# R Workshop: Non-Parametric Tests

Brier Gallihugh, M.S.

### 2023-06-18

# **Table of contents**

Independent Sample 1 Test (i.e., Wilcoxon's Rank-Sum Test) <sup>2</sup>	. 2
Dependent Sample T Test (i.e., Wilcoxon Signed-Rank Test)	. 3
One Way ANOVA (i.e., Kruskal–Wallis Test)	. 4
Repeated Measures ANOVA (i.e., Friedman's ANOVA)	. 6
<pre>set.seed(10311993) library(tidyverse) library(psych) library(palmerpenguins)</pre>	
<pre>data &lt;- palmerpenguins::penguins %&gt;% dplyr::select(island,body_mass_g)   dplyr::filter(island == "Biscoe"   island == "Dream") %&gt;% na.omit()</pre>	%>% <u>1</u>
data <- droplevels(data)	3

- (1) Take the palmerpenguins data set and use the select function to isolate the island and body\_mass\_g variables
- ② Use the filter() function to select for only observations where the island variable is "Biscoe" or "Dream" and omit any missing data after using the na.omit() function.
- (3) Use the droplevels() function to ensure factor levels no longer present are excluded from analysis.

#### Independent Sample T Test (i.e., Wilcoxon's Rank-Sum Test)<sup>1</sup>



1. This is computationally the same as the Mann-Whitley test (Field et al, 2012)

## Danger

Non-parametric tests are NOT less powerful than their parametric counterpart as long as the distribution of interest is not normally distributed (Field et al, 2012)

```
# Two-Tailed (one-tailed "greater" diff is + OR "lesser" diff is -)
# Correction for p value (TRUE by default, but can change to FALSE correction = FALSE)
ind_t_nonpar <- wilcox.test(body_mass_g ~ island, data = data)</pre>
                                                                           1
                                                                           2
ind_t_nonpar
# Effect Size (from Field et al, 2012)
rFromWilcox<-function(wilcoxModel, N)
                                                                           (3)
z<- qnorm(wilcoxModel$p.value/2)</pre>
r < - z / sqrt(N)
return(r)
}
rFromWilcox(ind_t_nonpar,291)
                                                                           (4)
# Typical reporting for non-parametric involves median values
descriptive_stats <- data %>%
                                                                           (5)
  group_by(island) %>%
  summarize(group_size = n(),
                                                                           (6)
            median = median(body_mass_g),
            mean = round(mean(body_mass_g, na.rm = TRUE),2),
            sd = round(sd(body_mass_g, na.rm = TRUE),2))
print(descriptive_stats)
                                                                           (7)
```

- (1) Initiate a non parametric independent samples t test using the wilcox.test() function.
- (2) Show the output of the independent sample non-parametric t test
- (3) Create a pseudo effect size measure for non parametric tests courtesy of Field et al (2012).
- (4) Generate an effect size measure r using the function defined above. The function requires

you provide the wilcox.test object and the sample size

- (5) Generate some simple descriptive statistics for reporting purposes grouped by island affiliation
- (6) Several statistics included such as sample size, mean, median, and standard deviation.
- (7) Show the output of the above calculations using the print() function.

Wilcoxon rank sum test with continuity correction

## Dependent Sample T Test (i.e., Wilcoxon Signed-Rank Test)

```
data <- psych::sat.act %>%
                                                                            (1)
  mutate(condition = rep(c(1,2),350),
         ACT_2 = if_else(age < 39, ACT + 3, ACT -1))
data_condition1 <- data %>%
                                                                            (2)
  filter(condition == 1)
data_condition2 <- data %>%
                                                                            (3)
  filter(condition == 2)
condition1_test <- wilcox.test(data_condition1$ACT,</pre>
                                                                            (4)
                             data_condition1$ACT_2,
                             paired = TRUE,
                             correct = FALSE)
condition2_test <- wilcox.test(data_condition2$ACT,</pre>
                                                                            (5)
                             data_condition2$ACT_2,
                             paired = TRUE,
                             correct = FALSE)
```

```
print(condition1_test)
print(condition2_test)

# Effect Size
rFromWilcox(condition1_test,350)
rFromWilcox(condition2_test,350)

?
```

- (1) Create a dependent samples t test data set
- 2 Filter data by condition variable (condition = 1) using the filter() function
- (3) Filter data by condition variable (condition = 2) using the filter() function
- (4) Run a wilcox.test() function for condition = 1. You want to include the time1 and time2 measures. The paired = TRUE tells R that this is a dependent sample t test
- (5) Do the same for the condition = 2 condition.
- (6) The print() function will give you the results for the non parametric dependent sample t test
- (7) Use the function shown earlier by Fields et al (2012) to calculate an effect size. It takes the same arguments as before.

Wilcoxon signed rank test

```
data: data_condition1$ACT and data_condition1$ACT_2
V = 666, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
Wilcoxon signed rank test
data: data_condition2$ACT and data_condition2$ACT_2
V = 666, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
[1] -0.9363034</pre>
```

## One Way ANOVA (i.e., Kruskal-Wallis Test)

[1] -0.9363034

- (1) Create a dummy data set
- (2) Run a non parametric ANOVA using the kruskal.test() function
- (3) Run a posthoc multiple comparisons test using the kruskalmc() function
- (4) Show "means" using the by() function. It takes the rank and condition variables as well as the mean() function
- (5) Sort levels so that the main comparison condition is first using the relevel() function. It takes your condition variable as well as the factor you want first (in this case 3).
- (6) The 'levels() function confirms that our ordering is correct (i.e., 3,1,2)
- (7) Use the kuskalmc() function with the cont argument set to "two-tailed" for planned comparison

Kruskal-Wallis rank sum test

```
data: ACT by condition
Kruskal-Wallis chi-squared = 2.4489, df = 2, p-value = 0.2939
Multiple comparison test after Kruskal-Wallis
alpha: 0.05
Comparisons
   obs.dif critical.dif stat.signif
1-2 20.7875    41.49949    FALSE
1-3 25.4050    41.49949    FALSE
2-3 4.6175    41.49949    FALSE
```

```
data2$condition: 1
[1] 285.1025
data2$condition: 2
[1] 305.89
data2$condition: 3
[1] 310.5075
[1] "3" "1" "2"
Multiple comparison test after Kruskal-Wallis, treatments vs control (two-tailed)
alpha: 0.05
Comparisons
    obs.dif critical.dif stat.signif
3-1 25.4050
                38.85457
                               FALSE
3-2 4.6175
                38.85457
                               FALSE
```

#### Repeated Measures ANOVA (i.e., Friedman's ANOVA)

```
data3 <- data2 %>%
    mutate(ACT_2 = if_else(age < 39, ACT + 3, ACT -1)) %>% select(ACT,ACT_2)

# Test
friedman.test(as.matrix(data3))

# Post Hoc
pgirmess::friedmanmc(as.matrix(data3))

3
```

- (1) Create a repeated measures ANOVA dummy data set
- (2) Run a repeated measures non parametric ANOVA using the friedman.test() function. Note that the as.matrix() function is required for the test to run properly.
- 3 Run a posthoc test using the friedmanmc() function.

```
Friedman rank sum test

data: as.matrix(data3)

Friedman chi-squared = 377.63, df = 1, p-value < 2.2e-16

Multiple comparisons between groups after Friedman test alpha: 0.05

Comparisons
obs.dif critical.dif stat.signif p.value
```

1-2 476 48.00912 TRUE 1.22687e-83



For the Friedman test, the function requires that ONLY the variables needed for the analysis are included in the data frame