Introduction to R and RStudio

Part 3: Introduction to Descriptive and Inferential Statistics with R

Rob Cribbie
Department of Psychology
York University
A researcher is interested in evaluating two therapies for perfectionism; specifically, investigating whether they will be effective in reducing levels of perfectionism.

- Levels of perfectionism are recorded at baseline (perf1), after 1 month of therapy (perf2) and after 2 months of therapy (perf3) for each experimental group (General Stress, CBT) and a control group.

- At baseline the researcher also records levels of depression and the sex of each subject.
## Dataset

```r
> head(dat)

<table>
<thead>
<tr>
<th>sex</th>
<th>group</th>
<th>dep1</th>
<th>perf1</th>
<th>perf2</th>
<th>perf3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>m</td>
<td>cbt</td>
<td>40</td>
<td>78</td>
<td>72</td>
</tr>
<tr>
<td>2</td>
<td>m</td>
<td>cbt</td>
<td>56</td>
<td>66</td>
<td>55</td>
</tr>
<tr>
<td>3</td>
<td>m</td>
<td>cbt</td>
<td>108</td>
<td>82</td>
<td>80</td>
</tr>
<tr>
<td>4</td>
<td>m</td>
<td>cbt</td>
<td>98</td>
<td>75</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>m</td>
<td>cbt</td>
<td>67</td>
<td>71</td>
<td>63</td>
</tr>
<tr>
<td>6</td>
<td>m</td>
<td>cbt</td>
<td>70</td>
<td>73</td>
<td>62</td>
</tr>
</tbody>
</table>
```
Frequencies

- The ‘table’ function is helpful for obtaining frequencies for single or multiple variables.

```r
> table(dat$sex)  > table(dat$sex, dat$group)
```

<table>
<thead>
<tr>
<th></th>
<th>cbt</th>
<th>control</th>
<th>stress</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>13</td>
<td>15</td>
<td>12</td>
</tr>
<tr>
<td>m</td>
<td>17</td>
<td>15</td>
<td>18</td>
</tr>
</tbody>
</table>
Frequencies

- We can also obtain simple graphical descriptions of frequencies
  - `fgroup <- table(dat$group)`
  - `barplot(fgroup)`
Frequencies

- Or, slightly more sophisticated graphical descriptions of frequencies
  ```
  > freq<-table(dat$sex, dat$group)
  > barplot(freq, legend=rownames(freq))
  ```
There are numerous ways to obtain basic descriptive statistics, for example:

- `summary(x)`
- `max(x)`
- `min(x)`
- `range(x)`
- `mean(x)`
- `median(x)`
- `var(x)`
- `sd(x)`
- `quantile(x, probs =)`

These can also be used with all the subsetting/indexing we previously discussed.
However, there are also packages with functions for obtaining a lot of descriptive info quickly
  ◦ E.g., ‘describe’ function in the ‘psych’ package

```r
> library(psy ch)
> describe(dat)
```

<table>
<thead>
<tr>
<th></th>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex*</td>
<td>1</td>
<td>90</td>
<td>1.56</td>
<td>0.50</td>
<td>2.00</td>
<td>1.57</td>
<td>0.00</td>
<td>1.00</td>
<td>2.00</td>
<td>1.00</td>
</tr>
<tr>
<td>group*</td>
<td>2</td>
<td>90</td>
<td>2.00</td>
<td>0.82</td>
<td>2.00</td>
<td>2.00</td>
<td>1.48</td>
<td>1.00</td>
<td>3.00</td>
<td>2.00</td>
</tr>
<tr>
<td>dep1</td>
<td>3</td>
<td>90</td>
<td>81.51</td>
<td>16.79</td>
<td>80.08</td>
<td>80.83</td>
<td>17.61</td>
<td>49.25</td>
<td>129.48</td>
<td>80.24</td>
</tr>
<tr>
<td>perf1</td>
<td>4</td>
<td>90</td>
<td>80.53</td>
<td>9.79</td>
<td>80.24</td>
<td>80.09</td>
<td>8.73</td>
<td>59.67</td>
<td>111.19</td>
<td>51.52</td>
</tr>
<tr>
<td>perf2</td>
<td>5</td>
<td>90</td>
<td>72.55</td>
<td>13.39</td>
<td>71.96</td>
<td>72.06</td>
<td>11.65</td>
<td>43.80</td>
<td>109.34</td>
<td>65.54</td>
</tr>
<tr>
<td>perf3</td>
<td>6</td>
<td>90</td>
<td>69.50</td>
<td>14.83</td>
<td>68.09</td>
<td>68.54</td>
<td>15.17</td>
<td>40.81</td>
<td>119.09</td>
<td>78.28</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>sex*</td>
<td>-0.22</td>
<td>-1.97</td>
<td>0.05</td>
</tr>
<tr>
<td>group*</td>
<td>0.00</td>
<td>-1.53</td>
<td>0.09</td>
</tr>
<tr>
<td>dep1</td>
<td>0.43</td>
<td>-0.30</td>
<td>1.77</td>
</tr>
<tr>
<td>perf1</td>
<td>0.50</td>
<td>0.40</td>
<td>1.03</td>
</tr>
<tr>
<td>perf2</td>
<td>0.37</td>
<td>0.26</td>
<td>1.41</td>
</tr>
<tr>
<td>perf3</td>
<td>0.63</td>
<td>0.37</td>
<td>1.56</td>
</tr>
</tbody>
</table>
Basic Descriptive Statistics

- A similar function in the ‘psych’ package also allows us to obtain descriptive statistics separated by a grouping variable.

```
> describeBy(dat, sex)
group: f

       vars  n    mean    sd median trimmed  mad
  sex*    1  40    1.00 0.00    1.00    1.00    0.00
group*   2  40    1.98 0.80    2.00    1.97    1.48
  dep1    3  40   81.36 18.32  80.62  80.61  20.18
  perf1   4  40   79.47 10.28  79.33  79.24   8.28
  perf2   5  40   72.30 13.70  72.96  71.76  13.72
  perf3   6  40   69.66 14.76  68.21  68.78  15.22
```

This is just the first few lines, several statistics and the info for males is left out.
Hypothesis #1: Are baseline depression and perfectionism scores correlated?

```r
> cor.test(dat$perf1, dat$dep1)

Pearson's product-moment correlation

data:  dat$perf1 and dat$dep1
t = 6.047, df = 88, p-value = 3.493e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
  0.3770138 0.6733433
sample estimates:
  cor
  0.541803
```
Rank–based Correlation

- Hypothesis #1: Are baseline depression and perfectionism scores correlated
  - Spearman’s rank–based correlation coefficient

```r
> cor.test(dat$perf1, dat$dep1, method = "spearman")

Spearman's rank correlation rho

data:  dat$perf1 and dat$dep1
s = 53726, p-value = 1.996e-08
alternative hypothesis: true rho is not equal to 0
sample estimates:
  rho
0.5577561
```
Hypothesis #2: Can we predict posttest perfectionism scores from pretest depression scores?

Scatterplot

\[ \text{plot(dat$dep1,dat$perf3)} \]
Create a linear model object and print a summary of the results

```r
> mod1 <- lm(perf3 ~ dep1, data = dat)
> summary(mod1)
```

```
Call:
  lm(formula = perf3 ~ dep1, data = dat)

Residuals:
   Min     1Q Median     3Q    Max
-26.540 -10.605  -0.683   9.031  47.663

Coefficients:   Estimate Std. Error t value Pr(>|t|)
(Intercept)  45.47798    7.38149   6.161  2.12e-08 ***
dep1          0.29476    0.08871   3.323  0.0013 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.06 on 88 degrees of freedom
Multiple R-squared:  0.1115, Adjusted R-squared:  0.1014
F-statistic: 11.04 on 1 and 88 DF,  p-value: 0.001301
```
Diagnostics are available for identifying normality issues
- `hist(resid(mod1))`

Normal probability (qq) plots are also available

`resid` is a function for extracting the residuals from a model object.
Regression Diagnostics

- It is also easy to produce several diagnostic plots simply by typing:
  - `plot(mod)`
    - Where you replace ‘mod’ with the name of your model

- These plots include:
  - Residuals vs Fitted
  - Normal Probability Plot of the Residuals
  - Scale vs Location Plot
  - Residuals vs Leverage
Regression Diagnostics

- We can compute Studentized Deleted Residuals to identify outlying cases in the solution
  - There are also tools for identifying cases with extreme leverage
- Below case 31 seems discrepant

```r
> library(car)
> outlierTest(mod1)
        rstudent unadjusted p-value Bonferonni p
     31 3.643018    0.00045739    0.041165
```
Simple Regression, cont’d

- It is also important to identify influential observations (e.g., Cook’s Distance)
- This plot indicates which cases exceed a cutoff for Cook’s distance of $4/(N - p)$ where $p$ is the number of predictors

```r
> cutoff <- 4/(length(perf1)-1)
> plot(mod1, which=4, cook.levels=cutoff)
```

Based on fourth plot created when you plot a model object
Identifying Influential Observations

Has a large influence on the model coefficients
Hypothesis #3: Can posttest perfectionism scores be predicted from depression scores, controlling for pretest perfectionism?

```r
> mod2<-lm(perf3 ~ dep1 + perf1, data=dat)
> summary(mod2)
```

```
Call:
  lm(formula = perf3 ~ dep1 + perf1, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-27.4451  -9.1199  -0.8234  10.2411  25.3162

Coefficients:  
  Estimate Std. Error t value  Pr(>|t|)  
(Intercept) -4.75514    10.47354  -0.454  0.6511  
dep1         0.00531     0.08938   0.059  0.9532  
dep1         0.91674     0.15337   5.977  4.87e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.9 on 87 degrees of freedom 
Multiple R-squared:  0.3701, Adjusted R-squared:  0.3557 
F-statistic: 25.56 on 2 and 87 DF,  p-value: 1.851e-09
```
Multiple Regression Diagnostics

- All of the same diagnostics used in a simple regression can also be applied in a multiple regression.
- For example:
  plot(mod2,which=4,cook.levels=cutoff)
Multiple Regression Diagnostics

- There are also diagnostics that are unique to multiple regression, such as collinearity diagnostics
  - Time 1 Depression and Perfectionism do not seem to be too highly related

```r
> vif(mod2)
dep1    perf1
1.415529 1.415529
```
We center the variables first to enhance interpretation. These two models are identical:

```
> dat$deplc <- dat$depl - mean(dat$depl)
> dat$perf1c <- dat$perf1 - mean(dat$perf1)
> mod3 <- lm(perf3 ~ deplc + perf1c + deplc:perf1c, data = dat)
> dat$deplc <- dat$depl - mean(dat$depl)
> dat$perf1c <- dat$perf1 - mean(dat$perf1)
> mod3 <- lm(perf3 ~ deplc + perf1c + deplc:perf1c, data = dat)
> mod3 <- lm(perf3 ~ deplc*perf1c, data = dat)
> summary(mod3)
```

```
Call:
  lm(formula = perf3 ~ deplc * perf1c, data = dat)

Residuals:
     Min      1Q  Median      3Q     Max
-27.1428 -8.2889 -0.1236  6.9596  23.4833

Coefficients:
                       Estimate Std. Error  t value  Pr(>|t|)
  (Intercept)       69.746849   1.296595  53.7923 < 2e-16 ***
  deplc            -0.050374   0.067761  -0.7429     0.459
  perf1c            0.997633   0.143303   6.9662  6.31e-10 ***
  deplc:perf1c  -0.00791908   0.005788  -1.3680     0.174
```
There are infinite numbers of cool stats, plots, etc. that you can find just by playing around with R.
Imagine that we wanted to determine if a perfectionism diagnosis (normal, clinical) at time 1 is related to depression at time 1.

In this case we might want to use a logistic regression, a type of generalized linear model (glm).

First, we will split our time 1 perfectionism scores into normal/clinical:

- `library(Hmisc)`
- `dat$perfdich <- cut2(dat$perf1, cuts=90)`
- `levels(dat$perfdich) <- c("normal", "clinical")`

> 90 is the clinical cutoff

Cut2 performs a split at the specified cutoff.
Logistic Regression

> mod4 <- glm(perfdich ~ dep1, family = binomial(link = "logit"), data = dat)
> summary(mod4)

Call:
glm(formula = perfdich ~ dep1, family = binomial(link = "logit"),
    data = dat)

Deviance Residuals:
     Min       1Q   Median       3Q      Max
-1.4817   -0.5411  -0.3710   -0.2329   2.2472

Coefficients:
            Estimate  Std. Error    z value    Pr(>|z|)
(Intercept)  -7.70958     1.95033    -3.9533     7.72e-05 ***
dep1          0.06880     0.02088     3.2950     0.000983 ***

Generalized Linear Model (glm)
Hypothesis #4: Is there a difference between males and females on pretest perfectionism? In this case we would likely want to use a $t$-test.

```r
> t.test(perf1~sex, data=dat, var.equal=TRUE)

Two Sample t-test

data:  perf1 by sex
t = -0.9167, df = 88, p-value = 0.3618
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:  
  -6.034613  2.224762
sample estimates:
mean in group f mean in group m
  79.47360     81.37852
```

This statement is required since the default test is the Welch test that does not assume equal variances.
Independent Samples $t$-tests

- We can also run a $t$-test by specifying the groups to be compared

```r
> t.test(perf1[sex == "m"], perf1[sex == "f"], var.equal=TRUE, data=dat)

Two Sample t-test

data:  perf1[sex == "m"] and perf1[sex == "f"]
t = 0.9167, df = 88, p-value = 0.3618
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval: -2.224762 6.034613
sample estimates: mean of x mean of y
 81.37852 79.47360
```
Welch’s two independent samples \( t \)-test

- This is the default since this a better overall test than the traditional \( t \)-test

```r
> t.test(perf1~sex, data=dat)

Welch Two Sample t-test

data:  perf1 by sex
t = -0.9075, df = 80.079,
p-value = 0.3669
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  -6.082220  2.272369
sample estimates:
mean in group f  mean in group m
  79.47360       81.37852
```
Wilcoxon–Mann–Whitney nonparametric two independent samples test

> wilcox.test(dat$perf1 ~ dat$sex)

wilcoxon rank sum test with continuity correction

data:  dat$perf1 by dat$sex
w = 905, p-value = 0.4429
alternative hypothesis: true location shift is not equal to 0

This time we skipped the ‘data’ statement and specified the dataset for each variable.
What about a t–test for Nonnormality and Variance Inequality?

- Several procedures have been proposed, although the Welch $t$–test on trimmed means has garnered the most attention.
- Rand Wilcox has written functions that accompany his texts on robust statistics that includes a function for computing the trimmed Welch $t$ (which was developed by Yuen and often referred to as the Yuen test).
  - We will use a package called ‘WRS2’ that makes available many of Wilcox’s most popular functions.
Since we will be testing the null hypothesis that the population trimmed means are equal, it is informative to look at the trimmed means:

```r
> mean(dat$perf1[dat$sex=='m'], tr=.2)
[1] 80.44451
> tapply(dat$perf1, dat$sex, mean, tr=.2)
   f     m
79.04677 80.44451
```
Yuen Test

- Note that like many of Wilcoxon’s functions, the output is not very fancy.

```r
> yuen(dat$perf1 ~ dat$sex)
call:
yuen(formula = dat$perf1 ~ dat$sex)

Test statistic: 0.7588, p-value = 0.45194
95 percent confidence interval:
-5.108    2.3125
```

Population trimmed means do not differ significantly.
Paired Samples t-tests

- Hypothesis #5: Is there a difference between pre and post perfectionism scores?

```r
> t.test(dat$perf1, dat$perf3, paired=TRUE)

Paired t-test

data:  dat$perf1 and dat$perf3
t = 8.8712, df = 89, p-value = 6.942e-14
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
   8.557716 13.497733
sample estimates:
mean of the differences
   11.02772
```
If the distribution of difference scores is not normally distributed, the Wilcoxon signed ranks test can be much more powerful than the paired samples t-test.

```r
> wilcox.test(dat$perf1, dat$perf3, paired=TRUE)

Wilcoxon signed rank test with continuity correction

data:  dat$perf1 and dat$perf3
V = 3676, p-value = 5.732e-11
alternative hypothesis: true location shift is not equal to 0