hw4

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

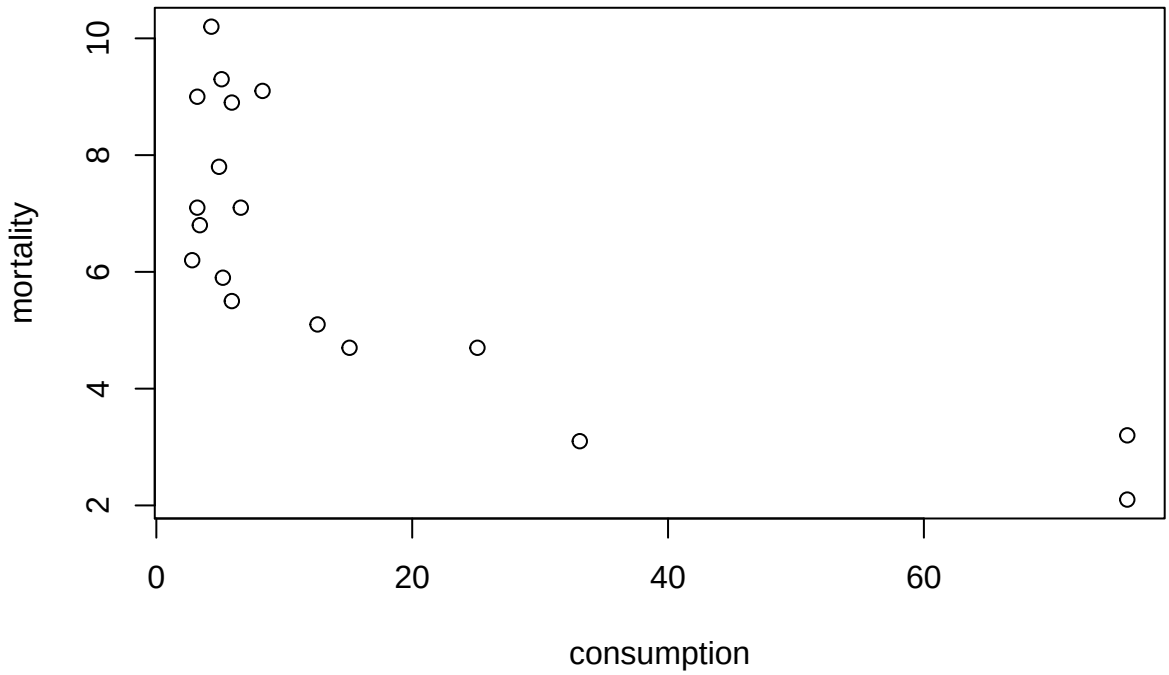
Including Plots

You can also embed plots, for example:

```
#1a
data = read.table("wine.txt")
data
```

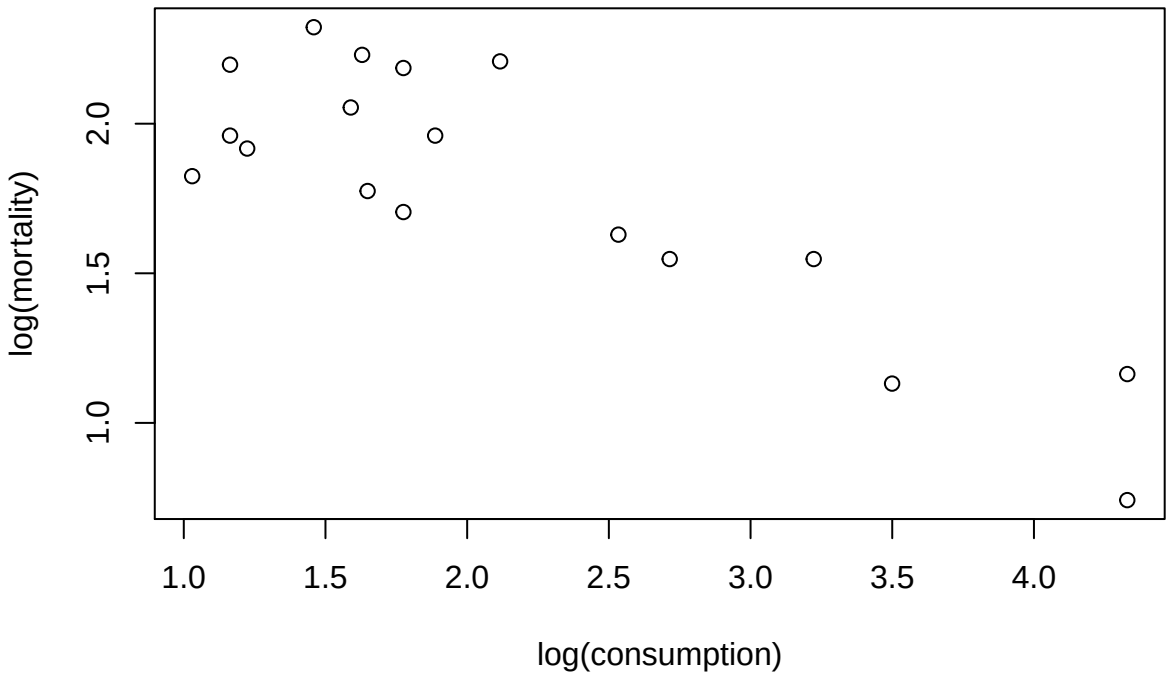
```
##    consumption mortality      country
## 1          2.8       6.2       Norway
## 2          3.2       9.0     Scotland
## 3          3.2       7.1      England
## 4          3.4       6.8      Ireland
## 5          4.3      10.2      Finland
## 6          4.9       7.8       Canada
## 7          5.1       9.3 UnitedStates
## 8          5.2       5.9  Netherlands
## 9          5.9       8.9   NewZealand
## 10         5.9       5.5      Denmark
## 11         6.6       7.1       Sweden
## 12         8.3       9.1    Australia
## 13        12.6       5.1      Belgium
## 14        15.1       4.7      Germany
## 15        25.1       4.7      Austria
## 16        33.1       3.1  Switzerland
## 17        75.9       3.2        Italy
## 18        75.9       2.1       France
```

```
plot(mortality ~ consumption, data= data)
```




```
# If we consider mortality as a response variable to the predictor variable, consumption, it is not nec
```

```
#1b
plot(log(mortality) ~ log(consumption), data= data)
```




```
#1c
#The model should predict the response log of mortality rate from heart disease per thousand. We need a
#X is the predictor matrix , yhat is the estimate of the response, beta.hat is the coefficients of the
#y.hat = beta.hat[1] + X*beta.hat[2]
#beta.hat = inv(X'*X)* (X'*y)
#residuals = y - X*beta.hat
```

```r
#1d
X = cbind(1, log(data$consumption))
y = log(data$mortality)
beta.hat = qr.solve(crossprod(X), crossprod(X,y))
beta.hat
```

```
##            [,1]
## [1,]  2.5555519
## [2,] -0.3555959
```

```r
residuals= y- X%*% beta.hat
sE = sqrt(sum(residuals^2)/(length(residuals)-length(beta.hat)))
sE
```

```
## [1] 0.2285367
```

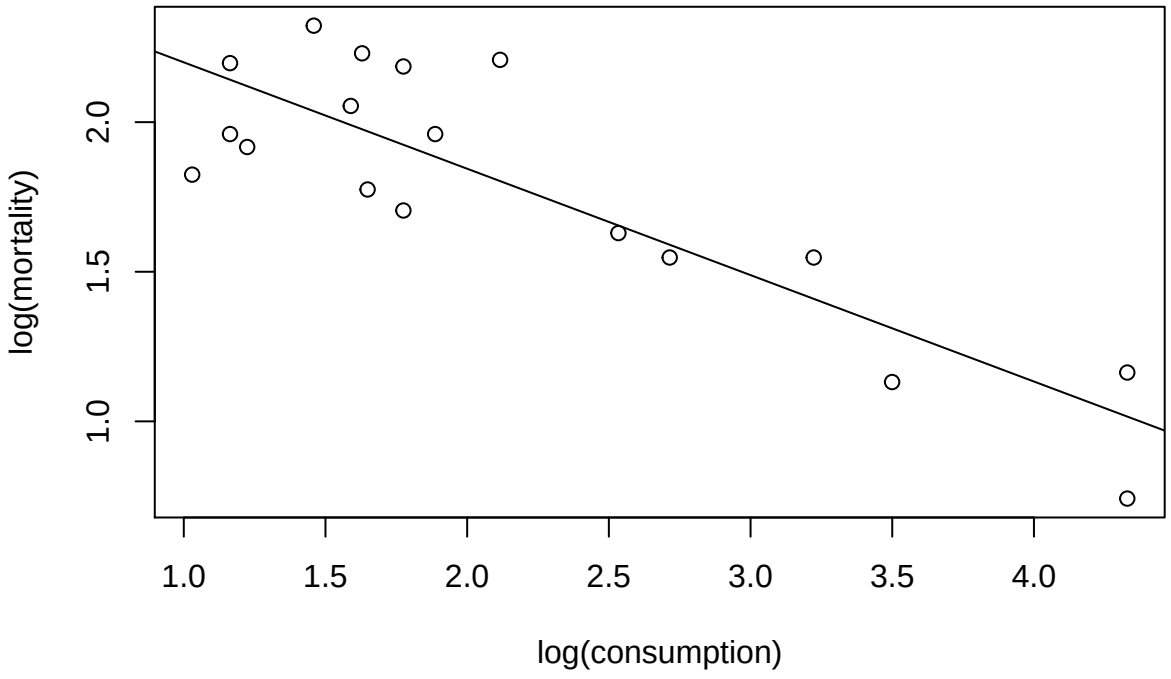
```r
x.vals = seq(from=1.0, to=4.5, by=.1)
X.new = cbind(1, x.vals)
y.hat = X.new%*%beta.hat


#The values of the coefficients: beta1 = 2.55, beta2 = -0.3555. The error of the standard deviation bet
```

```r
#1e
yest = beta.hat[1]+ beta.hat[2]*3.7
yest
```

```
## [1] 1.239847
```

```r
#1f
plot( log(mortality)~ log(consumption), data = data)
abline(a = beta.hat[1], b= beta.hat[2])
```

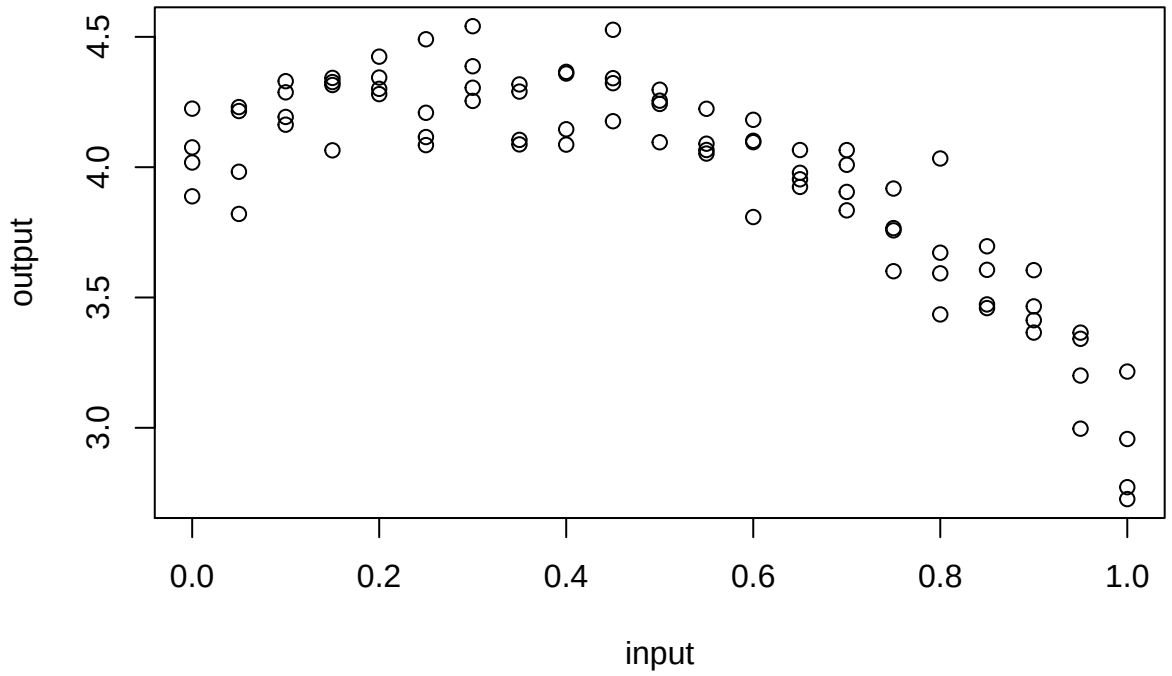



```r
#1g)
logL = log(14.4)
```

```
xnew = c(1,logL)
exp(sum(beta.hat*xnew))
```

```
## [1] 4.988306
```

```
#2a
data1 = read.table("machine.txt")
plot(output ~ input, data= data1)
```



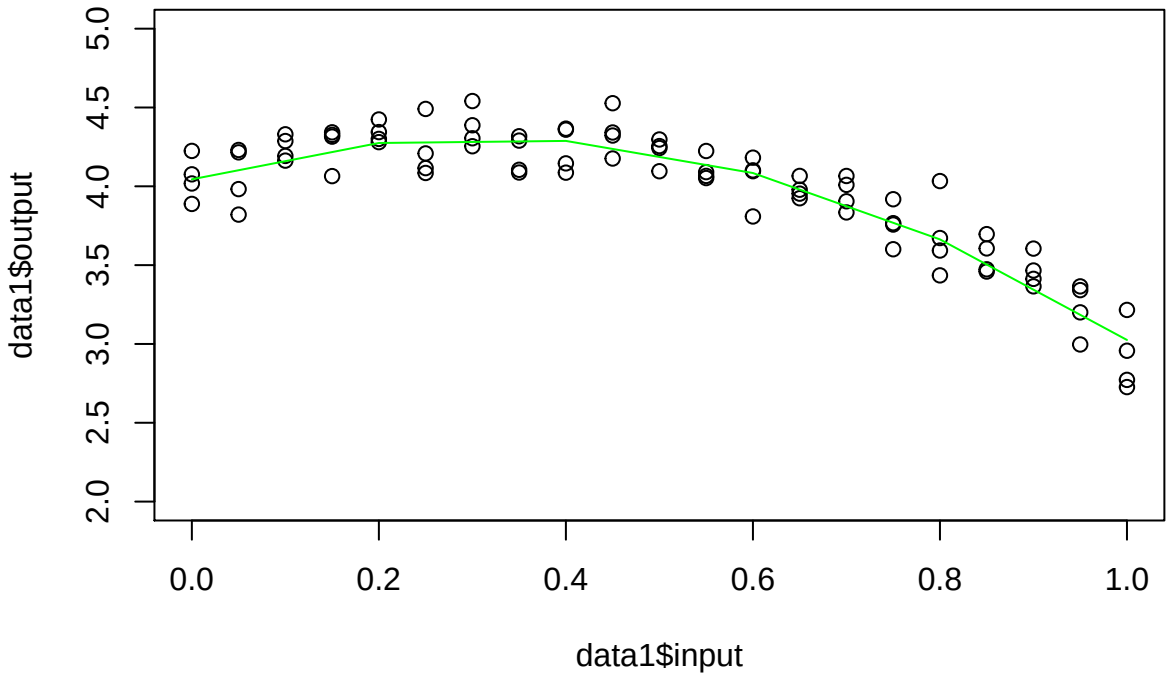

```
#2b
X1 = cbind( 1, data1$input, data1$input^2)
y1 = data1$output
beta.hat1 = qr.solve(crossprod(X1), crossprod(X1,y1))
beta.hat1
```

```
##           [,1]
## [1,]  4.044330
## [2,]  1.694911
## [3,] -2.713946
```

```
residuals = y1- X1%*%beta.hat1
sE = sqrt(sum(residuals^2)/(length(residuals)- length(beta.hat1)))
sE
```

```
## [1] 0.1393174
```

```
#2c
plot(data1$output ~ data1$input, ylim = c(2,5))
xvals1 = seq(from=0, to=1, by=.2)
X.new = cbind(1, xvals1,xvals1^2)
y.hat2 = X.new%*%beta.hat1
lines(xvals1, y.hat2, col="green")
```


data1$output
5.0
4.5
4.0
3.5
3.0
2.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
data1$input