Predicting with regression

Jeffrey Leek

May 18, 2016

Key ideas

- Fit a simple regression model
- Plug in new covariates and multiply by the coefficients
- Useful when the linear model is (nearly) correct

Pros: * Easy to implement * Easy to interpret

Cons: * Often poor performance in nonlinear settings

Example: Old faithful eruptions

Image Credit/Copyright Wally Pacholka
`http://www.astropics.com/`

Example: Old faithful eruptions

```
library(caret);data(faithful); set.seed(333)
```

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

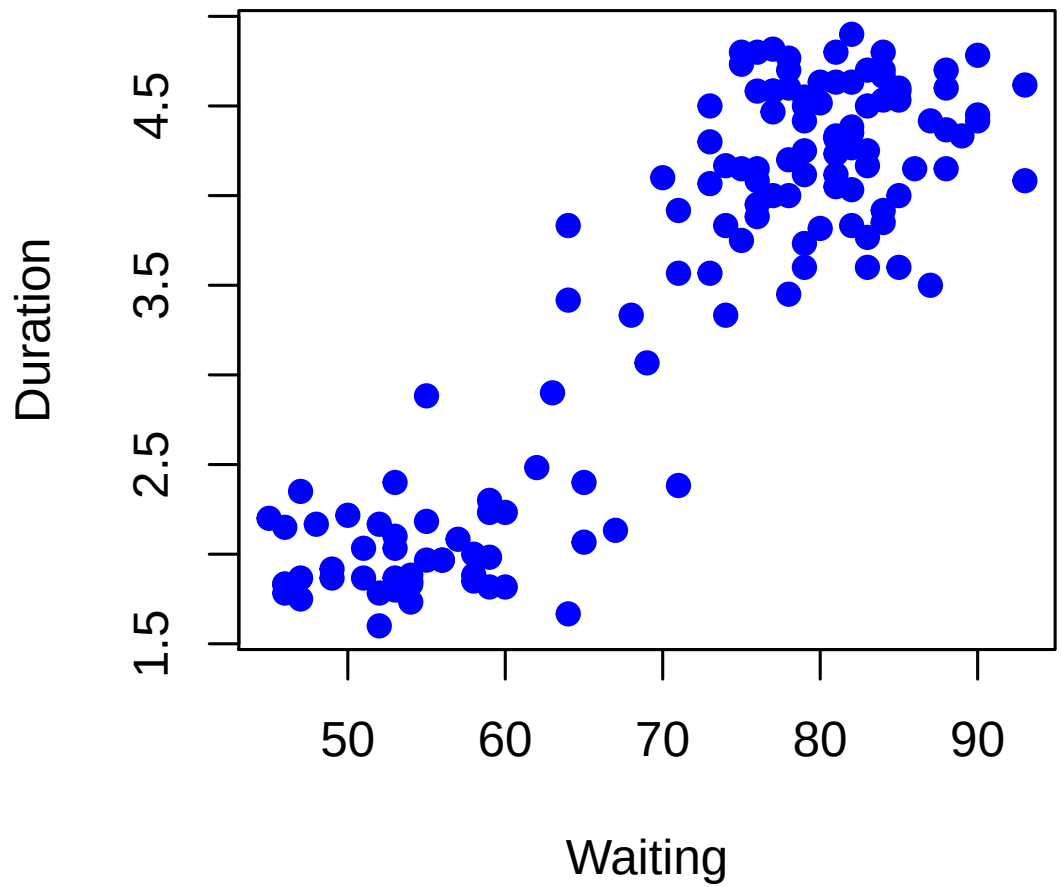
```
## Loading required package: ggplot2
```

```
inTrain <- createDataPartition(y=faithful$waiting,
                               p=0.5, list=FALSE)
trainFaith <- faithful[inTrain,]; testFaith <- faithful[-in
head(trainFaith)
```

```
##   eruptions waiting
## 1     3.600      79
## 3     3.333      74
## 5     4.533      85
## 6     2.883      55
## 7     4.700      88
## 8     3.600      85
```

Eruption duration versus waiting time

```
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="bl
```



Fit a linear model

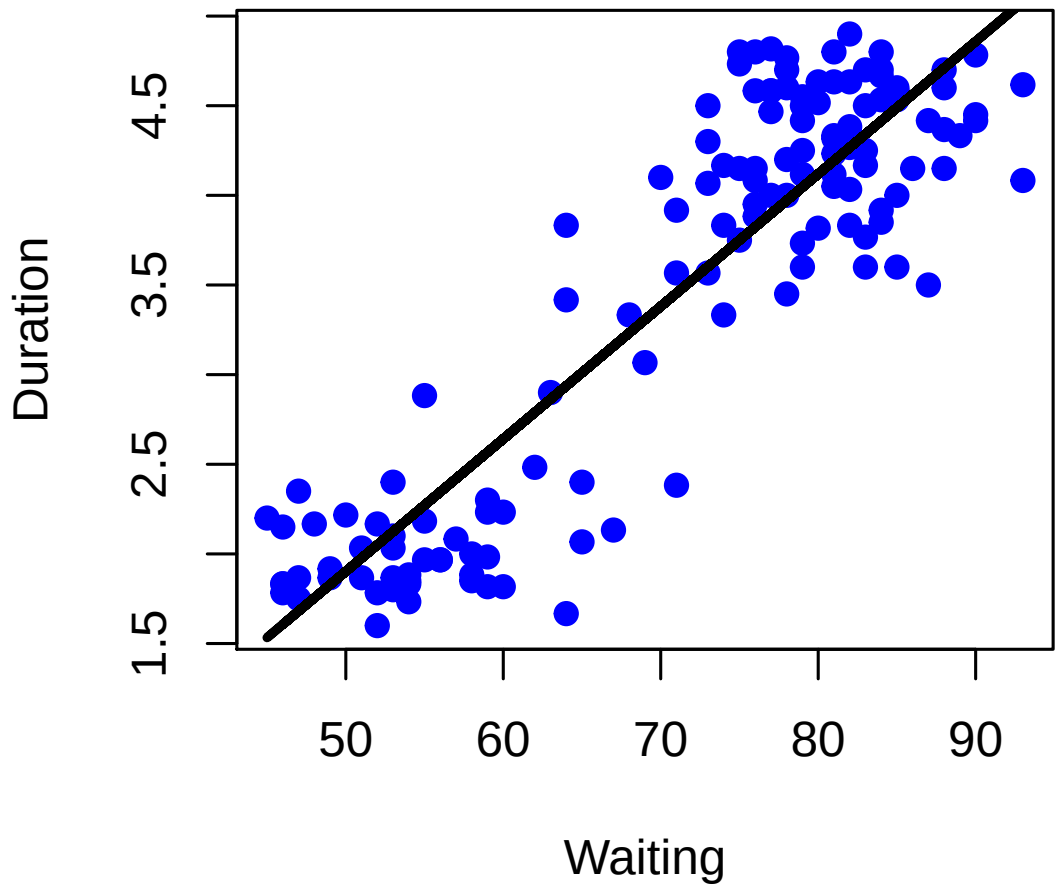
$$ED_i = b_0 + b_1 WT_i + e_i$$

```
lm1 <- lm(eruptions ~ waiting,data=trainFaith)
summary(lm1)
```

```
##
## Call:
## lm(formula = eruptions ~ waiting, data = trainFaith)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26990 -0.34789  0.03979  0.36589  1.05020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.792739   0.227869  -7.867 1.04e-12 ***
## waiting      0.073901   0.003148  23.474  < 2e-16 ***
```

Model fit

```
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="bl
lines(trainFaith$waiting,lm1$fitted,lwd=3)
```



Predict a new value

$$\hat{ED} = \hat{b}_0 + \hat{b}_1 WT$$

```
coef(lm1)[1] + coef(lm1)[2]*80
```

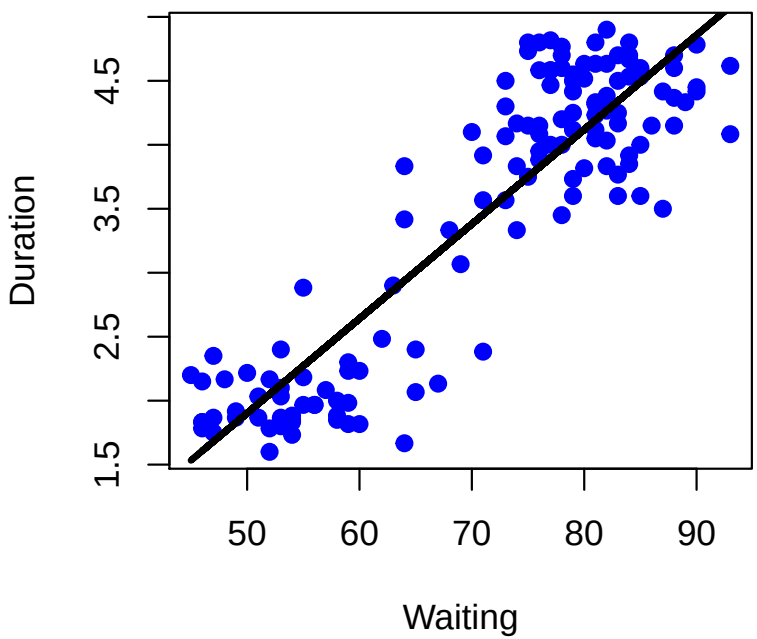
```
## (Intercept)
##    4.119307
```

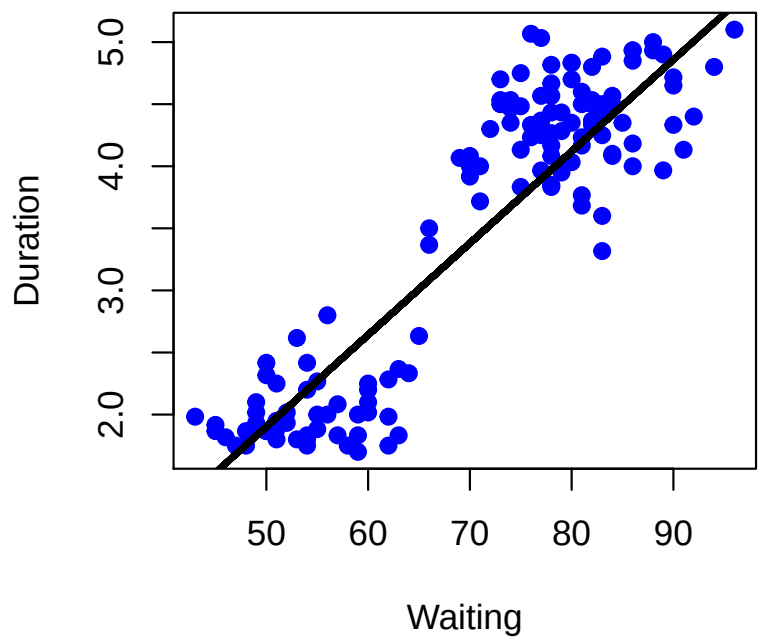
```
newdata <- data.frame(waiting=80)
predict(lm1,newdata)
```

```
##        1
## 4.119307
```

Plot predictions - training and test

```
par(mfrow=c(1,2))
plot(trainFaith$waiting,trainFaith$eruptions,pch=19,col="bl
lines(trainFaith$waiting,predict(lm1),lwd=3)
plot(testFaith$waiting,testFaith$eruptions,pch=19,col="blue
lines(testFaith$waiting,predict(lm1,newdata=testFaith),lwd=
```





Get training set/test set errors

```
# Calculate RMSE on training
sqrt(sum((lm1$fitted-trainFaith$eruptions)^2))
```

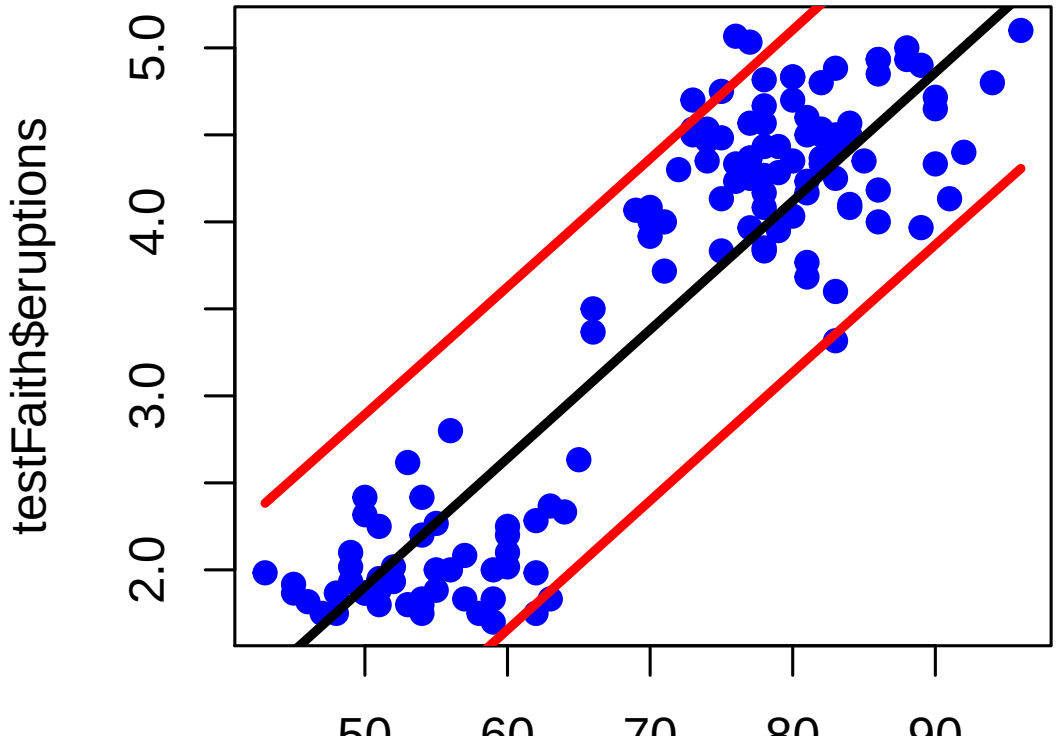
```
## [1] 5.75186
```

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

```
## [1] 5.838559
```

Prediction intervals

```
pred1 <- predict(lm1,newdata=testFaith,interval="predictio
ord <- order(testFaith$waiting)
plot(testFaith$waiting,testFaith$eruptions,pch=19,col="blue
matlines(testFaith$waiting[ord],pred1[ord,],type="l",,col=c
```



Same process with caret

```
modFit <- train(eruptions ~ waiting,data=trainFaith,method=
summary(modFit$finalModel)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26990 -0.34789  0.03979  0.36589  1.05020
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.792739   0.227869  -7.867 1.04e-12 ***
## waiting      0.073901   0.003148  23.474  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
##
```

Notes and further reading

- Regression models with multiple covariates can be included
- Often useful in combination with other models
- Elements of statistical learning
- Modern applied statistics with S
- Introduction to statistical learning