0 1 4 2  1


XT = xtabs( ~ Instructor + Likert.f,
           data = Data)

prop.table(XT,
           margin = 1)

                   Likert.f
   Instructor           4     5     6     7     8     9    10
     Bob Belcher    0.250 0.125 0.500 0.000 0.000 0.000 0.125
     Linda Belcher  0.000 0.000 0.125 0.125 0.500 0.125 0.125
     Tina Belcher   0.000 0.125 0.000 0.250 0.250 0.250 0.125
     Gene Belcher   0.125 0.500 0.375 0.000 0.000 0.000 0.000
     Louise Belcher 0.000 0.000 0.000 0.125 0.500 0.250 0.125
```

Bar plots by group

Note that the bar plots don’t show the effect of the blocking variable.

```
library(lattice)

histogram(~ Likert.f | Instructor,
          data=Data,
          layout=c(1,5),
          col="darkgray")

   ###  (1,5) indicates the columns and rows for the plots
```



Summarize data treating Likert scores as numeric

```
   library(FSA)

   Summarize(Likert ~ Instructor,
             data=Data,
             digits=3)
```

```
            Instructor n  mean    sd min   Q1 median   Q3 max percZero
      1    Bob Belcher 8 5.875 1.885   4 4.75      6 6.00  10        0
      2  Linda Belcher 8 8.000 1.195   6 7.75      8 8.25  10        0
      3   Tina Belcher 8 7.875 1.553   5 7.00      8 9.00  10        0
      4   Gene Belcher 8 5.250 0.707   4 5.00      5 6.00   6        0
      5 Louise Belcher 8 8.375 0.916   7 8.00      8 9.00  10        0
```

Friedman test example

This example uses the formula notation indicating that *Likert* is the dependent variable, *Instructor* is the independent variable, and *Rater* is the blocking variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?friedman.test* or documentation for other employed functions.

```
friedman.test(Likert ~ Instructor | Rater,
              data = Data)

   Friedman rank sum test

   Friedman chi-squared = 23.139, df = 4, p-value = 0.0001188


library(coin)

friedman_test(Likert ~ Instructor | Rater,
              data = Data)

   Asymptotic Friedman Test

   chi-squared = 23.139, df = 4, p-value = 0.0001188


library(PMCMRplus)

friedmanTest(y      = Data$Likert,
             groups = Data$Instructor,
             blocks = Data$Rater)

   Friedman rank sum test

   Friedman chi-squared = 23.139, df = 4, p-value = 0.0001188
```

Effect size

Kendall's *W*, or Kendall's coefficient of concordance, can be used as an effect size statistic for Friedman's test.

The following interpretations are based on personal intuition. They are not intended to be universal.

		small	medium	large
Kendall's *W*	**k** = 3	< 0.10	0.10 – < 0.30	≥ 0.30
	k = 5	< 0.10	0.10 – < 0.25	≥ 0.25
	k = 7	< 0.10	0.10 – < 0.20	≥ 0.20
	k = 9	< 0.10	0.10 – < 0.20	≥ 0.20

```
XT = xtabs(Likert ~ Instructor + Rater,
           data = Data)

XT

   Instructor        a  b  c  d  e  f  g  h
```

```
     Bob Belcher     4  5  4  6  6  6 10  6
     Linda Belcher   8  6  8  8  8  7 10  9
     Tina Belcher    7  5  7  8  8  9 10  9
     Gene Belcher    6  4  5  5  6  6  5  5
     Louise Belcher  8  7  8  8  9  9  8 10
```

For the *KendallW* function, groups must be in rows, and raters must be in columns.

```
library(DescTools)

KendallW(XT,
         correct=TRUE,
         test=TRUE)

   Kendall's coefficient of concordance Wt

   Kendall chi-squared = 23.139, df = 4, subjects = 5, raters = 8,
   p-value = 0.0001188

   sample estimates:
          Wt
   0.7230903
```

In the output above, check that the correct number of groups and raters is listed under "subjects" and "raters", respectively.

```
library(rcompanion)

kendallW(XT, correct=TRUE)

       W
   0.723


kendallW(XT, correct=TRUE, ci=TRUE)

         W lower.ci upper.ci
   1 0.723    0.547    0.917

      ###  Confidence intervals by bootstrap may vary
```

Post-hoc tests

Conover test

```
### Order groups by median

Data$Instructor = factor(Data$Instructor,
                   levels = c("Linda Belcher", "Louise Belcher",
                              "Tina Belcher", "Bob Belcher",
                              "Gene Belcher"))
```

```
   library(PMCMRplus)

   CT = frdAllPairsConoverTest(y      = Data$Likert,
                               groups = Data$Instructor,
                               blocks = Data$Rater,
                               p.adjust.method="single-step")


   CT

      Pairwise comparisons using Conover's all-pairs test for a two-way balanced
      complete block design

                     Linda Belcher Louise Belcher Tina Belcher Bob Belcher
      Louise Belcher 0.9794        -              -            -
      Tina Belcher   0.9884        0.8278         -            -
      Bob Belcher    0.0853        0.0169         0.2490       -
      Gene Belcher   0.0099        0.0012         0.0447       0.9489

      P value adjustment method: single-step


library(rcompanion)

CTT =PMCMRTable(CT)

CTT

                                  Comparison p.value
      1  Louise Belcher - Linda Belcher = 0   0.979
      2    Tina Belcher - Linda Belcher = 0   0.988
      3     Bob Belcher - Linda Belcher = 0  0.0853
      4    Gene Belcher - Linda Belcher = 0 0.00993
      5   Tina Belcher - Louise Belcher = 0   0.828
      6    Bob Belcher - Louise Belcher = 0  0.0169
      7   Gene Belcher - Louise Belcher = 0 0.00123
      8      Bob Belcher - Tina Belcher = 0   0.249
      9     Gene Belcher - Tina Belcher = 0  0.0447
      10     Gene Belcher - Bob Belcher = 0   0.949


   library(rcompanion)

   cldList(p.value ~ Comparison, data = CTT)

                Group Letter MonoLetter
      1 LouiseBelcher      a         a
      2   TinaBelcher     ab        ab
      3    BobBelcher     bc         bc
      4   GeneBelcher      c          c
      5  LindaBelcher     ab        ab
```

Exact test

```
library(PMCMRplus)

ET = frdAllPairsExactTest(y      = Data$Likert,
                          groups = Data$Instructor,
                          blocks = Data$Rater,
                          p.adjust.method="fdr")


ET

   Pairwise comparisons using Eisinga, Heskes, Pelzer & Te Grotenhuis all-pairs test
   with exact p-values for a two-way balanced complete block design

   data: y, groups and blocks

                  Linda Belcher Louise Belcher Tina Belcher Bob Belcher
   Louise Belcher 0.65081       -              -            -          
   Tina Belcher   0.69729       0.44188        -            -          
   Bob Belcher    0.02456       0.00768        0.07761      -          
   Gene Belcher   0.00601       0.00047        0.01833      0.60368    

   P value adjustment method: fdr


library(rcompanion)

ETT =PMCMRTable(ET)

ETT

                                Comparison  p.value
   1  Louise Belcher - Linda Belcher = 0    0.651
   2    Tina Belcher - Linda Belcher = 0    0.697
   3     Bob Belcher - Linda Belcher = 0   0.0246
   4    Gene Belcher - Linda Belcher = 0  0.00601
   5   Tina Belcher - Louise Belcher = 0    0.442
   6    Bob Belcher - Louise Belcher = 0  0.00768
   7   Gene Belcher - Louise Belcher = 0 0.000467
   8      Bob Belcher - Tina Belcher = 0   0.0776
   9     Gene Belcher - Tina Belcher = 0   0.0183
   10     Gene Belcher - Bob Belcher = 0    0.604


library(rcompanion)

cldList(p.value ~ Comparison, data = ETT)

             Group Letter MonoLetter
   1 LouiseBelcher      a        a  
   2   TinaBelcher     ab        ab 
   3    BobBelcher     bc         bc
   4   GeneBelcher      c          c
   5  LindaBelcher      a        a  
```

Nemenyi test

```
library(PMCMRplus)

NT = frdAllPairsNemenyiTest(Likert ~ Instructor | Rater, data = Data)


NT

   Pairwise comparisons using Nemenyi-Wilcoxon-Wilcox all-pairs test for a two-way
   balanced complete block design

                  Linda Belcher Louise Belcher Tina Belcher Bob Belcher
   Louise Belcher 0.9816        -              -            -
   Tina Belcher   0.9897        0.8426         -            -
   Bob Belcher    0.1021        0.0224         0.2775       -
   Gene Belcher   0.0136        0.0019         0.0557       0.9540

   P value adjustment method: single-step
```

```
library(rcompanion)

NTT =PMCMRTable(NT)

NTT

                              Comparison p.value
   1  Louise Belcher - Linda Belcher = 0    0.982
   2    Tina Belcher - Linda Belcher = 0     0.99
   3     Bob Belcher - Linda Belcher = 0    0.102
   4    Gene Belcher - Linda Belcher = 0   0.0136
   5   Tina Belcher - Louise Belcher = 0    0.843
   6    Bob Belcher - Louise Belcher = 0   0.0224
   7   Gene Belcher - Louise Belcher = 0 0.00189
   8      Bob Belcher - Tina Belcher = 0    0.278
   9     Gene Belcher - Tina Belcher = 0   0.0557
   10     Gene Belcher - Bob Belcher = 0    0.954
```

```
library(rcompanion)

cldList(p.value ~ Comparison, data = NTT)

             Group Letter MonoLetter
   1 LouiseBelcher      a        a
   2   TinaBelcher    abc        abc
   3    BobBelcher     bc         bc
   4   GeneBelcher      b         b
   5  LindaBelcher     ac        a c
```

Siegel test

```
library(PMCMRplus)

ST = frdAllPairsSiegelTest(y      = Data$Likert,
                           groups = Data$Instructor,
                            blocks = Data$Rater,
                          p.adjust.method="fdr")


ST
```

```
   Pairwise comparisons using Siegel-Castellan all-pairs test for a two-way balanced
   complete block design

                  Linda Belcher Louise Belcher Tina Belcher Bob Belcher
   Louise Belcher 0.6353        -              -            -
   Tina Belcher   0.6353        0.4344         -            -
   Bob Belcher    0.0285        0.0089         0.0802       -
   Gene Belcher   0.0078        0.0020         0.0180       0.5960

   P value adjustment method: fdr
```

```
library(rcompanion)

STT =PMCMRTable(ST)

STT
```

```
                              Comparison p.value
   1  Louise Belcher - Linda Belcher = 0   0.635
   2    Tina Belcher - Linda Belcher = 0   0.635
   3     Bob Belcher - Linda Belcher = 0  0.0285
   4    Gene Belcher - Linda Belcher = 0 0.00783
   5   Tina Belcher - Louise Belcher = 0   0.434
   6    Bob Belcher - Louise Belcher = 0 0.00888
   7   Gene Belcher - Louise Belcher = 0 0.00203
   8      Bob Belcher - Tina Belcher = 0  0.0802
   9     Gene Belcher - Tina Belcher = 0   0.018
   10     Gene Belcher - Bob Belcher = 0   0.596
```

```
library(rcompanion)

cldList(p.value ~ Comparison, data = STT)
```

```
             Group Letter MonoLetter
   1 LouiseBelcher      a        a
   2   TinaBelcher     ab        ab
   3    BobBelcher     bc         bc
   4   GeneBelcher      c          c
   5  LindaBelcher      a        a
```

Example from Conover

This example is taken from the Friedman test section of Conover (1999).

```
Conover1 = read.table(header=TRUE, stringsAsFactors=TRUE, text="

Homeowner Grass1 Grass2 Grass3 Grass4
 1        4      3      2      1
 2        4      2      3      1
 3        3      1.5    1.5    4
 4        3      1      2      4
 5        4      2      1      3
 6        2      2      2      4
 7        1      3      2      4
 8        2      4      1      3
 9        3.5    1      2      3.5
10        4      1      3      2
11        4      2      3      1
12        3.5    1      2      3.5
")

if(!require(tidyr)){install.packages("tidyr")}

library(tidyr)

Conover = gather(Conover1, Grass, Rating, Grass1:Grass4, factor_key=TRUE)


###  Check the data frame

library(psych)

headTail(Conover)

str(Conover)

summary(Conover)


###  Friedman test

friedman.test(Rating ~ Grass | Homeowner,
              data = Conover)

   Friedman rank sum test

   Friedman chi-squared = 8.0973, df = 3, p-value = 0.04404



GT = xtabs(Rating ~ Grass + Homeowner,
           data = Conover)

GT
```

```
          Homeowner
Grass      1   2   3   4   5   6   7   8   9   10  11  12
  Grass1 4.0 4.0 3.0 3.0 4.0 2.0 1.0 2.0 3.5 4.0 4.0 3.5
  Grass2 3.0 2.0 1.5 1.0 2.0 2.0 3.0 4.0 1.0 1.0 2.0 1.0
  Grass3 2.0 3.0 1.5 2.0 1.0 2.0 2.0 1.0 2.0 3.0 3.0 2.0
  Grass4 1.0 1.0 4.0 4.0 3.0 4.0 4.0 3.0 3.5 2.0 1.0 3.5
```

```
library(DescTools)

KendallW(GT, correct=TRUE, test=TRUE)

   Kendall's coefficient of concordance Wt

   Kendall chi-squared = 8.0973, df = 3, subjects = 4, raters = 12, p-value =
   0.04404

   sample estimates:
          Wt
   0.2249263


library(PMCMRplus)

frdAllPairsExactTest(y      = Conover$Rating,
                     groups = Conover$Grass,
                     blocks = Conover$Homeowner,
                     p.adjust.method = "fdr")

   Pairwise comparisons using Eisinga, Heskes, Pelzer & Te Grotenhuis all-pairs test
   with exact p-values for a two-way balanced complete block design

          Grass1 Grass2 Grass3
   Grass2 0.094  -      -
   Grass3 0.094  0.938  -
   Grass4 0.701  0.194  0.201

   P value adjustment method: fdr
```

References

Conover, W.J. 1999. Practical Nonparametric Statistics, 3rd. John Wiley & Sons.

Quade Test

The Quade test is used for similar data and hypotheses as the Friedman test, namely for unreplicated complete block designs. Because the Quade test uses subtraction to determine the ranges of values within blocks, it may not be appropriate for strictly ordinal data.

Post-hoc tests
The outcome of the Quade test tells you if there are differences among the groups, but doesn't tell you *which* groups are different from other groups. In order to determine which groups are different from others, post-hoc testing can be conducted.

Appropriate data

- Two-way data arranged in an unreplicated complete block design
- Dependent variable is interval, or ratio.
- Treatment or group independent variable is a factor with two or more levels. That is, two or more groups
- Blocking variable is a factor with two or more levels
- Blocks are independent of each other and have no interaction with treatments

Hypotheses

- Null hypothesis: The distributions of values for each group are equal.
- Alternative hypothesis (two-sided): There is systematic difference in the distribution of values for the groups.

Interpretation

Significant results can be reported as "There was a significant difference in values among groups."

Other notes and alternative tests
The Friedman test is used for the same kinds of data and hypotheses. The Friedman test is described in the previous chapter. Cumulative link models or aligned ranks transformation anova allow for more flexibility in design.

Packages used in this chapter

The packages used in this chapter include:

- psych
- FSA
- lattice
- PMCMRplus
- rcompanion

The following commands will install these packages if they are not already installed:

```
if(!require(psych)){install.packages("psych")}
if(!require(FSA)){install.packages("FSA")}
if(!require(lattice)){install.packages("lattice")}
if(!require(PMCMRplus)){install.packages("PMCMRplus")}
if(!require(rcompanion)){install.packages("rcompanion")}
```

Quade test example

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Instructor        Rater  Likert
 'Bob Belcher'        a      4
 'Bob Belcher'        b      5
 'Bob Belcher'        c      4
 'Bob Belcher'        d      6
 'Bob Belcher'        e      6
 'Bob Belcher'        f      6
 'Bob Belcher'        g     10
 'Bob Belcher'        h      6
 'Linda Belcher'      a      8
 'Linda Belcher'      b      6
 'Linda Belcher'      c      8
 'Linda Belcher'      d      8
 'Linda Belcher'      e      8
 'Linda Belcher'      f      7
 'Linda Belcher'      g     10
 'Linda Belcher'      h      9
 'Tina Belcher'       a      7
 'Tina Belcher'       b      5
 'Tina Belcher'       c      7
 'Tina Belcher'       d      8
 'Tina Belcher'       e      8
 'Tina Belcher'       f      9
 'Tina Belcher'       g     10
 'Tina Belcher'       h      9
 'Gene Belcher'       a      6
 'Gene Belcher'       b      4
 'Gene Belcher'       c      5
 'Gene Belcher'       d      5
 'Gene Belcher'       e      6
 'Gene Belcher'       f      6
 'Gene Belcher'       g      5
 'Gene Belcher'       h      5
 'Louise Belcher'     a      8
 'Louise Belcher'     b      7
 'Louise Belcher'     c      8
 'Louise Belcher'     d      8
 'Louise Belcher'     e      9
 'Louise Belcher'     f      9
 'Louise Belcher'     g      8
 'Louise Belcher'     h     10
")


### Order levels of the factor; otherwise R will alphabetize them

Data$Instructor = factor(Data$Instructor,
                      levels=unique(Data$Instructor))
```

```
### Create a new variable which is the likert scores as an ordered factor

Data$Likert.f = factor(Data$Likert,
                          ordered=TRUE)


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)
```

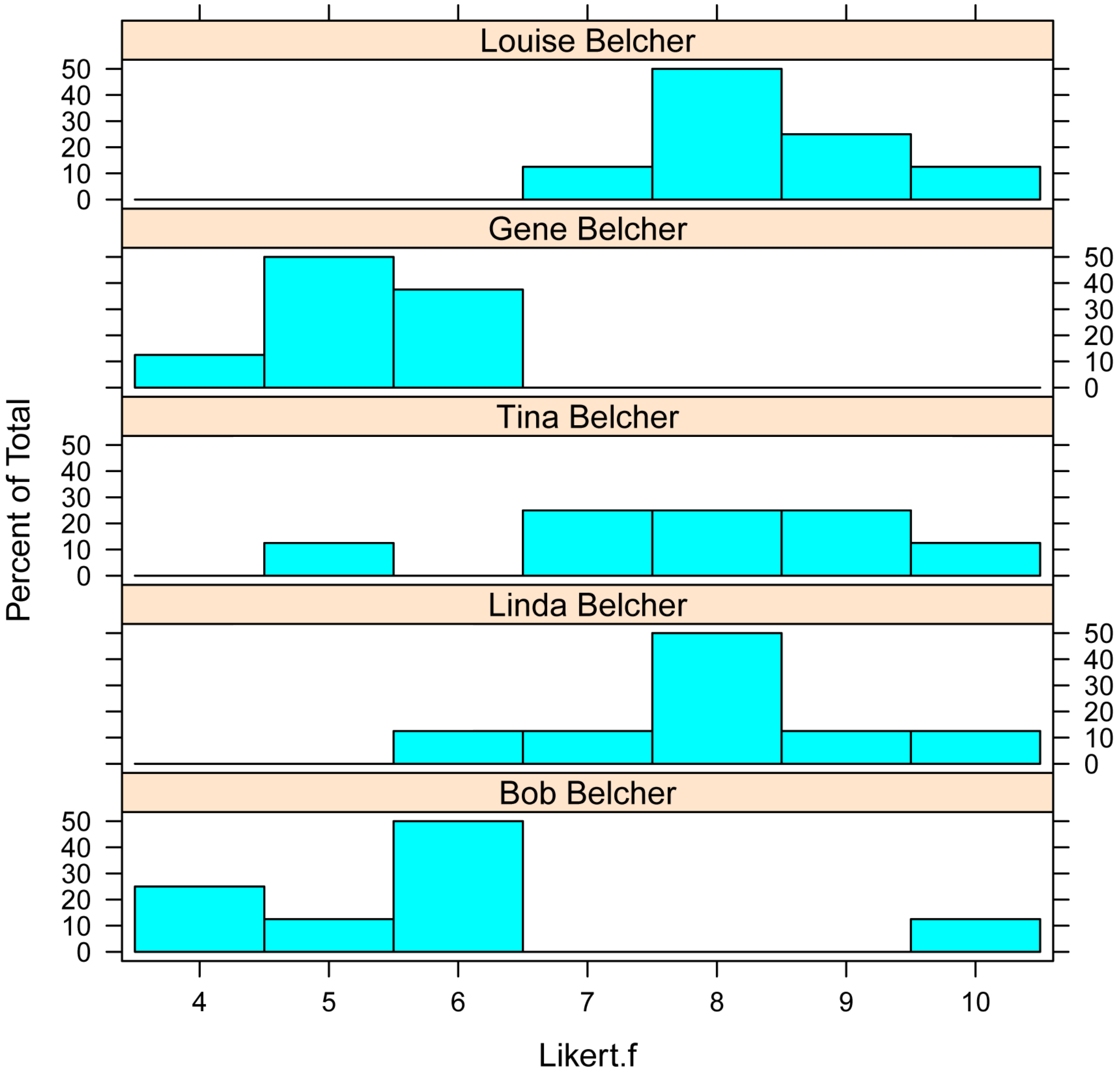
Summarize data treating Likert scores as factors

```
xtabs( ~ Instructor + Likert.f,
      data = Data)

                Likert.f
   Instructor       4 5 6 7 8 9 10
     Bob Belcher    2 1 4 0 0 0  1
     Linda Belcher  0 0 1 1 4 1  1
     Tina Belcher   0 1 0 2 2 2  1
     Gene Belcher   1 4 3 0 0 0  0
     Louise Belcher 0 0 0 1 4 2  1


XT = xtabs( ~ Instructor + Likert.f,
           data = Data)

prop.table(XT,
           margin = 1)

                Likert.f
   Instructor           4     5     6     7     8     9    10
     Bob Belcher    0.250 0.125 0.500 0.000 0.000 0.000 0.125
     Linda Belcher  0.000 0.000 0.125 0.125 0.500 0.125 0.125
     Tina Belcher   0.000 0.125 0.000 0.250 0.250 0.250 0.125
     Gene Belcher   0.125 0.500 0.375 0.000 0.000 0.000 0.000
     Louise Belcher 0.000 0.000 0.000 0.125 0.500 0.250 0.125
```

Bar plots by group

Note that the bar plots don’t show the effect of the blocking variable.

```
library(lattice)

histogram(~ Likert.f | Instructor,
          data=Data,
          layout=c(1,5)      #  columns and rows of individual plots
          )
```



Summarize data treating Likert scores as numeric

```
library(FSA)

Summarize(Likert ~ Instructor,
          data=Data,
          digits=3)
```

```
         Instructor n  mean    sd min   Q1 median   Q3 max percZero
   1     Bob Belcher 8 5.875 1.885   4 4.75      6 6.00  10        0
   2   Linda Belcher 8 8.000 1.195   6 7.75      8 8.25  10        0
   3    Tina Belcher 8 7.875 1.553   5 7.00      8 9.00  10        0
   4    Gene Belcher 8 5.250 0.707   4 5.00      5 6.00   6        0
   5  Louise Belcher 8 8.375 0.916   7 8.00      8 9.00  10        0
```

Quade test example

This example uses the formula notation indicating that *Likert* is the dependent variable, *Instructor* is the independent variable, and *Rater* is the blocking variable. The *data=* option indicates the data frame that contains the variables. For the meaning of other options, see *?quade.test*.

```
quade.test(Likert ~ Instructor | Rater,
           data = Data)

   Quade test

   Quade F = 8.0253, num df = 4, denom df = 28, p-value = 0.0001924
```

Post-hoc tests

```
### Order groups by median

Data$Instructor = factor(Data$Instructor,
                   levels = c("Linda Belcher", "Louise Belcher",
                              "Tina Belcher", "Bob Belcher",
                              "Gene Belcher"))


library(PMCMRplus)

QT = quadeAllPairsTest(y      = Data$Likert,
                       groups = Data$Instructor,
                       blocks = Data$Rater,
                       p.adjust.method = "fdr")


QT

   Pairwise comparisons using Quade's test with TDist approximation

                  Linda Belcher Louise Belcher Tina Belcher Bob Belcher
   Louise Belcher 0.77620       -              -            -
   Tina Belcher   0.50547       0.38132        -            -
   Bob Belcher    0.00902       0.00544        0.04533      -
   Gene Belcher   0.00108       0.00099        0.00539      0.36572

   P value adjustment method: fdr
```

```
library(rcompanion)

QTT =PMCMRTable(QT)

QTT

                                 Comparison  p.value
```

```
   1  Louise Belcher - Linda Belcher = 0    0.776
   2    Tina Belcher - Linda Belcher = 0    0.505
   3     Bob Belcher - Linda Belcher = 0  0.00902
   4    Gene Belcher - Linda Belcher = 0  0.00108
   5   Tina Belcher - Louise Belcher = 0    0.381
   6    Bob Belcher - Louise Belcher = 0  0.00544
   7   Gene Belcher - Louise Belcher = 0 0.000987
   8      Bob Belcher - Tina Belcher = 0   0.0453
   9     Gene Belcher - Tina Belcher = 0  0.00539
   10     Gene Belcher - Bob Belcher = 0    0.366


library(rcompanion)

cldList(p.value ~ Comparison, data = QTT)

              Group Letter MonoLetter
   1 LouiseBelcher      a          a
   2   TinaBelcher      a          a
   3    BobBelcher      b           b
   4   GeneBelcher      b           b
   5  LindaBelcher      a          a
```

Example from Conover

This example is taken from the Quade test section of Conover (1999).

```
Conover1 = read.table(header=TRUE, stringsAsFactors=TRUE, text="

Store    A     B     C     D     E
 1       5     4     7    10    12
 2       1     3     1     0     2
 3      16    12    22    22    35
 4       5     4     3     5     4
 5      10     9     7    13    10
 6      19    18    28    37    58
 7      10     7     6     8     7
")

if(!require(tidyr)){install.packages("tidyr")}

library(tidyr)

Conover = gather(Conover1, Brand, Bottles, A:E, factor_key=TRUE)


###  Check the data frame

library(psych)

headTail(Conover)

str(Conover)

summary(Conover)
```

```
###  Quade test

quade.test(Bottles ~ Brand | Store,
              data = Conover)

   Quade test

   Quade F = 3.8293, num df = 4, denom df = 24, p-value = 0.01519


library(PMCMRplus)

quadeAllPairsTest(y      = Conover$Bottles,
                  groups = Conover$Brand,
                  blocks = Conover$Store,
                  p.adjust.method = "fdr")

   Pairwise comparisons using Quade's test with TDist approximation

     A     B     C     D
   B 0.298 -     -     -
   C 0.840 0.359 -     -
   D 0.246 0.051 0.204 -
   E 0.104 0.021 0.090 0.575

   P value adjustment method: fdr
```

References

Conover, W.J. 1999. Practical Nonparametric Statistics, 3rd. John Wiley & Sons.

Scheirer–Ray–Hare Test

The Scheirer–Ray–Hare test is a nonparametric test used for a two-way factorial design. It appears to be not well documented, but it is discussed in Sokal and Rohlf (1995). In my experience, in some cases the Scheirer–Ray–Hare test is less likely to find the interaction effect significant than would an ordinary least squares analysis of variance, aligned ranks transformation ANOVA, normal scores ANOVA, or permutation tests. In some cases, these tests may be preferred alternatives.

It has been suggested that the observations should be balanced and that each cell in the interaction should have at least five observations.

Note that for unbalanced designs, the *scheirerRayHare* function uses a type-II sum-of-squares approach by default. There is an option to use type-I sum-of-squares.

Appropriate data

- Two-way data arranged in a factorial design
- Dependent variable is ordinal, interval, or ratio
- There are two treatment or group independent variables. Each is a factor with two or more levels
- Observations are independent. That is, they are not paired or repeated measures

Post-hoc tests

Appropriate post-hoc tests might be Dunn test for each significant factor or interaction

Packages used in this chapter

The packages used in this chapter include:

- rcompanion
- FSA

The following commands will install these packages if they are not already installed:

```
if(!require(rcompanion)){install.packages("rcompanion")}
if(!require(FSA)){install.packages("FSA")}
```

Scheirer–Ray–Hare test examples

Midichlorians example

```
### Assemble the data

Location = c(rep("Olympia" , 6), rep("Ventura", 6),
             rep("Northampton", 6), rep("Burlington", 6))

Tribe  = c(rep(c("Jedi", "Sith"), 12))

Midichlorians = c(10,  4, 12,  5, 15,  4, 15,  9, 15, 11, 18, 12,
                   8, 13,  8, 15, 10, 17, 22, 22, 20, 22, 20, 25)

Data = data.frame(Tribe, Location, Midichlorians)

str(Data)


### Scheirer–Ray–Hare test

library(rcompanion)

scheirerRayHare(Midichlorians ~ Tribe + Location,
                data = Data)

   DV:  Midichlorians
   Observations:  24
   D:  0.9917391
   MS total:   50

                  Df Sum Sq       H p.value
   Tribe           1   8.17  0.1647 0.68487
   Location        3 746.58 15.0560 0.00177
   Tribe:Location  3 315.58  6.3642 0.09517
   Residuals      16  70.17


### Post-hoc test

### Order groups by median

Data$Location = factor(Data$Location,
                       levels=c("Burlington", "Ventura", "Northampton", "Olympia"))

levels(Data$Location)


### Dunn test

library(FSA)

DT = dunnTest(Midichlorians ~ Location,
              data=Data,
              method="bh")      # Adjusts p-values for multiple comparisons;
                                # See ?dunnTest for options

DT


### Compact letter display

PT = DT$res

PT

library(rcompanion)

cldList(P.adj ~ Comparison,
        data = PT,
        threshold = 0.05)

           Group Letter MonoLetter
   1  Burlington      a          a
   2 Northampton      b           b
   3     Olympia      b           b
   4     Ventura      b           b

   Groups sharing a letter not signficantly different (alpha = 0.05).
```

Example from Sokal and Rohlf

```
### Assemble the data
```

```
Value = c(709,679,699,657,594,677,592,538,476,508,505,539)
Sex   = c(rep("Male",3), rep("Female",3), rep("Male",3), rep("Female",3))
Fat   = c(rep("Fresh", 6), rep("Rancid", 6))

Sokal = data.frame(Value, Sex, Fat)

str(Sokal)


### Scheirer–Ray–Hare test

library(rcompanion)

scheirerRayHare(Value ~ Sex + Fat,
                data=Sokal)

   DV:  Value
   Observations:  12
   D:  1
   MS total:  13

             Df  Sum Sq      H p.value
   Sex        1   8.333 0.6410 0.42334
   Fat        1 108.000 8.3077 0.00395
   Sex:Fat    1   5.333 0.4103 0.52184
   Residuals  8  21.333
```

Example from Real Statistics Using Excel

This example from www.real-statistics.com/two-way-anova/scheirer-ray-hare-test/.

```
### Assemble the data

Wheat = c(123,156,112,100,168,135,130,176,120,155,156,180,147,146,193)
Corn  = c(128,150,174,116,109,175,132,120,187,184,186,138,178,176,190)
Soy   = c(166,178,187,153,195,140,145,159,131,126,185,206,188,165,188)
Rice  = c(151,125,117,155,158,167,183,142,167,168,175,173,154,191,169)
Yield = c(Wheat, Corn, Soy, Rice)

Fert  = rep(c(rep("Blend X",5), rep("Blend Y",5), rep("Blend Z",5)),4)
Crop  = c(rep("Wheat",15), rep("Corn",15), rep("Soy",15), rep("Rice",15))

Real.stats = data.frame(Yield, Fert, Crop)

str(Real.stats)


### Scheirer–Ray–Hare test

library(rcompanion)
```

```
scheirerRayHare(Yield ~ Fert + Crop,
                data=Real.stats)

   DV:  Yield
   Observations:  60
   D:  0.9997221
   MS total:  305

             Df Sum Sq       H  p.value
   Fert       2 4004.4 13.1329 0.001407
   Crop       3 1339.0  4.3915 0.222175
   Fert:Crop  6 2893.6  9.4900 0.147839
   Residuals 48 9752.9


### Post-hoc test

### Order groups by median

Real.stats$Fert = factor(Real.stats$Fert,
                       levels=c("Blend Z", "Blend Y", "Blend X"))

levels(Real.stats$Fert)


### Dunn test

library(FSA)

DT = dunnTest(Yield ~ Fert,
              data=Real.stats,
              method="bh")      # Adjusts p-values for multiple comparisons;
                                # See ?dunnTest for options

DT


### Compact letter display

PT = DT$res

PT

library(rcompanion)

cldList(P.adj ~ Comparison,
        data = PT,
        threshold = 0.05)

      Group Letter MonoLetter
   1 BlendX      a          a
   2 BlendY      a          a
   3 BlendZ      b           b

   Groups sharing a letter not signficantly different (alpha = 0.05).
```

Example with unbalanced design

The following example is taken from Klasson in the *References* section. Note that because the design is not balanced, type-I sum-of-square results differ from those with type-II sum-of-squares.

```
### Assemble the data

Gambling = c(3.0, 2.8, 3.0, 5.1, 4.7, 4.9, 5.2, 4.9, 5.0, 2.1, 2.0, 1.9,
             1.8, 2.3, 2.1, 2.4, 3.9, 3.8, 4.1, 1.2, 1.1, 1.3, 1.1, 1.0)
Gender = c(rep("Male", 13), rep("Female", 11))
Athletic = c(rep("Current", 3), rep("Former", 6), rep("Non", 4),
             rep("Current", 3), rep("Former", 3), rep("Non", 5))

ARS = data.frame(Gender, Athletic, Gambling)


### Use type-II sum-of-squares

library(rcompanion)

scheirerRayHare(Gambling ~ Gender + Athletic, data=ARS)

   DV:  Gambling
   Observations:  24
   D:  0.9982609
   MS total:  50

                   Df Sum Sq       H p.value
   Gender           1 100.89  2.0214 0.15510
   Athletic         2 850.69 17.0434 0.00020
   Gender:Athletic  2   2.18  0.0437 0.97837
   Residuals       18  39.85


### Use type-I sum-of-squares

library(rcompanion)

scheirerRayHare(Gambling ~ Gender + Athletic, data=ARS, type=1)

   DV:  Gambling
   Observations:  24
   D:  0.9982609
   MS total:  50

                   Df Sum Sq       H p.value
   Gender           1 255.27  5.1143 0.02373
   Athletic         2 850.69 17.0434 0.00020
   Gender:Athletic  2   2.18  0.0437 0.97837
   Residuals       18  39.85
```

References

Klasson, K.T. 2020. *Two-way ANOVA for Unbalanced Data: The Spreadsheet Way*. USDA–ARS. www.ars.usda.gov/ARSUserFiles/60540520/Two-wayANOVAspreadsheet.pdf.

Sokal, R.R. and F.J. Rohlf. 1995. *Biometry,* 3rd ed. W.H. Freeman. New York.

Aligned Ranks Transformation ANOVA

Introduction

Aligned ranks transformation ANOVA (ART anova) is a nonparametric approach that allows for multiple independent variables, interactions, and repeated measures.

My understanding is that, since the aligning process requires subtracting values, the dependent variable needs to be interval in nature. That is, strictly ordinal data would be treated as numeric in the process.

The package *ARTool* makes using this approach in R relatively easy.

A few notes on using *ARTool*:

- All independent variables must be nominal

- All interactions of fixed independent variables need to be included in the model

- Post-hoc comparisons can be conducted

- For fixed-effects models, *eta*-squared can be calculated as an effect size

Packages used in this chapter

The packages used in this chapter include:

- ARTool
- emmeans
- multcomp
- rcompanion
- ggplot2
- psych

The following commands will install these packages if they are not already installed:

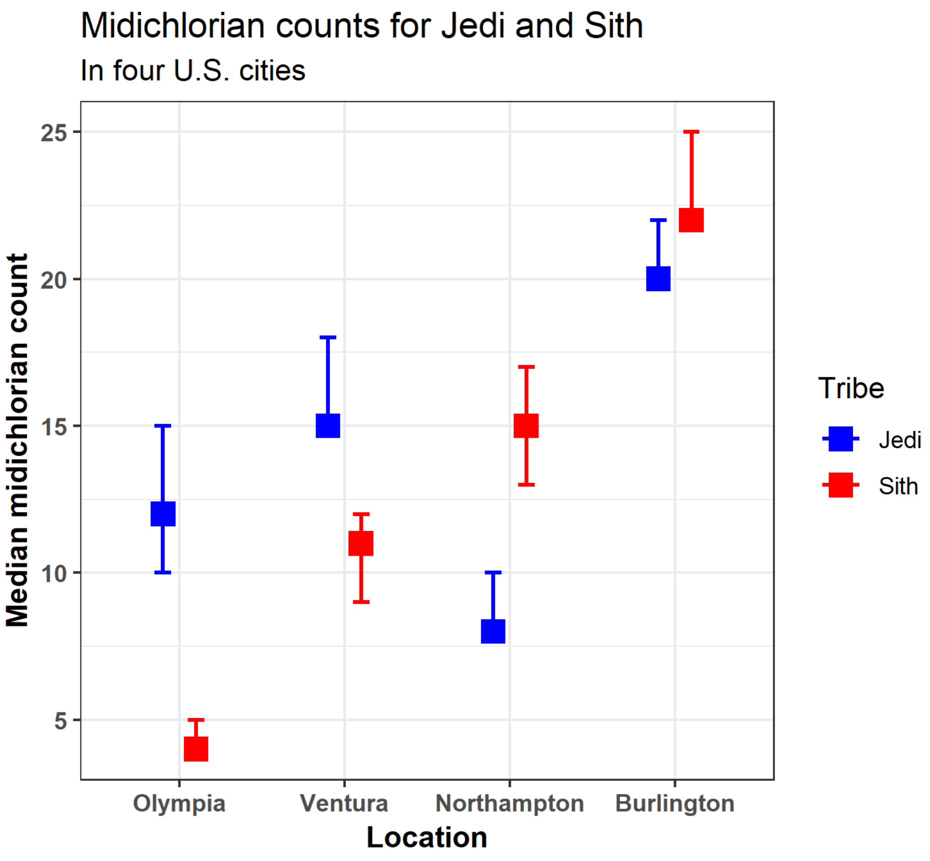
```
if(!require(ARTool)){install.packages("ARTool")}
if(!require(emmeans)){install.packages("emmeans")}
if(!require(multcomp)){install.packages("multcomp")}
if(!require(rcompanion)){install.packages("rcompanion ")}
if(!require(ggplot2)){install.packages("ggplot2")}
if(!require(psych)){install.packages("psych")}
```

Aligned Ranks Transformation ANOVA examples

Midichlorians example

This example reproduces the data used in the *Scheirer–Ray–Hare Test* chapter. Note that the aligned ranks anova finds a significant interaction, where the Scheirer–Ray–Hare test failed to detect this. Also note that the results are similar to those from a standard anova in the *Estimated Marginal Means for Multiple Comparisons* chapter.

The code for producing the plot is found at the end of the chapter.



Midichlorian counts for two tribes across four locations. Boxes indicate the median. Error bars indicate the 95% confidence interval of the median.

```
### Assemble the data

Location = as.factor(c(rep("Olympia" , 6), rep("Ventura", 6),
                rep("Northampton", 6), rep("Burlington", 6)))

Tribe  = as.factor(c(rep(c("Jedi", "Sith"), 12)))

Midichlorians = c(10,  4, 12,  5, 15,  4, 15,  9, 15, 11, 18, 12,
                   8, 13,  8, 15, 10, 17, 22, 22, 20, 22, 20, 25)

Data = data.frame(Tribe, Location, Midichlorians)

str(Data)


### Aligned ranks anova

library(ARTool)

model = art(Midichlorians ~ Tribe + Location + Tribe:Location,
                data = Data)

### Check the success of the procedure

model

   Aligned Rank Transform of Factorial Model

   Call:
   art(formula = Midichlorians ~ Tribe + Location + Tribe:Location,
       data = Data)

   Column sums of aligned responses (should all be ~0):
            Tribe       Location Tribe:Location
                0              0              0


### Conduct ANOVA

anova(model)

   Analysis of Variance of Aligned Rank Transformed Data

   Table Type: Anova Table (Type III tests)
   Model: No Repeated Measures (lm)
   Response: art(Midichlorians)

                    Df Df.res F value     Pr(>F)    
   1 Tribe           1     16  3.0606   0.099364   .
   2 Location        3     16 34.6201 3.1598e-07 ***
   3 Tribe:Location  3     16 29.9354 8.4929e-07 ***
```

Post-hoc comparisons for main effects

```
marginal = art.con(model, "Location")

marginal
```

```
 contrast                  estimate   SE df t.ratio p.value
  Burlington - Northampton    10.83 1.78 16   6.075  0.0001
  Burlington - Olympia        17.83 1.78 16  10.000  <.0001
  Burlington - Ventura         7.33 1.78 16   4.112  0.0041
  Northampton - Olympia        7.00 1.78 16   3.925  0.0060
  Northampton - Ventura       -3.50 1.78 16  -1.963  0.2426
  Olympia - Ventura          -10.50 1.78 16  -5.888  0.0001

 Results are averaged over the levels of: Tribe
 P value adjustment: tukey method for comparing a family of 4 estimates
```

```
Sum = as.data.frame(marginal)

library(rcompanion)

cldList(p.value ~ contrast, data=Sum)
```

```
         Group Letter MonoLetter
 1  Burlington      a         a
 2 Northampton      b          b
 3     Olympia      c           c
 4     Ventura      b          b
```

Post-hoc comparisons for interactions in a two-way model

Estimate values in the *emmeans* output should be ignored.

```
marginal = art.con(model, "Tribe:Location", adjust="none")

marginal

   ### For Tukey-adjusted p-values, use adjust="tukey"


   ### Here, results truncated to comparisons within each Location

   contrast                               estimate   SE df t.ratio p.value
    Jedi,Burlington - Sith,Burlington        -2.33 1.71 16  -1.365  0.1913
    Jedi,Northampton - Sith,Northampton      -9.00 1.71 16  -5.264  0.0001
    Jedi,Olympia - Sith,Olympia               8.83 1.71 16   5.166  0.0001
    Jedi,Ventura - Sith,Ventura               7.17 1.71 16   4.191  0.0007
```

Partial *eta*-squared

Partial *eta*-squared can be calculated as an effect size statistic for aligned ranks transformation anova.

Interpretation of eta*-squared*

Interpretation of effect sizes necessarily varies by discipline and the expectations of the experiment, but for behavioral studies, the guidelines proposed by Cohen (1988) are sometimes followed. They should not be considered universal.

	Small	Medium	Large
***eta*-squared**	0.01 – < 0.06	0.06 – < 0.14	≥ 0.14

Source: Cohen (1988).

```
Result = anova(model)

Result$part.eta.sq = with(Result, `Sum Sq`/(`Sum Sq` + `Sum Sq.res`))

Result

   Analysis of Variance of Aligned Rank Transformed Data

                    Df Df.res F value     Pr(>F) part.eta.sq.    
   1 Tribe           1     16  3.0606   0.099364      0.16057   .
   2 Location        3     16 34.6201 3.1598e-07      0.86651 ***
   3 Tribe:Location  3     16 29.9354 8.4929e-07      0.84878 ***
```

Alternatively, partial *eta*-squared can be calculated from the F value and degrees of freedom.

```
Result = anova(model)

Result$part.eta.sq = with(Result, `F value` * `Df` / (`F value` * `Df` + `Df.res`))

Result

   Analysis of Variance of Aligned Rank Transformed Data

                    Df Df.res F value     Pr(>F) part.eta.sq.    
   1 Tribe           1     16  3.0606   0.099364      0.16057   .
   2 Location        3     16 34.6201 3.1598e-07      0.86651 ***
   3 Tribe:Location  3     16 29.9354 8.4929e-07      0.84878 ***
```

Efron’s *pseudo r-squared, RMSE*

Efron’s *pseudo r-squared* is based on the actual values of the dependent variable and the values predicted by the model.

At the time of writing, the *ARTool* object doesn’t contain the actual values of the dependent variable, so these values have to be called from the original data.

Also, at the time of writing, the residuals in the *ARTool* object don’t reflect the random effects in the model. The effect is that Efron’s *pseudo r-squared* for an *ARTool* object will be equal to the *r-squared* for a similar linear model ignoring any random effects in the *ARTool* model.

```
library(rcompanion)

efronRSquared(actual   = Data$Midichlorians,
              residual = model$residuals)

   EfronRSquared
           0.948


model.lm = lm(Midichlorians ~ Tribe + Location + Tribe:Location,
               data = Data)

summary(model.lm)$r.squared

   [1] 0.9484945
```

The *efronRSquared* function can produce other useful statistics, like mean absolute percent error, root mean square error, and coefficient of variation as a percentage.

```
library(rcompanion)

efronRSquared(actual     = Data$Midichlorians,
              residual   = model$residuals,
              statistic = "MAPE")

efronRSquared(actual     = Data$Midichlorians,
              residual   = model$residuals,
              statistic = "RMSE")

efronRSquared(actual     = Data$Midichlorians,
              residual   = model$residuals,
              statistic = "CV")

   MAPE
   0.0908

   RMSE
   1.34

     CV
   9.71
```

One-way example

The following example addresses the data from the *Kruskal–Wallis Test* chapter. The results are relatively similar to results from the Kruskal–Wallis and Dunn tests, and to those from ordinal regression. Here, the *p*-value for the global test by ART anova is lower than that from the Kruskal–Wallis test.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Speaker  Likert
 Pooh      3
 Pooh      5
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      4
 Pooh      5
 Pooh      5
 Piglet    2
 Piglet    4
 Piglet    2
 Piglet    2
 Piglet    1
 Piglet    2
 Piglet    3
 Piglet    2
 Piglet    2
 Piglet    3
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    4
 Tigger    5
 Tigger    3
 Tigger    5
 Tigger    4
 Tigger    4
 Tigger    3
")


### Order levels of the factor; otherwise R will alphabetize them

Data$Speaker = factor(Data$Speaker,
                      levels=unique(Data$Speaker))


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)


### Aligned ranks anova

library(ARTool)

model = art(Likert ~ Speaker,
                data = Data)

anova(model)

   Analysis of Variance of Aligned Rank Transformed Data

   Table Type: Anova Table (Type III tests)
   Model: No Repeated Measures (lm)
   Response: art(Likert)

             Df Df.res F value     Pr(>F)
   1 Speaker  2     27  18.702 8.0005e-06 ***



### Post-hoc comparisons

model.lm = artlm(model, "Speaker")

library(emmeans)

marginal = emmeans(model.lm,
                   ~ Speaker)

pairs(marginal,
      adjust = "tukey")

   contrast        estimate   SE df t.ratio p.value
    Pooh - Piglet      14.1 2.51 27   5.619  <.0001
    Pooh - Tigger       1.8 2.51 27   0.717  0.7555
    Piglet - Tigger   -12.3 2.51 27  -4.901  0.0001

   P value adjustment: tukey method for comparing a family of 3 estimates


Sum = as.data.frame(marginal)

library(rcompanion)

cldList(p.value ~ contrast, data=Sum)

      Group Letter MonoLetter
   1   Pooh      a          a
   2 Piglet      b           b
   3 Tigger      a          a
```

Partial *eta*-squared

```
Result = anova(model)

Result$part.eta.sq = with(Result, `Sum Sq`/(`Sum Sq` + `Sum Sq.res`))

Result
```

```
   Analysis of Variance of Aligned Rank Transformed Data

             Df Df.res F value     Pr(>F) part.eta.sq
   1 Speaker  2     27  18.702 8.0005e-06     0.58077 ***
```

Efron's *pseudo r-squared*

```
library(rcompanion)

efronRSquared(actual   = Data$Likert,
              residual = model$residuals)

   EfronRSquared
           0.614
```

Repeated measures example

The following example addresses the data from the *Friedman Test* chapter. Results are relatively similar to results from the Friedman and Conover tests, and to those from ordinal regression. Here, the *p*-value for the global test by ART anova is lower than that from the Friedman test.

```
Data = read.table(header=TRUE, stringsAsFactors=TRUE, text="

 Instructor        Rater  Likert
 'Bob Belcher'        a      4
 'Bob Belcher'        b      5
 'Bob Belcher'        c      4
 'Bob Belcher'        d      6
 'Bob Belcher'        e      6
 'Bob Belcher'        f      6
 'Bob Belcher'        g     10
 'Bob Belcher'        h      6
 'Linda Belcher'      a      8
 'Linda Belcher'      b      6
 'Linda Belcher'      c      8
 'Linda Belcher'      d      8
 'Linda Belcher'      e      8
 'Linda Belcher'      f      7
 'Linda Belcher'      g     10
 'Linda Belcher'      h      9
 'Tina Belcher'       a      7
 'Tina Belcher'       b      5
 'Tina Belcher'       c      7
 'Tina Belcher'       d      8
 'Tina Belcher'       e      8
 'Tina Belcher'       f      9
 'Tina Belcher'       g     10
 'Tina Belcher'       h      9
 'Gene Belcher'       a      6
 'Gene Belcher'       b      4
 'Gene Belcher'       c      5
 'Gene Belcher'       d      5
 'Gene Belcher'       e      6
 'Gene Belcher'       f      6
 'Gene Belcher'       g      5
 'Gene Belcher'       h      5
 'Louise Belcher'     a      8
 'Louise Belcher'     b      7
 'Louise Belcher'     c      8
 'Louise Belcher'     d      8
 'Louise Belcher'     e      9
 'Louise Belcher'     f      9
 'Louise Belcher'     g      8
 'Louise Belcher'     h     10
")


### Order levels of the factor; otherwise R will alphabetize them

Data$Instructor = factor(Data$Instructor,
                      levels=unique(Data$Instructor))


###  Check the data frame

library(psych)

headTail(Data)

str(Data)

summary(Data)


### Aligned ranks anova

library(ARTool)

model = art(Likert ~ Instructor + (1|Rater),
                data = Data)

anova(model)

   Analysis of Variance of Aligned Rank Transformed Data

   Table Type: Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger
   df)
   Model: Mixed Effects (lmer)
   Response: art(Likert)

                    F Df Df.res     Pr(>F)
   1 Instructor 16.052  4     28 6.0942e-07 ***


### Post-hoc comparisons

marginal = art.con(model, "Instructor")
```

```
marginal

   contrast                        estimate    SE df t.ratio p.value
    Bob Belcher - Linda Belcher     -13.562 3.23 28  -4.202  0.0021
    Bob Belcher - Tina Belcher      -12.750 3.23 28  -3.950  0.0040
    Bob Belcher - Gene Belcher        4.312 3.23 28   1.336  0.6717
    Bob Belcher - Louise Belcher    -16.125 3.23 28  -4.996  0.0003
    Linda Belcher - Tina Belcher      0.812 3.23 28   0.252  0.9991
    Linda Belcher - Gene Belcher     17.875 3.23 28   5.538  0.0001
    Linda Belcher - Louise Belcher   -2.562 3.23 28  -0.794  0.9302
    Tina Belcher - Gene Belcher      17.062 3.23 28   5.287  0.0001
    Tina Belcher - Louise Belcher    -3.375 3.23 28  -1.046  0.8319
    Gene Belcher - Louise Belcher   -20.438 3.23 28  -6.332  <.0001

   Degrees-of-freedom method: kenward-roger
   P value adjustment: tukey method for comparing a family of 5 estimates


Sum = as.data.frame(marginal)

library(rcompanion)

cldList(p.value ~ contrast, data=Sum)

               Group Letter MonoLetter
   1    BobBelcher      a          a
   2  LindaBelcher      b           b
   3   TinaBelcher      b           b
   4   GeneBelcher      a          a
   5 LouiseBelcher      b           b
```

Partial *eta*-squared

For mixed effects models, the partial *eta*-squared can be calculated from the *F* values and the degrees of freedom.

```
Result = anova(model)

Result$part.eta.sq = with(Result, `F` * `Df` / (`F` * `Df` + `Df.res`))

Result

   Analysis of Variance of Aligned Rank Transformed Data

   Table Type: Analysis of Deviance Table (Type III Wald F tests with Kenward-Roger
   df)
   Model: Mixed Effects (lmer)
   Response: art(Likert)

                     F Df Df.res     Pr(>F) part.eta.sq
   1 Instructor 16.052  4     28 6.0942e-07     0.69633 ***
```

Efron's pseudo r-squared

At the time of writing, the residuals in the *ARTool* object don't reflect the random effects in the model. The effect is that Efron's *pseudo r-squared* for an *ARTool* object will be equal to the *r-squared* for a similar linear model ignoring any random effects in the *ARTool* model.

```
library(rcompanion)

efronRSquared(actual   = Data$Likert,
              residual = model$residuals)

   EfronRSquared
           0.51


model.lm = lm(Likert ~ Instructor,
               data = Data)

summary(model.lm)$r.squared

   [1] 0.5101182
```

Optional: Plot of medians and confidence intervals for midichlorians data

```
library(rcompanion)

Sum = groupwiseMedian(Midichlorians ~ Tribe + Location,
                      data=Data,
                      bca=FALSE, percentile=TRUE)

Sum

     Tribe    Location n Median Conf.level Percentile.lower Percentile.upper
   1  Jedi  Burlington 3     20       0.95               20               22
   2  Jedi Northampton 3      8       0.95                8               10
   3  Jedi     Olympia 3     12       0.95               10               15
   4  Jedi     Ventura 3     15       0.95               15               18
   5  Sith  Burlington 3     22       0.95               22               25
   6  Sith Northampton 3     15       0.95               13               17
   7  Sith     Olympia 3      4       0.95                4                5
   8  Sith     Ventura 3     11       0.95                9               12


### Order the levels for printing

Sum$Location = factor(Sum$Location,
                       levels=c("Olympia", "Ventura", "Northampton", "Burlington"))

Sum$Tribe = factor(Sum$Tribe,
                       levels=c("Jedi", "Sith"))


### Plot
```

```
library(ggplot2)

pd = position_dodge(0.4)    ### How much to jitter the points on the plot

png(filename = "Rplot01.png",
    width  = 5,
    height = 5,
    units  = "in",
    res    = 300)

ggplot(Sum,
       aes(x     = Location,
           y     = Median,
           color = Tribe)) +

    geom_point(shape   = 15,
               size    = 4,
             position = pd) +

    geom_errorbar(aes(ymin  =  Percentile.lower,
                      ymax  =  Percentile.upper),
                      width =  0.2,
                      size  =  0.7,
                      position = pd) +

    theme_bw() +
    theme(axis.title   = element_text(face = "bold"),
          axis.text    = element_text(face = "bold"),
          plot.caption = element_text(hjust = 0)) +

    ylab("Median midichlorian count") +
     ggtitle ("Midichlorian counts for Jedi and Sith",
            subtitle = "In four U.S. cities") +

            labs(caption  = paste0("\nMidichlorian counts for two tribes across ",
                                   "four locations. Boxes indicate \n",
                                   "the median. ",
                                   "Error bars indicate the 95% confidence ",
                                   "interval ",
                                    "of the median."),
                            hjust=0.5) +

  scale_color_manual(values = c("blue", "red"))

dev.off()
```

References

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*, 2nd Edition. Routledge.

Elkin, L.A., Kay, M., Higgins, J. and Wobbrock, J.O. 2021. An aligned rank transform procedure for multifactor contrast tests. Proceedings of the ACM Symposium on User Interface