ST625 Chapter 3 Example using R

Ryan

September 7, 2015

3.9 A complete example

Step1. Model is $y = \beta_0 + \beta_1 x + \epsilon$

```
formula = formula('damageY~distanceX')
```

Step2. get the data

fire damage data

```
fire <- cbind.data.frame(
        distanceX = c(3.4,1.8,4.6,2.3,3.1,5.5,.7,3.0,
                      2.6,4.3,2.1,1.1,6.1,4.8,3.8),
        damageY = c(26.2,17.8,31.3,23.1,27.5,36.0,14.1,
                    22.3,19.6,31.3,24.0,17.3,43.2,36.4,26.1))
```

Step3 fit the model using the data

model fitting using lm

```
fireModel <- lm(formula, data = fire)
fireModel$coefficients
```

```
## (Intercept)   distanceX
##   10.277929    4.919331
```

model fitting using formula

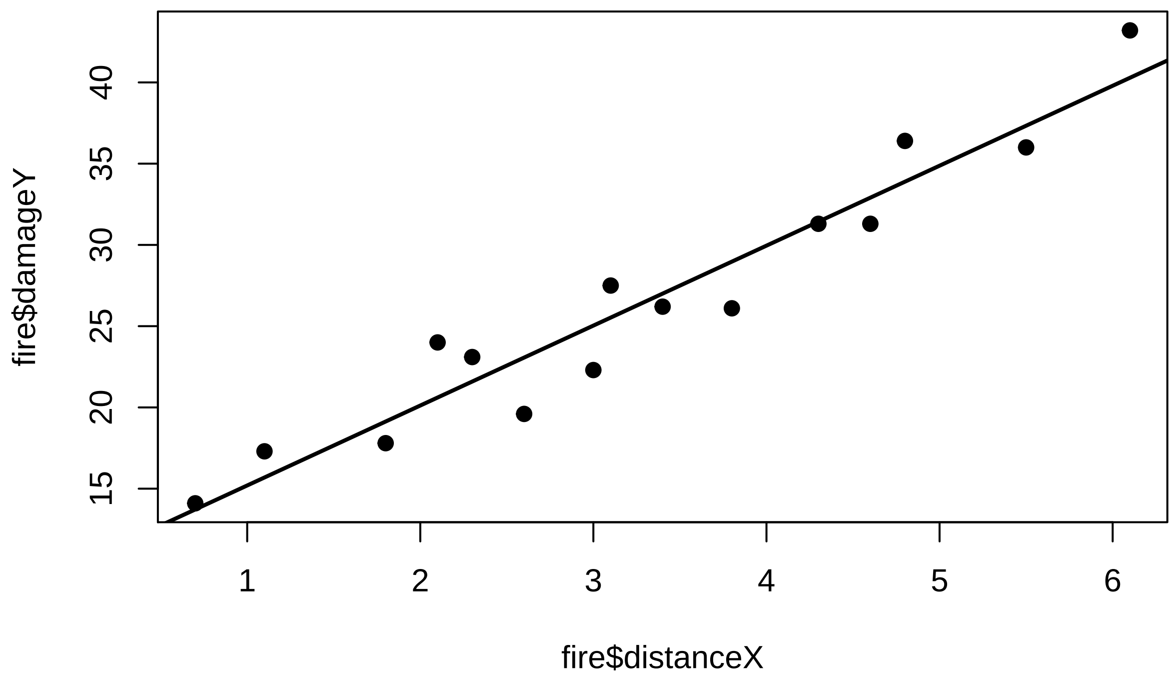
```
SSxx = sum((fire$distanceX-mean(fire$distanceX))^2)
SSxy = sum((fire$distanceX-mean(fire$distanceX))*(fire$damageY-mean(fire$damageY)))
b1 = SSxy/SSxx
b0 = mean(fire$damageY) - b1 * mean(fire$distanceX)
b0;b1
```

```
## [1] 10.27793
```

```
## [1] 4.919331
```

scatter plot with the regression line

```r
plot(x = fire$distanceX, y = fire$damageY, pch = 19)
abline(fireModel, lwd = 2)
```



Step4. Check assumptions on random error

1 $\mathbb{E}(\epsilon) = 0$

```r
mean(fireModel$residuals)
```

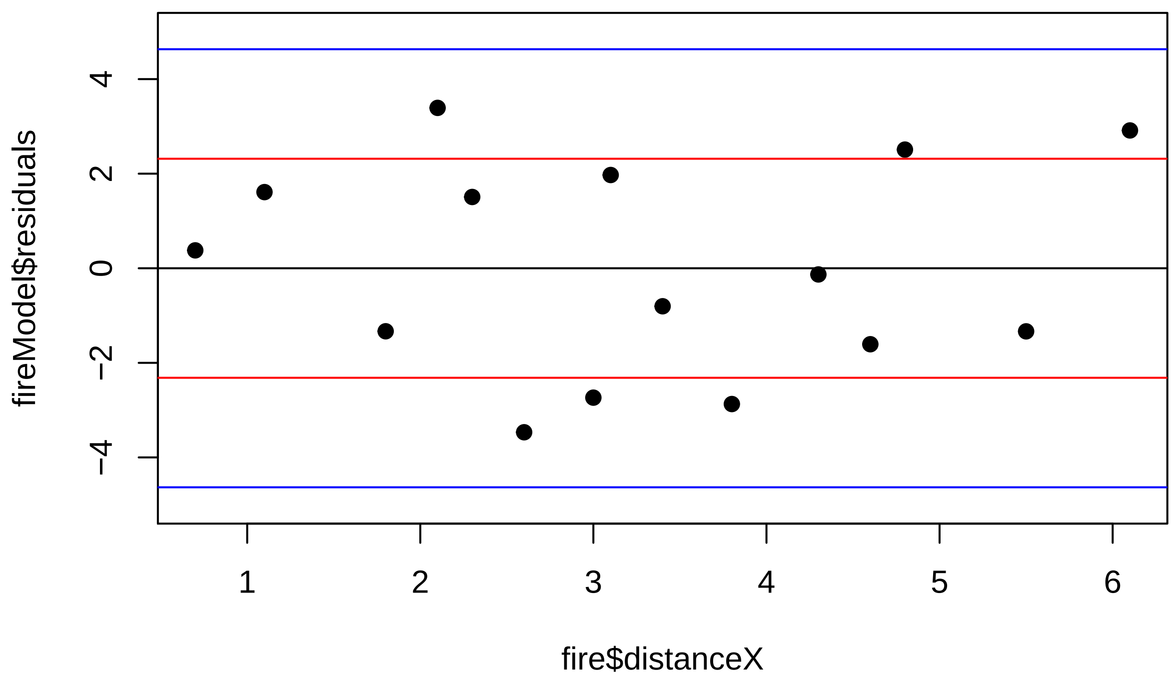
```
## [1] 4.070818e-17
```

2 Var(ϵ) is constant

```r
MSE = sum(fireModel$residuals^2)/fireModel$df.residual
MSE
```

```
## [1] 5.36546
```

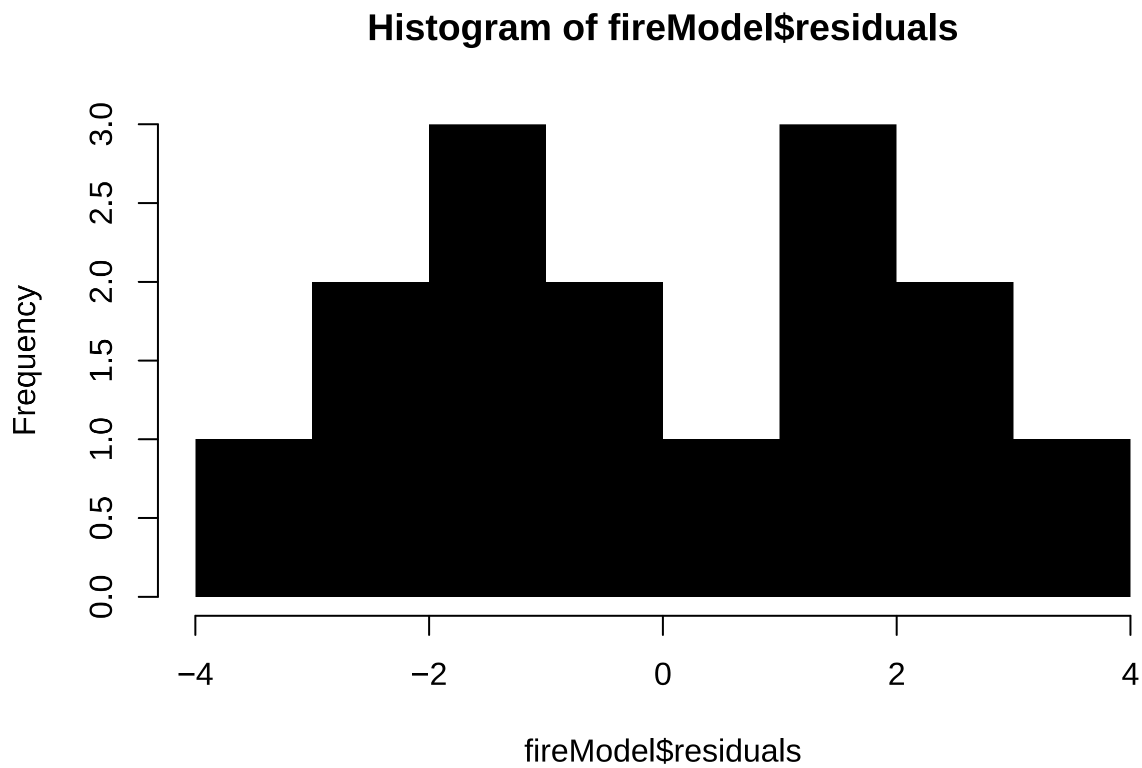
```r
RMSE = sqrt(MSE)
plot(fire$distanceX, fireModel$residuals, pch = 19, ylim = c(-5,5))
abline(0,0)
abline(2.316,0, col = 'red')
abline(-2.316,0, col = 'red')
abline(2 * 2.316,0, col = 'blue')
abline(2 * -2.316,0, col = 'blue')
```



3 ϵ normally distributed

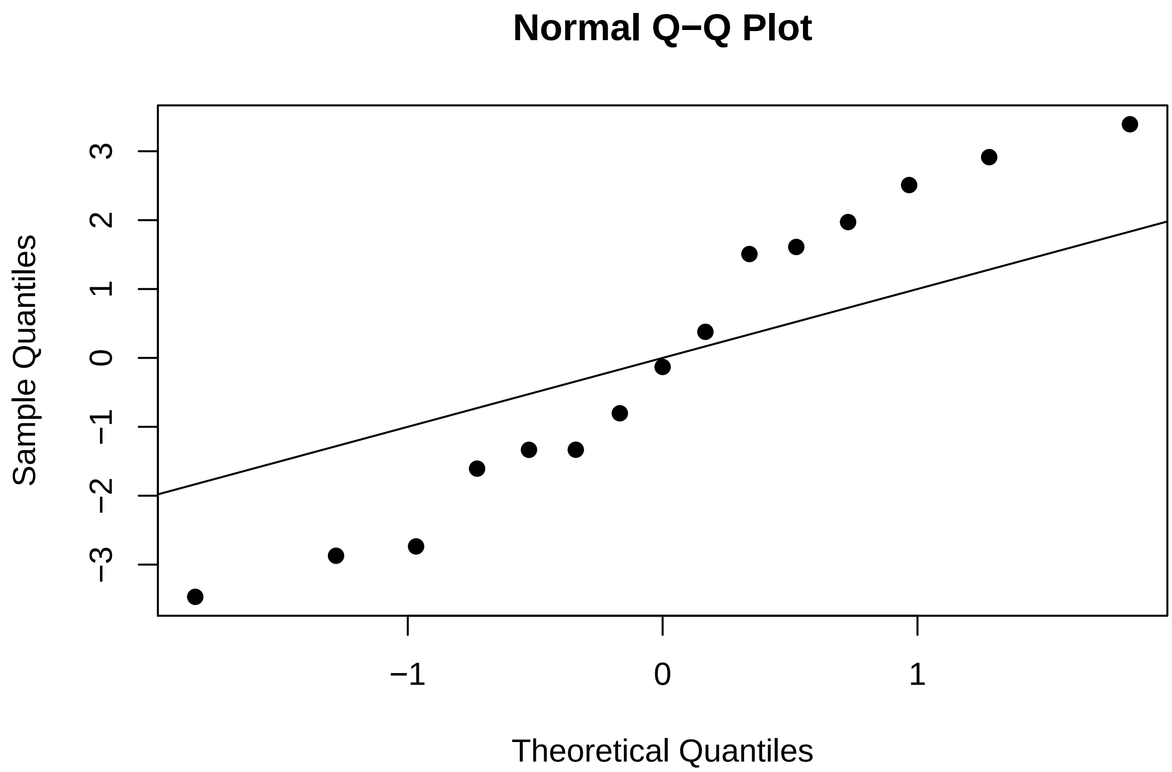
using histogram

```r
hist(fireModel$residuals, border = F, col = 'black')
```



using qq plot

```
qqnorm(fireModel$residuals, pch = 19)
abline(0,1)
```



coefficient of variation

```r
100*RMSE/mean(fire$damageY)
```

```
## [1] 8.769609
```

Summary information

```r
summary(fireModel)
```

```
##
## Call:
## lm(formula = formula, data = fire)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4682 -1.4705 -0.1311  1.7915  3.3915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2779     1.4203   7.237 6.59e-06 ***
## distanceX     4.9193     0.3927  12.525 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.316 on 13 degrees of freedom
## Multiple R-squared:  0.9235, Adjusted R-squared:  0.9176
## F-statistic: 156.9 on 1 and 13 DF,  p-value: 1.248e-08
```

Step5. Assess model adequacy

a Test of model utility

```
SE = RMSE/sqrt(SSxx)
t = b1/SE
2*pt(-1*abs(t), df = fireModel$df.residual)
```

```
## [1] 1.2478e-08
```

b Confidence interval for slope

using the R function

```
confint(fireModel)
```

```
##                 2.5 %    97.5 %
## (Intercept) 7.209605 13.346252
## distanceX   4.070851  5.767811
```

hand calculation

```
b1 + qt(0.025, df = fireModel$df.residual) * SE * c(1,-1)
```

```
## [1] 4.070851 5.767811
```

c Numerical descriptive measures of model adequacy

R Squared calculated by hand

```
SSE = sum(fireModel$residuals^2)
SSyy = sum((fire$damageY-mean(fire$damageY))^2)
RSquare = 1- SSE/SSyy
RSquare
```

```
## [1] 0.9234782
```

```
summary(fireModel)
```

```
##
## Call:
## lm(formula = formula, data = fire)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4682 -1.4705 -0.1311  1.7915  3.3915
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2779     1.4203   7.237 6.59e-06 ***
## distanceX     4.9193     0.3927  12.525 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.316 on 13 degrees of freedom
## Multiple R-squared:  0.9235, Adjusted R-squared:  0.9176
## F-statistic: 156.9 on 1 and 13 DF,  p-value: 1.248e-08
```

Coefficient of correlation by hand and using function

```
cor(fire$damageY, fire$distanceX)
```

```
## [1] 0.9609777
```

```
SSxy/sqrt(SSxx*SSyy)
```

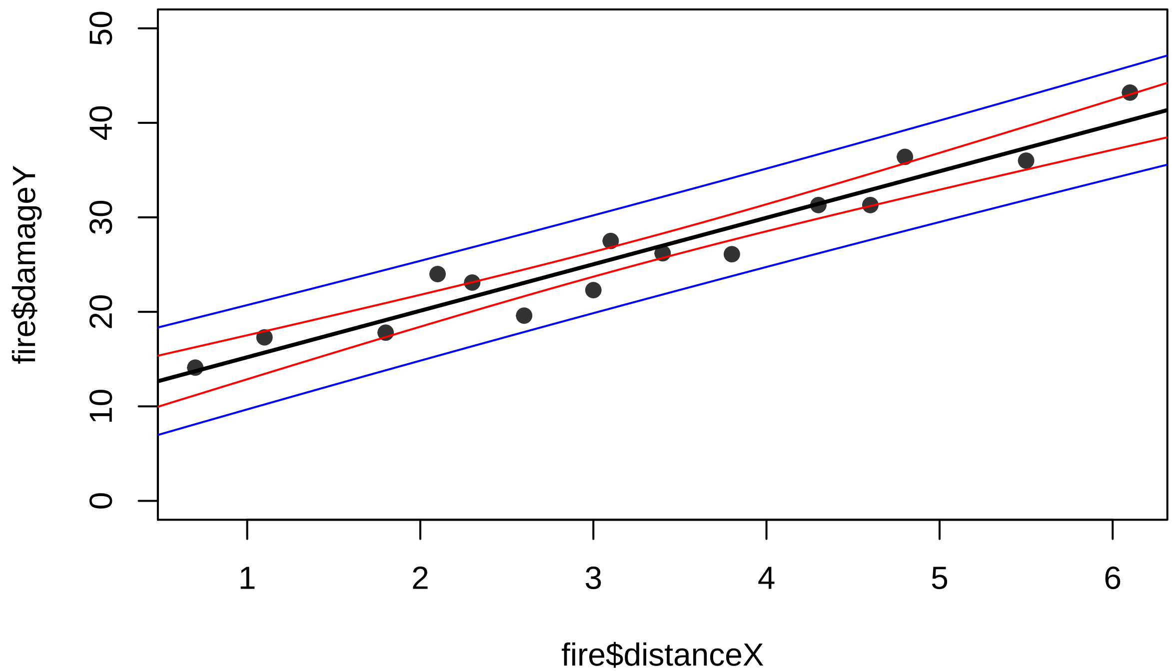
```
## [1] 0.9609777
```

Step6. Prediction

ignore the tideous calculation

```
linspace = seq(0,7,0.01)
confInterval <- predict(fireModel,
                        newdata = data.frame(distanceX = linspace),
                        interval = "confidence")
predInterval <- predict(fireModel,
                        newdata = data.frame(distanceX = linspace),
                        interval = "prediction")

plot(x = fire$distanceX,
     y = fire$damageY,
     pch = 19,
     ylim=c(0,50),
     col = adjustcolor('black', alpha.f =  0.8))
abline(fireModel, lwd = 2)
points(x= linspace, y = confInterval[,2], type = 'l', col = 'red')
points(x= linspace, y = confInterval[,3], type = 'l', col = 'red')
points(x= linspace, y = predInterval[,2], type = 'l', col = 'blue')
points(x= linspace, y = predInterval[,3], type = 'l', col = 'blue')
```



Make perdiction on perticular value

```r
predict(fireModel,
        newdata = data.frame(distanceX = 3.5),
        interval = "prediction")
```

```
##        fit      lwr      upr
## 1 27.49559 22.32394 32.66723
```