

Cheat Sheet for R and RStudio

L. Jason Anastasopoulos

April 29, 2013

1 Downloading and Installation

- First download R for your OS: R
- Next download RStudio for your OS: RStudio

2 Uploading Data into R-Studio

R-Studio Makes uploading CSV files into R extremely simple. Just follow these instructions and you'll be using R in no time.

1. Download your **.csv** data to a folder that you can easily find.
2. Open R-Studio.
3. In the interpreter (lower left-hand box of RStudio), type *library(foreign)* and hit Enter. This will install the package that reads your **.csv** files.
4. In the box on the upper-right hand corner of RStudio, click on the tab that says "Workspace".
5. Then click on "Import Dataset > From Text File...". Find your **.csv** dataset and open it.
6. In the interpreter (lower left-hand box), type in *attach(name-of-dataset)* and hit Enter. You can find the name of the dataset listed under the "Workspace" tab in the upper right-hand corner of RStudio.
7. To find the variable names in your dataset type *names(name-of-dataset)* and hit Enter.

3 Doing Statistics in RStudio

After you have opened your data, doing statistics is really easy. Below is a list of commands that you will need to do any kind of statistics in RStudio.

3.1 Summary Statistics

- **summary(X)** - Summary statistics such as mean, median, mode and quartiles for a variable.

```
> summary(X)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.0360 -0.8855 -0.2475 -0.2382  0.3345  3.4460
```

- **mean(X,na.rm=TRUE)** - Produces the mean of the variable. Removes missing observations.

```
> mean(X,na.rm=TRUE)
[1] -0.2382041
```

- **sd(X,na.rm=TRUE)** - Produces the standard deviation of the variable. Removes missing observations.

```
> sd(X,na.rm=TRUE)
[1] 0.9604155
```

3.2 Regression

- **lm(Y ~ X)** - Runs a regression of **Y** on **X** where **Y** is your *dependent* variable and **X** is your *independent* variable. You need to save your model in R's memory first and can get the regression coefficients and other info you need by using the *summary()* command. For example, for simple regression:

```
> model1 = lm(Y~X)
> summary(model1)
```

```
Call:
lm(formula = Y ~ X)
```

```
Residuals:
Min       1Q   Median       3Q      Max
```

```
-2.6068 -0.8068 0.0700 0.7027 3.3292
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.18866 0.11548 -1.634 0.106
X 0.07123 0.11726 0.607 0.545
```

```
Residual standard error: 1.121 on 98 degrees of freedom
Multiple R-squared: 0.003752, Adjusted R-squared: -0.006414
F-statistic: 0.369 on 1 and 98 DF, p-value: 0.5449
```

for multiple regression...

```
> model1.1 = lm(Y~X + Z)
> summary(model1.1)
```

Call:

```
lm(formula = Y ~ X + Z)
```

Residuals:

```
Min      1Q  Median      3Q      Max
-2.6534 -0.7729  0.0340  0.6860  3.2037
```

Coefficients:

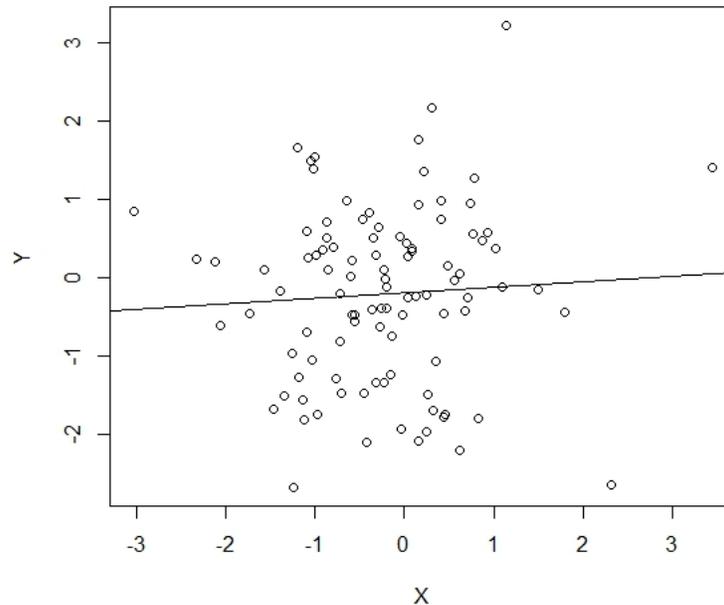
```
Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.19525 0.11582 -1.686 0.095 .
X 0.06916 0.11739 0.589 0.557
Z -0.10228 0.11333 -0.902 0.369
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.122 on 97 degrees of freedom
Multiple R-squared: 0.01205, Adjusted R-squared: -0.008323
F-statistic: 0.5914 on 2 and 97 DF, p-value: 0.5555
```

- **plot(X,Y)** - Will produce a scatterplot of the variables **X** and **Y** with **X** on the x-axis and **Y** on the y-axis.
- **abline(regression model)** - Will draw a regression line of the regression model that you saved through a scatterplot. For example:

```
> model2 = lm(Y~X)
> plot(X,Y)
> abline(model2)
```



3.3 Hypothesis Testing

- `t.test(X,Y)` - Performs a t-test of means between two variables **X** and **Y** for the hypothesis $H_0 : \mu_X = \mu_Y$. Gives t-statistic, p-value and 95% confidence interval. Example:

```
> t.test(X,Y)
```

Welch Two Sample t-test

data: X and Y

t = -0.2212, df = 193.652, p-value = 0.8252

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

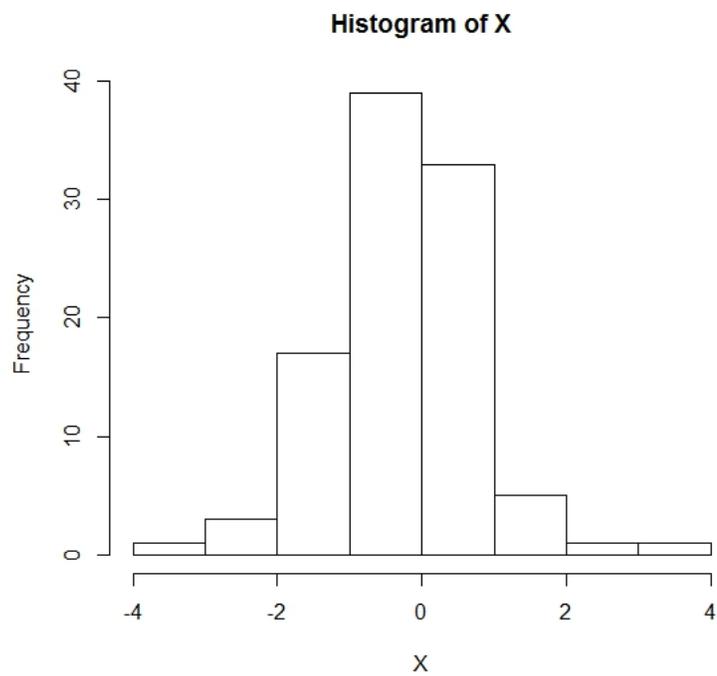
-0.3231116 0.2579525

```
sample estimates:
mean of x  mean of y
-0.2382041 -0.2056246
```

3.4 Graphics and Plots

- **hist(X)** - Will produce a histogram of the variable **X**.

```
> hist(X)
```



- **plot(X,Y)** - Will produce a scatterplot of the variables **X** and **Y** with **X** on the x-axis and **Y** on the y-axis.

```
> plot(X,Y)
```

