

R Workshop Module 4: Some Statistical Analyses

Katherine Thompson (katherine.thompson@uky.edu)

Department of Statistics, University of Kentucky

October 22, 2013

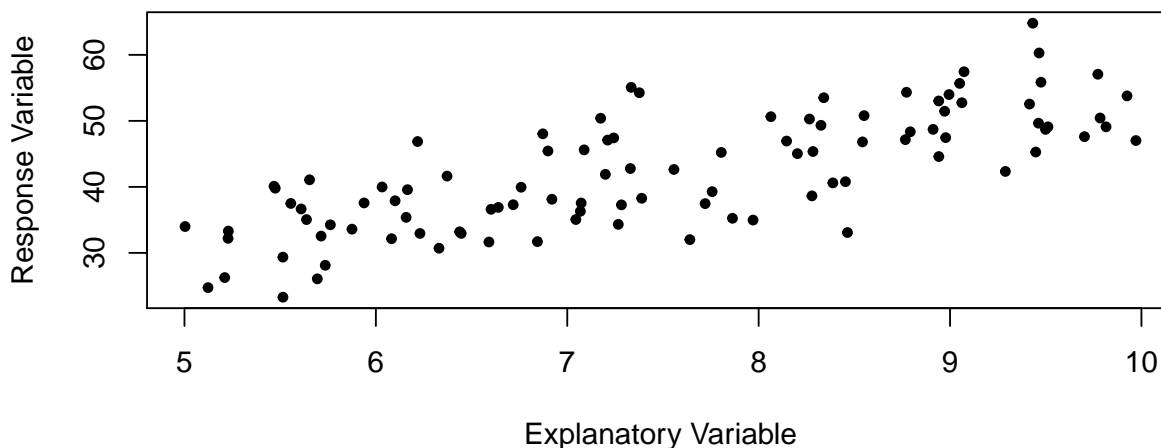
Reading in Data

Start by reading the dataset 'practicedata.txt' into R. (Having trouble? See the instructions in Module 2.)

Linear Regression

For this example, we will investigate the relationship between the variables `respvar` and `expvar` from the data set, `practicedata`. Remember than by using a '\$', we can refer to the variable as `practicedata$respvar` in the following code. To fit a linear model, we use the function `lm()` as follows.

```
## Scatter plot of the data
plot(practicedata$expvar,practicedata$respvar,xlab='Explanatory Variable',
      ylab='Response Variable',pch=20)
```



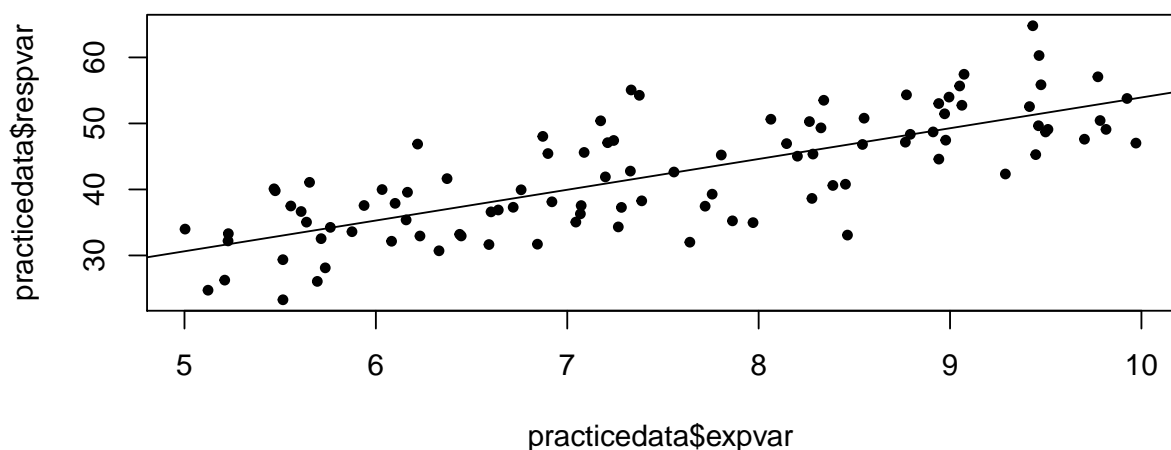
```
## Fit a linear model to the Data
fitted.model = lm(respvar~expvar,
                  data=practicedata) # Fit a linear model with
                                     # y-variable respvar and x-variable expvar
summary(fitted.model) #Summarize the linear model

##
## Call:
## lm(formula = respvar ~ expvar, data = practicedata)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.692  -4.147  -0.238   4.210  13.561
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.33      3.05      2.4   0.018 *
## expvar           4.66      0.40     11.6 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.67 on 98 degrees of freedom
## Multiple R-squared:  0.581, Adjusted R-squared:  0.577
## F-statistic: 136 on 1 and 98 DF,  p-value: <2e-16
```

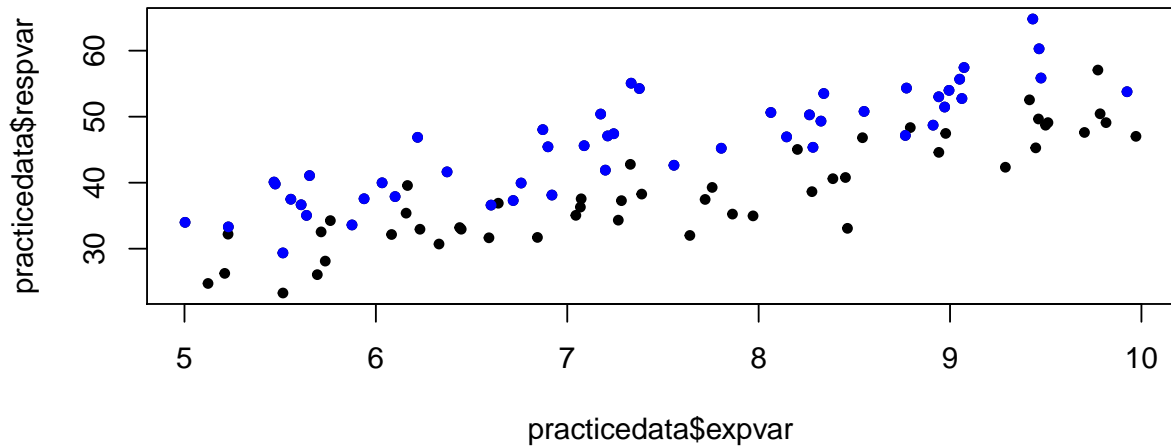
```
## Check to make sure that the model looks appropriate
plot(fitted.model)
```

```
##Plot the data with the fitted regression line
plot(practicedata$expvar,practicedata$respvar,pch=20)
abline(fitted.model)
```



```
##Plotting the data and the fitted model
plot(practicedata$expvar,practicedata$respvar,pch=20)
points(practicedata$expvar[practicedata$groupvar=='Treatment'],
       practicedata$respvar[practicedata$groupvar=='Treatment'],
```

```
col='blue',pch=20)
```



```
##You can also fit a linear model with more than one variable
fitted.group.model = lm(respvar~expvar + groupvar,
                        data=practicedata) # Fit a linear model with
                        # y-variable respvar and x-variables expvar and groupvar
summary(fitted.group.model) #Summarize the linear model

##
## Call:
## lm(formula = respvar ~ expvar + groupvar, data = practicedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.723  -2.411  -0.168   2.164   9.410
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.463      2.118   0.69   0.49
## expvar           4.885      0.269  18.13 <2e-16 ***
## groupvarTreatment 8.374      0.764  10.96 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.81 on 97 degrees of freedom
## Multiple R-squared:  0.813, Adjusted R-squared:  0.809
## F-statistic: 211 on 2 and 97 DF, p-value: <2e-16
```

```

##You can also fit a linear model with both variables and their interaction
fitted.int.model = lm(respvar~expvar + groupvar + expvar*groupvar,
                      data=practicedata) # Fit a linear model with
                      # y-variable respvar and
                      # x-variables expvar, groupvar and their interaction
summary(fitted.int.model) #Summarize the linear model

##
## Call:
## lm(formula = respvar ~ expvar + groupvar + expvar * groupvar,
##     data = practicedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.634 -2.455 -0.068  2.238  9.416
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.241      2.874   0.78   0.44
## expvar              4.782      0.371  12.88 <2e-16 ***
## groupvarTreatment    6.741      4.131   1.63   0.11
## expvar:groupvarTreatment 0.218      0.542   0.40   0.69
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.83 on 96 degrees of freedom
## Multiple R-squared:  0.813, Adjusted R-squared:  0.807
## F-statistic: 139 on 3 and 96 DF, p-value: <2e-16

```

Installing and Using Packages

The R components we have used thus far only show a very small part of the available functions in R, the “base” version. Any user can upload a “package” to R with new functions. Here is an example of installing and using the package `lme4`.

Installing a Package

You only need to install a package once per machine that you use. To install a package in RStudio, click “Tools” and “Install Packages...” Then type the name of the package you want to install (`lme4` here) and “Install.”

Loading a Package

Now that you have a package installed, you need to load it before you can use any functions from it. To load a package, use the `library()` function. If you use a certain package a lot, you can put this line at the top of each script you write so that it is loaded by default.

```
library(lme4)

## Warning: package 'lme4' was built under R version 3.0.2
## Loading required package: lattice
## Loading required package: Matrix
```

Using a Package

Using a package involves using the functions defined within the package. For instance, there is a function `lmer()` in `lme4` which allows you to fit mixed models.

```
## Example from lmer function and documentation
## Model with only fixed effects:
fixed.model <- lm(Reaction ~ Days + Subject, sleepstudy)
summary(fixed.model)
## Model with fixed and random effects
random.model <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
summary(random.model)
```