Workshop

An introduction to R

Andy Buerki

Overview

R is a software tool for statistical analysis. It is today the tool of choice for quantitative linguists and is used increasingly by linguists and scientists/scholars of all specialisations due to its flexibility and expandability to cover virtually any known statistical procedure. In this workshop we are going to use R through an interface called R Studio which facilitates an enhanced user experience. The workshop aims to introduce participants to ways in which R can be used to conduct a range of common statistical analyses. The focus is on how to conduct analyses in R rather than on statistical procedures per se. A full set of handouts and transcripts will enable participants to follow up on topics discussed in the workshop and review everything that was covered.

Topics

<table>
<thead>
<tr>
<th>Time</th>
<th>Basics</th>
<th>Elements of the R Studio interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Importing and exporting data into and out of R</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data manipulation in R: displaying, partially displaying, copying and creating data objects</td>
</tr>
<tr>
<td></td>
<td>Descriptive statistics</td>
<td>Data summarisation functions, checking distribution</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data visualisation functions: producing and exporting plots</td>
</tr>
<tr>
<td>2.10-3.00</td>
<td>Inferential statistics</td>
<td>Correlation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chi-square tests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>t-tests</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ANOVA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regression and multiple regression</td>
</tr>
</tbody>
</table>

Please note that due to time constraints, some of these topics will be covered only very superficially, but a full set of support materials will enable participants to follow up examples in greater detail.

Prerequisites

No prior knowledge of R is assumed. A general understanding of statistical data analysis and advanced computer skills will be helpful, but not essential.

Software installation

For the workshop, university laptops with R and R Studio pre-installed will be supplied. To install R and R Studio on other university-owned computers running Windows, there is an installer in Cardiff Apps > Cardiff Apps > School Applications > ENCAP. To install the software on a private computer, download and install, in this order, R (http://www.stats.bris.ac.uk/R/) and R Studio (http://www.rstudio.com/). Both R and R Studio are free.
Reading List

No preparation is required for the workshop, but for keen participants, I would recommend Mizumoto and Plonsky (2015) for some background on R and its advantages.


Crawley, M. (2013) The R Book, 2nd ed. Chichester: Wiley and Sons [e-book available through the library] This is a very comprehensive reference book on R and statistical analyses using R. It is not written with linguistic data in mind, but is still useful as a reference work.


Levshina, N. (forthcoming). How to do Linguistics with R: Data exploration and statistical analysis, Amsterdam: Benjamins. Hopefully this will be a easier-to-follow book while still dealing with advanced topics. Natalia Levshina is at Leuven where they have an excellent quantitative linguistics research group.

R Basics

First off, R is unforgiving about typos, so unless names of objects and everything else is typed exactly right, we will get errors or unexpected results.

1 Creating and removing objects

Objects are created using arrows to a name

```
AGE <- c(37, 24, 30, 46) or c(8, 6, 5, 10) -> SCORES
c("m", "f", "m", "f") -> GENDER
```

make a data frame out of existing variables

```
our.data <- data.frame(GENDER, AGE, SCORES)
```

you can always create and edit a data frame in Excel, export it as .csv file and import it into R (it is worth making sure headers are imported correctly)

We remove objects like this: `rm(X)`

where 'X' is the object to be removed. The object disappears irretrievably after this command.

2 Exporting data frames

```
write.csv(X, file="FILENAME.csv", col.names=F)
```

X is the name of the data frame, FILENAME.csv is the name of the file you want to create.

3 Editing data frames

Again, if you feel more confident doing this in Excel, that's fine, just export and re-import the data frame into R

```
fix(X) # where X is the name of the data frame. Make changes in the window that comes up, save and close. You can change the name of variables by clicking on them. Some edits (like removing or re-ordering columns or rows) cannot be done with fix(). See '6 change data frames' below for how to do such things.
```

4 Navigating data frames

```
<table>
<thead>
<tr>
<th>AGE</th>
<th>GENDER</th>
<th>SCORES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>30</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>46</td>
<td>10</td>
</tr>
</tbody>
</table>
```

Picking out values WITHOUT column names and row numbers (this only picks out the values themselves and this is usually what you want if you use the values as input to a function):
- we use the '$' sign after the name of data frame to specify the column name
- we can further specify the rows to be displayed in square brackets []

```
our.data$AGE    # displays the values of the variable AGE inside our.data
our.data$AGE[c(1, 4)]
    # displays the values in row 1 AND 4 of the variable AGE inside our.data
```
our.data$AGE[1:3]
    # displays the values in row 1 TO 3 of the variable AGE inside our.data
our.data$AGE[our.data$GENDER == "m"]
    # displays the values of AGE where GENDER is 'm'
We can now put those values we pick out into a function like mean():
mean(our.data$AGE[our.data$GENDER == "f"])
    # displays the mean age of males in our data

Picking out values WITH column names and row numbers
(this usually causes errors if used in functions, but is fine otherwise):
- we provide either only column names or numbers in [] or row numbers COMMA column
  names/numbers.
- we can leave out row or column names/numbers if we want all rows or columns
our.data[1] or  our.data["AGE"]  # first column only
first and second columns:
our.data[1:2] or  our.data[c(1,2)]  or  our.data[c("AGE","GENDER")]
our.data[c(1,4),]  # rows 1 AND 4 of all columns
our.data[c(1,2,4),c("GENDER","SCORES")]
    # rows 1, 2 and 4 of columns 2 and 3
our.data[c(4,3,2,1),c(2,1,3)]
    # rows 1 to 4 in reverse order and columns 2, 1 and 3 in that order

5  Copy data frames (it's a good idea to make a backup copy before changing data frames)
To copy a data frame (for backup for example) we can export it (see above) or just put it
under a new name
our.data->bkup.our.data
now 2 identical data frames exist under our.data and bkup.our.data

6  Change the order of variables or cases in data frames
Again, you can to export to Excel and re-importing into R if you feel that is easiest. In R, to
change the order of columns and rows, deleting columns, rows, etc., we simply display what
we want in the new data frame (see 'picking out values WITH column names and row
numbers' under 'navigating data frames') and then put it into a new name or the same name
if we want to replace the data frame:
our.data[c(4,3,2,1),c(1,3)]->new.name
the new data frame will be new.name
our.data[c(4,3,2,1),c(1,3)]->our.data
this overwrites/replaces our.data
To add a new column, we just tell R what data to put where, e.g.
our.data$AGE*2  # we display each value in AGE, multiplied by 2
our.data$AGE*2->our.data$DBL.AGE  # we put it into a column called
DBL.AGE in our.data
7 Getting an overview
These functions give an overview of a data frame:

\begin{verbatim}
length(X)  # gives the number of columns (or other elements) in X
str(X)    # displays information about the data frame X
summary(X) # displays a summary of the data frame X
\end{verbatim}

8 Converting variables between character, factor and numeric
Here is how we can make certain R uses the correct type for a variable

\begin{verbatim}
c(1,2,3,4,5) -> a
This creates a vector with numbers 1 to 4. This will automatically be a numeric type

as.character(a) -> a  # now the type is changed to character
as.factor(a) -> a  # now the type is changed to factor (= categorical variable)
as.numeric(a) -> a  # now the type is changed back to numeric (= interval variable)
\end{verbatim}

To create an ordinal variable, we might do this

\begin{verbatim}
ranks=c("first","third","second","first","third")
created a vector of character type

ordered(ranks,c("first","second","third"))
created a vector of type ordered factor (=ordinal variable). The ‘ordered’ function takes the
data vector first, then then you need to indicate the ordering after the comma.
\end{verbatim}
# Descriptive statistics

See transcript for application examples to actual data.

1. **Visualising**

<table>
<thead>
<tr>
<th>Tool</th>
<th>Suitable Variables (levels of measurement, how many variables, other things to consider)</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>frequency tables</td>
<td>one or more categorical variables (typically, although ordinal and interval/ratio variables possible)</td>
<td>table(X)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>table(X,Y)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>prop.table(table(X,Y))</td>
</tr>
<tr>
<td>scatterplot</td>
<td>typically two interval/ratio variables, although ordinal variables can be plotted here as well</td>
<td>plot(X, Y)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>abline(lm(Y~X))</td>
</tr>
<tr>
<td>barplot</td>
<td>one or more categorical variables</td>
<td>barplot(table(X,Y),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beside=T,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>legend=c(&quot;a&quot;,&quot;b&quot;)</td>
</tr>
<tr>
<td>histogram</td>
<td>one interval/ratio variable that is continuous</td>
<td>hist(X, breaks=10)</td>
</tr>
<tr>
<td>line graph</td>
<td>on the x-axis you need a variable at least on an ordinal level, typically involving time periods on the y-axis you can either have the values of an interval/ratio variable for frequencies of a categorical one</td>
<td>plot(type=&quot;l&quot;, X,Y)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lines(type=&quot;l&quot;,X,Z)</td>
</tr>
<tr>
<td>pie chart</td>
<td>one categorical variable</td>
<td>pie(table(X),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>labels=c(&quot;a&quot;,&quot;b&quot;)</td>
</tr>
<tr>
<td>boxplot</td>
<td>one or more interval/ratio variables (those are given as X, Y and Z respectively in the code)</td>
<td>boxplot(X,Y,Z)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>text(1:3, mean(X),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mean(Y),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>mean(Z),c(&quot;+&quot;,&quot;+&quot;,&quot;+&quot;)</td>
</tr>
</tbody>
</table>

R commands to adjust graphs

- `xlim=c(0,10), ylim=c(0,10)` # to set the minimum and maximum values for x-axis (xlim) or y-axis (ylim)
- `xaxt="n", yaxt="n"` # suppress the drawing of x-axis (xaxt) or y-axis (yaxt); usually because we want to add those later using axis(), see below
- `main="main title"` # to supply a main title for the graph; `xlab="name", ylab="name"` # to name the x-axis (xlab) or y-axis (ylab)
- `col=c("white", "grey20", "grey60", "grey80", "black")` # to define the colours with which variables are drawn. Include as many colours as you have variables
In combination with plot(): `type="l"` # this indicates: "l" = line (as opposed to points), "b" = both lines and points, "s" = stairs, "h" = histogram-type lines

In combination with plot(): `pch=1` # point character; try out values 1 to 25 to see the different styles

`lty=1` # line type, you can try out different values and see what they look like

`lwd=1` # the weight of lines drawn, a higher number draws a bolder line

To add an axis: (while drawing the plot, use `xaxt="n"` / `yaxt="n"` to suppress the automatic axes)

```r
axis(1, at=c(1,2,3), labels=c("a","b","c"))
```

1=x-axis, 2=y-axis
where (at which values) on the axis to place tick marks

'labels' labels the tick marks with the labels provided

---

2 Summarising

<table>
<thead>
<tr>
<th>Tool</th>
<th>Suitable Variables</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>normally distributed interval/ratio variable (you can calculate it for non-normally distributed variables, but it is not very meaningful in that case)</td>
<td><code>mean(X)</code></td>
</tr>
<tr>
<td>median</td>
<td>ordinal variable or higher level of measurement</td>
<td><code>median(X)</code></td>
</tr>
<tr>
<td>mode</td>
<td>categorical variable or higher level of measurement</td>
<td><code>sort(table(X))</code></td>
</tr>
</tbody>
</table>
| range              | ordinal variable or higher level of measurement                                    | `range(X)`
|                   |                                                                                  | `diff(range(X))` |
| interquartile range| ordinal variable or higher level of measurement                                    | `quantile(X)`
|                   |                                                                                  | `quantile(X)[4]-quantile(X)[2]` |
| standard deviation | interval/ratio variable                                                           | `sd(X)`   |
| variance           | interval/ratio variable                                                           | `var(X)`  |
Inferential Statistics

Hypothesis testing procedure
a. formulate H0 and H1
b. set significance level (also called alpha level)
c. get an overview using descriptive stats
d. test assumptions of stat. procedure to be used
e. calculate p-values for H0
f. decide if result is significant (that is, whether to reject the H0)

Levels of measurement
• ratio scale (for present purposes no different from interval scale)
• Interval scale (values are scaled with equidistant intervals, e.g. 4 is twice as much as 2)
• Ordinal scale (values are ordered but not necessarily w/ equal intervals, e.g. 4th place is not (necessarily) twice 2nd place)
• Nominal / categorical scale (values cannot be ordered, just different, e.g. ‘male’ vs. ‘female’)
• Frequencies: typically need to be treated as frequencies of categories, but can occasionally be abstracted into a ‘measure’ of an interval scale, e.g. number of occurrences of the word ‘blue’ in a text.

Overview of common statistical procedures and their R commands
For further details see handouts. In code examples, X is the name of the first variable, Y the name of the second, Z of the third.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Applications</th>
<th>Assumptions</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation</td>
<td>correlations between 2 variables</td>
<td>two variables (ordinal scale or above) (this is Spearman’s rho)</td>
<td>cor.test(X,Y, method=&quot;spearman&quot;)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>two interval/ratio variables; normally distributed (this is Pearson’s r)</td>
<td>cor.test(X,Y)</td>
</tr>
</tbody>
</table>
| Partial correlation       | correlations between 2 variables while controlling for a third variable | three interval/ratio variables; normally distributed (if using Pearson’s r). In the R command, X is the first variable, Y the second, and Z is the one that needs to be controlled for. Here also, Spearman’s rho can be calculated if normality doesn’t hold by specifying method as spearman | install.packages("ppcor"); library("ppcor")
cor.test(X,Y)
or
cor.test(X,Y, Z, method="spearman") |
| Chi-squared for goodness of fit | comparing frequencies to see if they differ | one categorical variable = one variable holding frequency counts | chisq.test(X)                                                  |
| Chi-squared for independence | to check if frequencies in cross tables are independent | We need to be able to assemble a contingency table with real counted frequencies of occurrence of two categorical variables | matrix(c(650,233,392,623),nrow=2) ->TABLE;
chisq.test(TABLE) |
<table>
<thead>
<tr>
<th>Test Type</th>
<th>Description</th>
<th>Required Variables</th>
<th>R Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-sample t-test</td>
<td>compare sample mean to a known mean</td>
<td>one interval/ratio variable; normally distributed (X) and one known mean of a normally distributed interval/ratio variable (M)</td>
<td><code>t.test(X, mu=M)</code></td>
</tr>
<tr>
<td>t-test (independent)</td>
<td>compare means of two independent samples</td>
<td>two interval/ratio variables; normally distributed = one interval/ratio variable (normally distr.) and one binary variable (w/ 2 categories)</td>
<td><code>t.test(X,Y)</code></td>
</tr>
<tr>
<td>t-test (paired)</td>
<td>compare means of two samples with paired values</td>
<td>two interval/ratio variables; normally distributed (= 1 int/ratio &amp; 1 binary); values in variable X are paired up with values in variable Y</td>
<td><code>t.test(X,Y,paired=T)</code></td>
</tr>
<tr>
<td>One-way ANOVA</td>
<td>compare means of three or more independent variables</td>
<td>three or more interval/ratio variables; normally distributed (= 1 int/ratio &amp; 1 categorical variable with at least 3 categories)</td>
<td><code>aov(X~Y, data=NAME_OF_DATAFRAME) -&gt; result; anova(result); TukeyHSD(result)</code></td>
</tr>
<tr>
<td>Two-way ANOVA</td>
<td>compare means of 3+ variables, classified differently</td>
<td>sets of three or more interval/ratio variables; normally distributed (= 1 int/ratio &amp; 2+ categorical variables with at least 2 categories)</td>
<td><code>install.packages(&quot;car&quot;); library(&quot;car&quot;); aov(X ~ Y*Z, data=DATAFRAME) -&gt; result; Anova(result, type=&quot;III&quot;)</code></td>
</tr>
<tr>
<td>Simple linear regression</td>
<td>predicting values of one variable using another variable (= showing that one variable causes values of the other to change)</td>
<td>typically we need two ratio/interval scale variables, but we can predict a ratio/interval variable from a categorical variable, too. Further assumptions apply.</td>
<td><code>lm(X~Y)-&gt;model summary(model)</code></td>
</tr>
<tr>
<td>Multiple linear regression</td>
<td>predicting values of a variable using several other variables</td>
<td>typically one ratio/interval variable as the dependent variable, then combinations of categorical and/or ratio/interval variables as independent variables. Further assumptions apply.</td>
<td><code>lm(X~Y * Z)-&gt;model summary(model)</code></td>
</tr>
</tbody>
</table>

In general, you get numbers in scientific notation as p-value, type: `options(scipen=9999)` then re-run the command.
Example Correlation

Example data: http://goo.gl/KxruY4 (name: Alcohol)

- learners of Esperanto at level B2
- given different amounts of alcohol
- then given a speaking test
- researcher wants to know if there is a relationship between
  - amount of alcohol
  - score in the speaking test

Analysis: significance of Spearman's rho
Assumptions: ordinal scale or better
Alternatives: if normally distributed we could use Pearson's r

R-command: cor.test(Alcohol$ALCO, Alcohol$SCORE, method="spearman")

R output:

<table>
<thead>
<tr>
<th>Spearman's rank correlation rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>data: Alcohol$ALCO and Alcohol$SCORE</td>
</tr>
<tr>
<td>S = 307.7129, p-value = 3.242e-07</td>
</tr>
<tr>
<td>alternative hypothesis: true rho is not equal to 0</td>
</tr>
<tr>
<td>sample estimates:</td>
</tr>
<tr>
<td>rho</td>
</tr>
<tr>
<td>0.847968</td>
</tr>
</tbody>
</table>

Reporting:

“There was a significant positive correlation ($r_s = 0.85, p < 0.001$) between amount of alcohol consumed and the scores on the speaking test.”
Example chi-squared test for goodness of fit

Example data: http://goo.gl/tjiHA7 (name: Augen)
- frequencies of ‘blaue[n] Augen’ (blue eyes) in a sample of German books from 1900 to 2000.
- source: Google Books (https://books.google.com/ngrams/)
- researcher wants to know if frequency fluctuations year on year are within the sort of fluctuation one would get due to chance

Assumptions: frequency data

R-command: chisq.test(Augen$AUGEN)

```
R output:

Chi-squared test for given probabilities

data:  AUGEN$AUGEN
X-squared = 194.169, df = 20, p-value < 2.2e-16
```

2.2e-16 is scientific notation, to convert: options(scipen=9999), then re-run the command.

Reporting:
“A chi-squared test showed differences were statistically significant (chi-squared 194.169, df = 20, p < 0.001).”

Example chi-squared test for independence

Example data: in two samples of American speech, the following were found:

<table>
<thead>
<tr>
<th></th>
<th>-iŋ</th>
<th>-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>African American English</td>
<td>388</td>
<td>671</td>
</tr>
<tr>
<td>General American</td>
<td>530</td>
<td>221</td>
</tr>
</tbody>
</table>

- researcher wants to know if the pronunciation variation (-iŋ vs. –in) depends on the speech variety (AAE or GA) or is independent of it.

Assumptions: frequency data, no expected values should be smaller than 5.

R-commands: matrix(c(388,530,671,221),nrow=2)-> ING; chisq.test(ING)

```
R output:

Pearson's Chi-squared test with Yates' continuity correction

data:  ING
X-squared = 201.0784, df = 1, p-value < 0.0000000000000022
```

Reporting:
“A chi-square test for independence (with Yates’ continuity correction) indicated that occurrences of –ing and –in were not equally distributed across African American English and General American ( chi-squared 201.08, df = 1, p < 0.001).”
Example t-test

Example data: http://goo.gl/waCNtA (name: Formant)

- frequencies in Hz of the first formant (F1) of male and female subjects
- researcher wants to know if there is a difference in F1 frequencies between females and males
- data taken from Gries (2013)

Analysis: t-test for independent samples

Assumptions: normal distribution, interval scale variables, no paired data

Alternatives: if NOT normally distributed, a Mann-Whitney U Test (aka Wilcoxon test) can be used.

R-command: t.test(formant$HZ_F1[formant$SEX == "M"],
formant$HZ_F1[formant$SEX == "F"])

R output:

Welch Two Sample t-test
data: Formant$HZ_F1[Formant$SEX == "M"] and
Formant$HZ_F1[Formant$SEX == "F"]
t = -2.4416, df = 112.195, p-value = 0.01619
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-80.758016 -8.403651
sample estimates:
mean of x mean of y
 484.2740 528.8548

Reporting:

“A t-test showed that the mean F1 frequency of males (M = 484.3, SD = 87.9) was significantly different from that of females (M = 528.9, SD = 110.8), t(112) = 2.44, p = 0.0162.”
Example one-way ANOVA with post-hoc test

Example data: http://goo.gl/qnUAI (name: Reaction)
- reaction times in word recognition task for words of 3 different levels of familiarity
- researcher wants to know if the reaction times are different for words of differing familiarity

Assumptions: normal distribution of reaction times

Alternatives: if normally distributed we could use a Kruskal Wallis test

R commands: aov(RT~FAMILIARITY, data=Reaction)->results; anova(results)
R output:

<table>
<thead>
<tr>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAMILIARITY</td>
<td>2</td>
<td>33553</td>
<td>16776.6</td>
<td>7.982</td>
</tr>
<tr>
<td>Residuals</td>
<td>52</td>
<td>109294</td>
<td>2101.8</td>
<td></td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

Reporting:
“An ANOVA showed that reaction times differed significantly by familiarity, F(2,52) = 7.98, p < 0.001.”

Post-hoc test: Tukey HSD (or pairwise t-tests with Bonferroni correction of sig.-level)

R command: TukeyHSD(results)
R output:

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = RT ~ FAMILIARITY, data = Reaction)

$FAMILIARITY

<table>
<thead>
<tr>
<th>diff</th>
<th>lwr</th>
<th>upr</th>
<th>p adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>lo-hi</td>
<td>71.90008</td>
<td>26.74512</td>
<td>117.05503</td>
</tr>
<tr>
<td>med-hi</td>
<td>22.25916</td>
<td>-15.34568</td>
<td>59.86401</td>
</tr>
<tr>
<td>med-lo</td>
<td>-49.64091</td>
<td>-87.24575</td>
<td>-12.03607</td>
</tr>
</tbody>
</table>

Reporting:
“An A Tukey HSD post-hoc test showed significant differences between low and high familiarity (p < 0.001) as well as low and medium familiarity (p = 0.007), but the difference between medium and high familiarity was not significant (p = 0.334).”
**Example simple linear regression**

**Example data:** http://goo.gl/KxruY4 (name: Alcohol)

- learners of Esperanto at level B2
- given different amounts of alcohol
- then given a speaking test
- researcher wants to know if we can predict the test scores based on the amount of alcohol

**Assumptions:** linear relationship between X and Y, errors must be normally distributed

**R-command:** `lm(Alcohol$SCORE ~ Alcohol$ALCO) -> result; summary(result)`

**R output:**

```
Call: lm(formula = Alcohol$SCORE ~ Alcohol$ALCO)

Residuals:       Min        1Q  Median        3Q       Max
-6.7855   -1.8190  -0.6919   1.6810   8.1877

Coefficients:             Estimate Std. Error   t value   Pr(>|t|)
(Intercept)         6.6785     1.3367    4.9960 0.00006047 ***
Alcohol$ALCO        1.0134     0.1497    6.7719 0.00000107 ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

Residual standard error: 3.146 on 21 degrees of freedom
Multiple R-squared:  0.6859, Adjusted R-squared:  0.6709
F-statistic: 45.85 on 1 and 21 DF,  p-value: 0.000001068
```

**Testing assumptions:** plot to see if it’s linear (should have done that anyways); plot residuals (errors): `hist(result$residuals)` or test if error (residuals) are normally distributed using a Shapiro-Wilk test: `shapiro.test(result$residuals)` – if this test is NOT significant, we should be fine because the residuals are then NOT distributed significantly differently from normal.

**Reporting:**

“A simple linear regression showed that the amount of alcohol predicted scores in the speaking test (adjusted $R^2 = 0.671$, df = 21, $p < 0.001$)”
Example multiple linear regression

**Example data:** http://goo.gl/KxruY4 (name: Alcohol)

- learners of Esperanto at level B2
- given different amounts of alcohol
- then given a speaking test and a personality test showing the degree to which a subject has an extrovert personality (scores 1 to 20)
- the researcher wants to know if we can predict the test scores based on the amount of alcohol and scores on the extrovert personality test.

**Assumptions:** linear relationship between X and Y/Z, errors must be normally distributed

**R-command:** `lm(Alcohol$SCORE~Alcohol$ALCO+Alcohol$EXTR) -> result; summary(result)`

**R output:**

```
Call:
  lm(formula = Alcohol$SCORE ~ Alcohol$ALCO + Alcohol$EXTR)

Residuals:
               Min           1Q       Median           3Q          Max
  -6.9958      -2.2081       0.0195       1.6808       7.8425

Coefficients:     Estimate Std. Error  t value Pr(>|t|)
  (Intercept)    5.3491     1.8512     2.890    0.00906 **
  Alcohol$ALCO   1.0253     0.1498     6.843 0.00000119 ***
  Alcohol$EXTR   0.1111     0.1072     1.036    0.31251
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1

Residual standard error: 3.14 on 20 degrees of freedom
Multiple R-squared:  0.7019,  Adjusted R-squared:  0.6721
F-statistic: 23.54 on 2 and 20 DF,  p-value: 0.000005544
```

**Testing assumptions:** same as simple linear regression

**Reporting:** “When speaking test scores were predicted using a multiple regression, it was found that the amount of alcohol consumed was a significant predictor (p < .001), but extroversion was not (p = 0.31). The overall model fit was adjusted $R^2 = 0.672$, df = 2,21, p < 0.001.”