practice

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:

```
#predicting with the regression
#dataset: old faithful eruptions
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
data(faithful)

#create the training and testing model
set.seed(333)
inTrain <- createDataPartition(y=faithful$waiting,
                               p=0.5, list=FALSE)
trainFaith <- faithful[inTrain,]
testFaith <- faithful[-inTrain,]
head(trainFaith)
```

```
##    eruptions waiting
## 3      3.333      74
## 6      2.883      55
## 7      4.700      88
## 8      3.600      85
## 9      1.950      51
## 11     1.833      54
```

```
#plot the training dataset (Eruption duration versus waiting time)
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")

#find the linear model
#Method 1:
lm1 <- lm(eruptions ~ waiting,data=trainFaith)
summary(lm1)
```

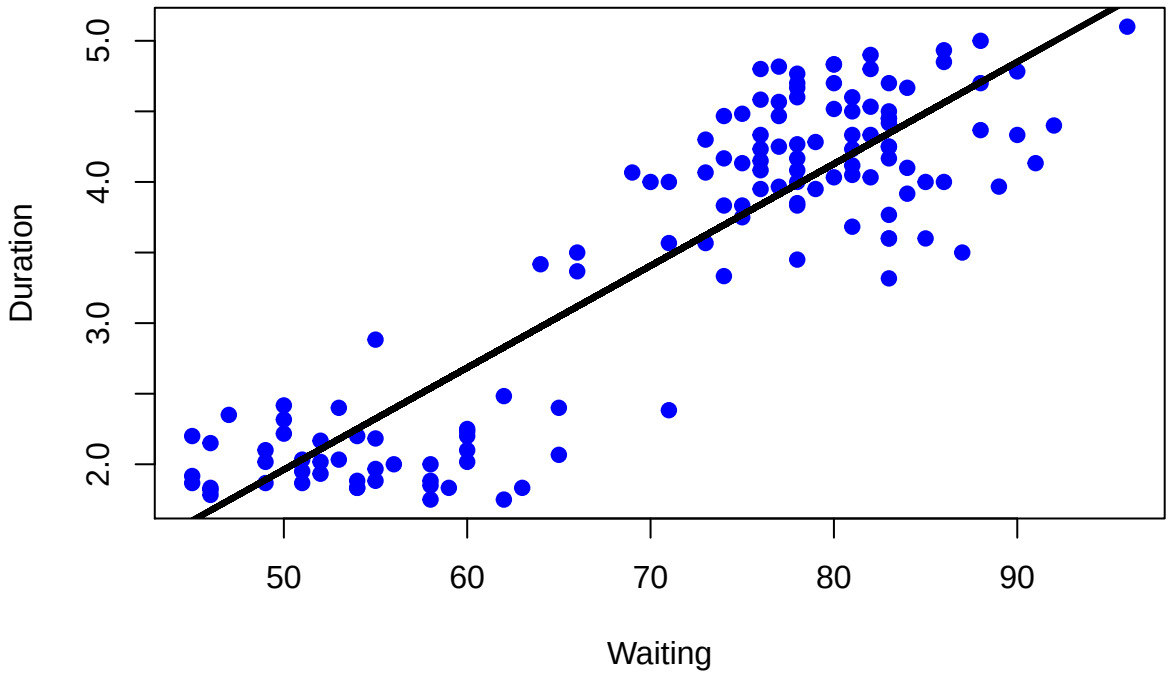
```
##
## Call:
## lm(formula = eruptions ~ waiting, data = trainFaith)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13375 -0.36778  0.06064  0.36578  0.96057
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.648629   0.226603  -7.275 2.55e-11 ***
```

```
## waiting      0.072211   0.003136  23.026  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4941 on 135 degrees of freedom
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7956
## F-statistic: 530.2 on 1 and 135 DF,  p-value: < 2.2e-16
```

```
#Method 2:
#use the caret package to build the model
modFit <- train(eruptions ~ waiting,data=trainFaith,method="lm")
summary(modFit$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.13375 -0.36778  0.06064  0.36578  0.96057
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.648629   0.226603  -7.275 2.55e-11 ***
## waiting      0.072211   0.003136  23.026  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4941 on 135 degrees of freedom
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7956
## F-statistic: 530.2 on 1 and 135 DF,  p-value: < 2.2e-16
```

```
#model fit
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
lines(trainFaith$waiting,lm1$fitted,lwd=3)
```




```
#predict news values based on the linear model fit ( only use the training dataset)
#compute the linear model
coef(lm1)[1] + coef(lm1)[2]*80
```

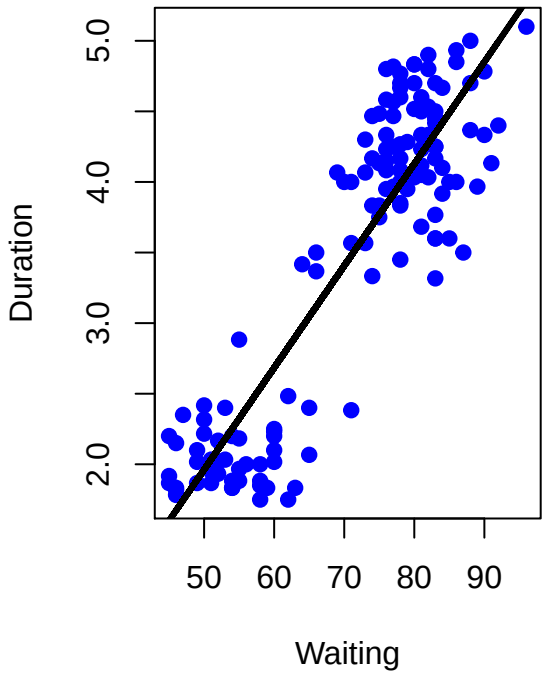
```
## (Intercept)
##    4.128276
```

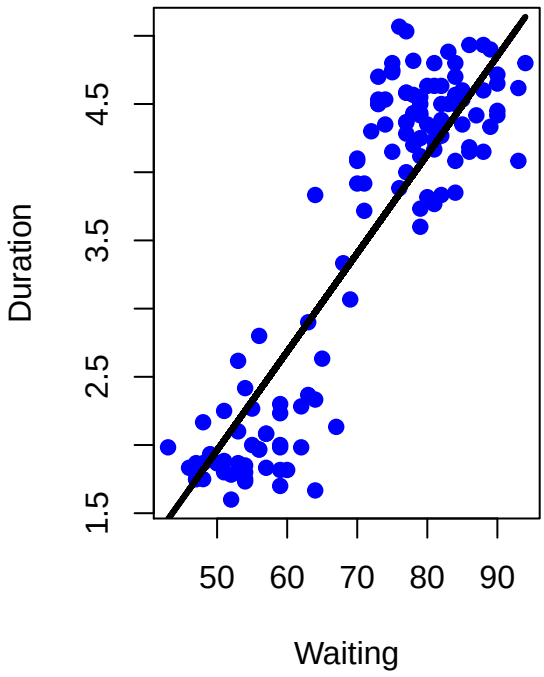
```
newdata <- data.frame(waiting=80)
predict(lm1,newdata)
```

```
##        1
## 4.128276
```

```
#Plot predictions based on fitting lines - training and test
par(mfrow=c(1,2))
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
lines(trainFaith$waiting,predict(lm1),lwd=3)

plot(testFaith$waiting,testFaith$eruptions,pch=19,col="blue",xlab="Waiting",ylab="Duration")
lines(testFaith$waiting,predict(lm1,newdata=testFaith),lwd=3)
```






```
#Get training set/test set errors
# Calculate RMSE on training
sqrt(sum((lm1$fitted-trainFaith$eruptions)^2))
```

```
## [1] 5.740844
```

```
# Calculate RMSE on test
sqrt(sum((predict(lm1,newdata=testFaith)-testFaith$eruptions)^2))
```

```
## [1] 5.853745
```