Regression Modeling

Jack Conway

5/4/2017

Introduction

Our project will be on the use of R for various types of regression analysis. We have been exposed to the use of simple linear regression in R, and now we will explore how to utilize existing R packages and functions to perform polynomial regression, piecewise regression, and splines. Within each type of regression we will compare different models and select the best one for each. Then, the goal is to compare the best model from each category and choose which model is the best to use for the particular data.

Explanation of Statistical Concepts

Polynomial Regression - Polynomial regression looks at different explanatory variables and whether there is a relationship between the explanatory variable, to some degree such as squared, cubed, etc. and tests whether this approach better predicts the data than the linear model.
Piecewise Regression - Piecewise Regression is picking specific intervals to have their own linear regression model slope applied to that interval. This is useful when the data exhibits different relationships in the different regions.
Splines - A spline is a function that is constructed piecewise from polynomial functions, which gives a smooth connection between the different pieces.

Sources

We will be using the a data set on gun related deaths in the US. for the second run through of the procedure. This data set compares the percentage of households that own guns in a state vs. the gun death rate per 100,000 people for 43 states.This was an observational study.

We also used handouts and our Applied Linear Regression Models edition 4 by Kutner et al. from our class..

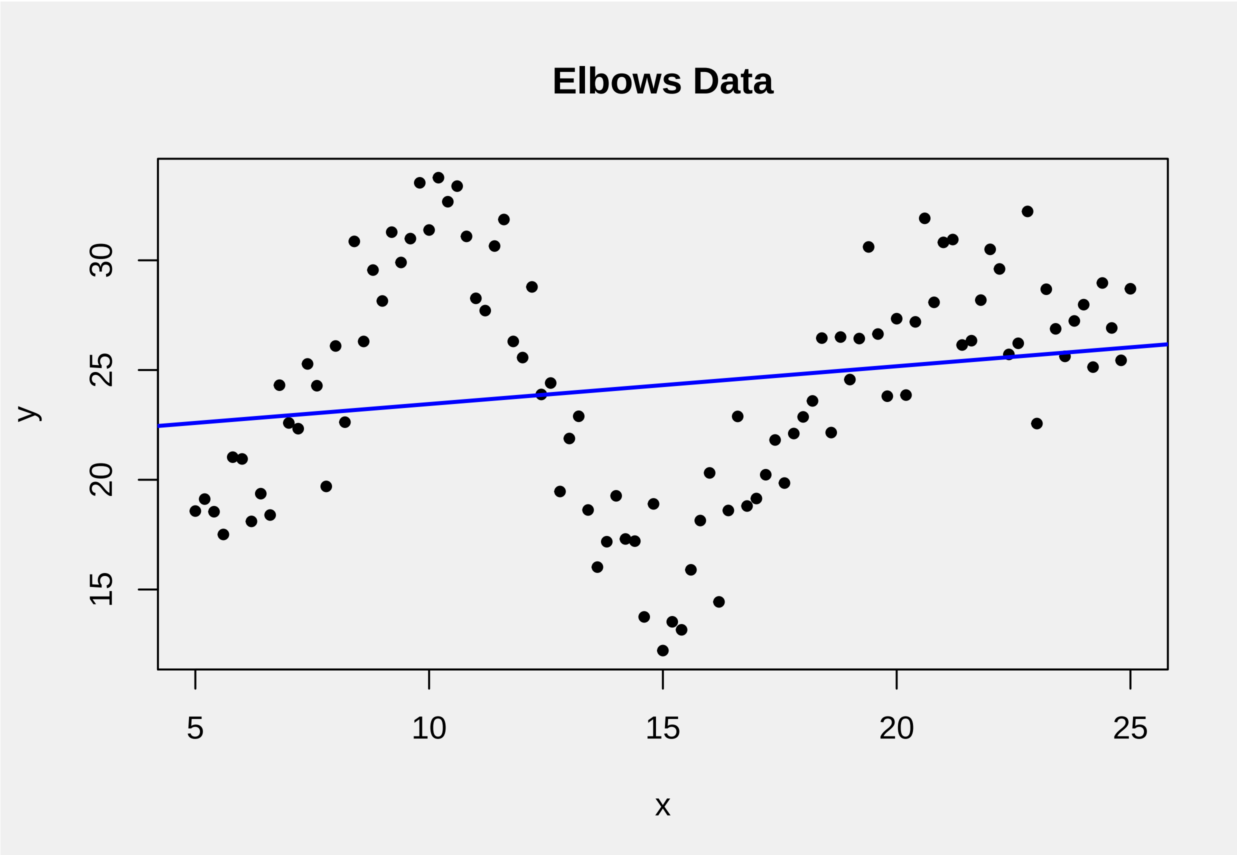
r-bloggers/splines

```
# Phase 3
#setwd("/Users/jackconway/Documents/STAT 300's/STAT 331")



#### (i) Polynomial Regression ####
#install.packages("ggplot2")
library(ggplot2)
```

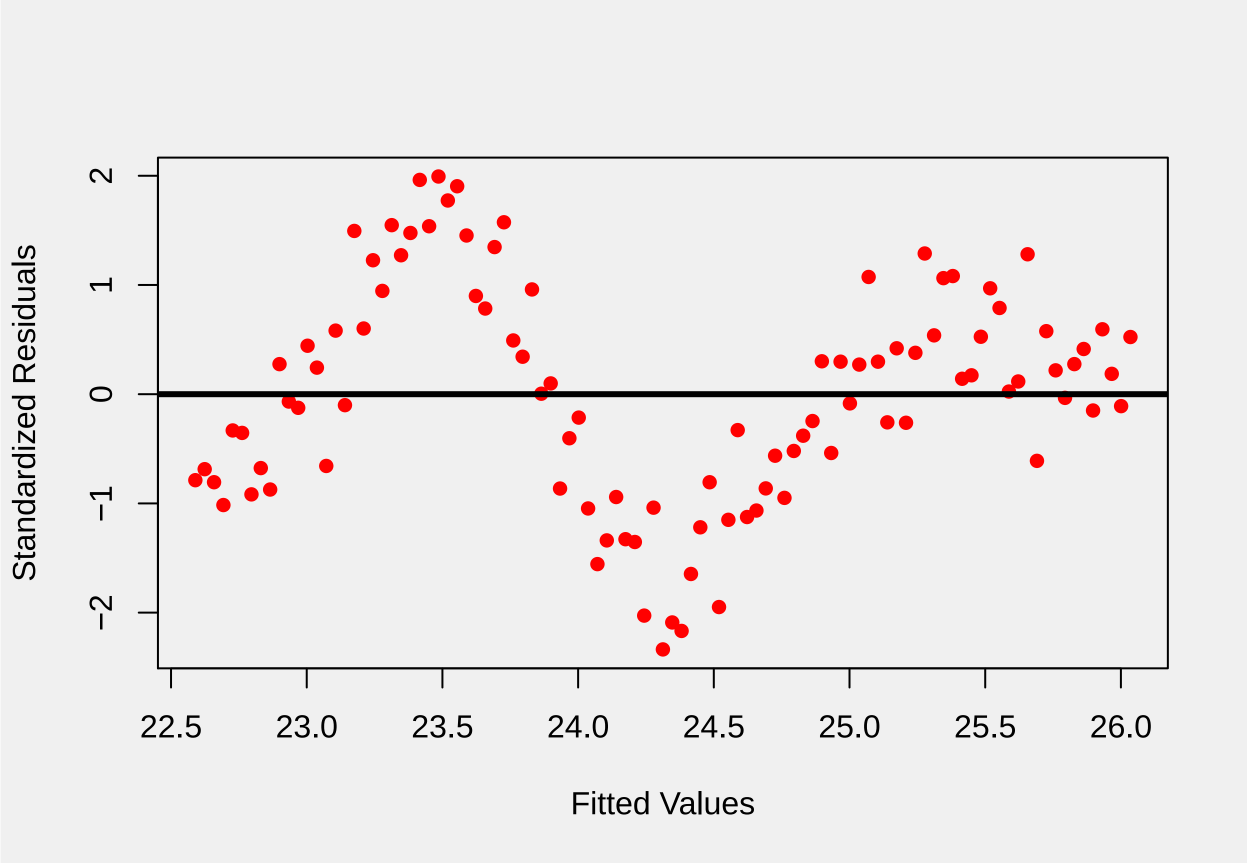
```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
par(bg = "gray94")


elbow <- read.table("/Users/jackconway/Documents/STAT 300's/STAT 331/ElbowData.txt", header = TRUE)
plot.new()
```

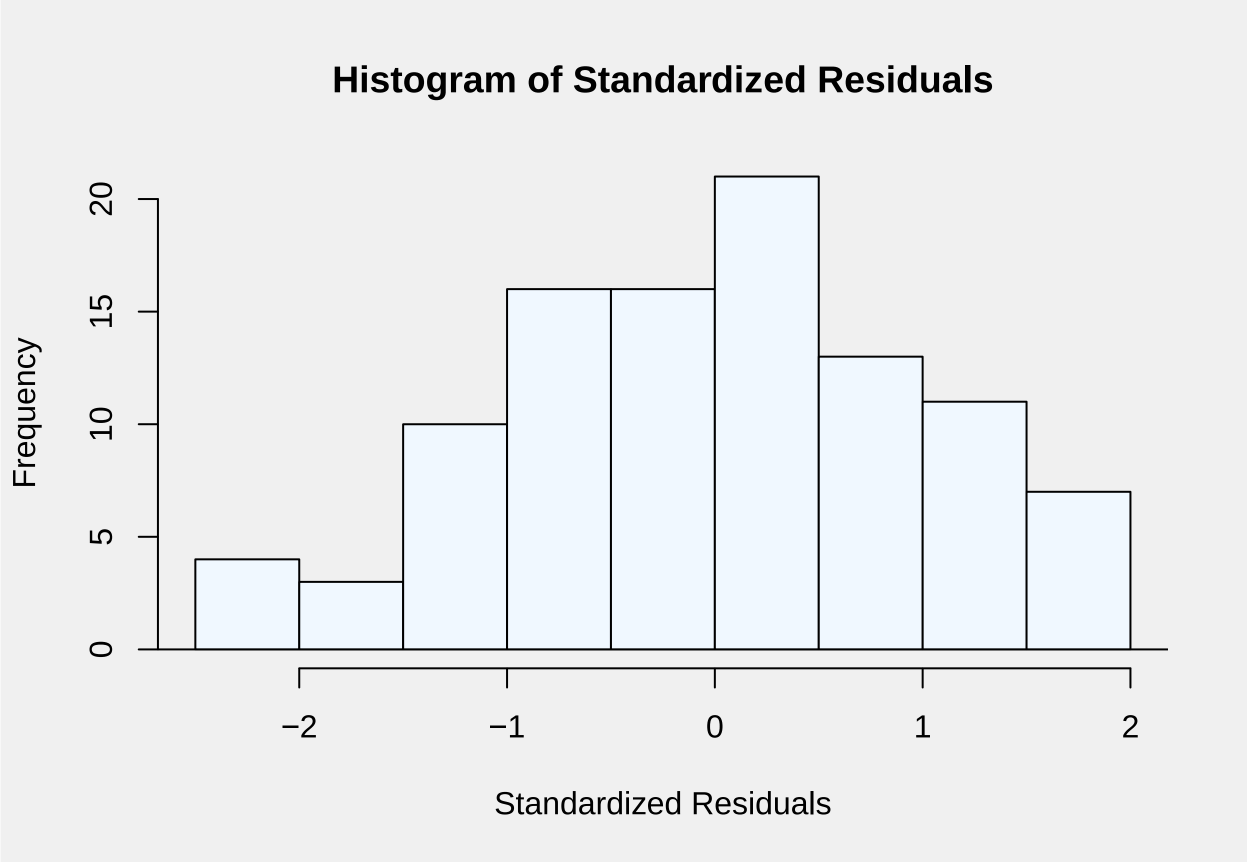
```r
plot(elbow$y ~ elbow$x, main = "Elbows Data", ylab = "y", xlab = "x", pch = 20)

elbowSimpleLinear <- lm(elbow$y ~ elbow$x)
abline(elbowSimpleLinear, col = "blue", lwd = "2")        # simple linear regression model
```




```r
stanResidSL <- rstandard(elbowSimpleLinear)    # saves standardized resids for simple linear model

plot.new()
plot(stanResidSL ~ elbowSimpleLinear$fitted.values,
     data = elbow,
     ylab = "Standardized Residuals",
     xlab="Fitted Values",
     pch = 16,
     col = "red")

abline(h=0, lwd = 3)
```



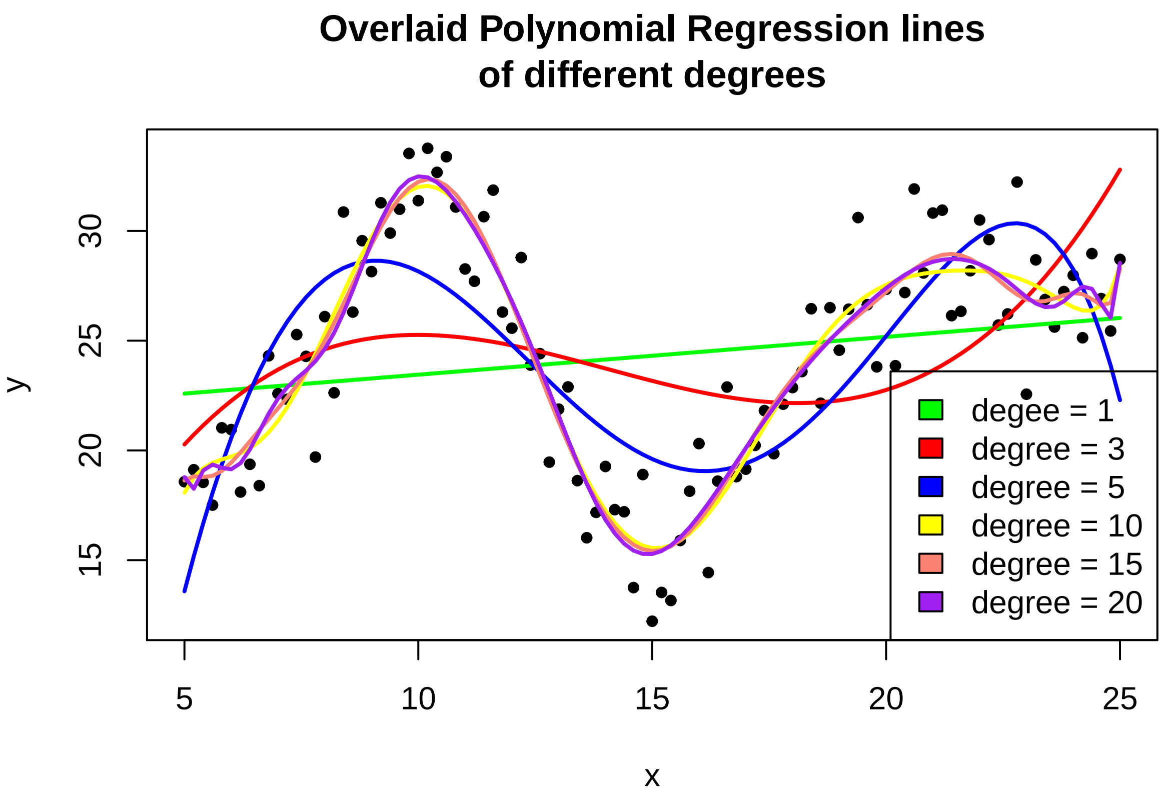

```r
hist(stanResidSL, xlab="Standardized Residuals",
     main="Histogram of Standardized Residuals",
     col = "aliceblue")

abline(h=0)
```



The simple linear regression model doesn't fit the data well. There is a clear violation in the linearity assumption when checking the standardized residuals vs fitted values.

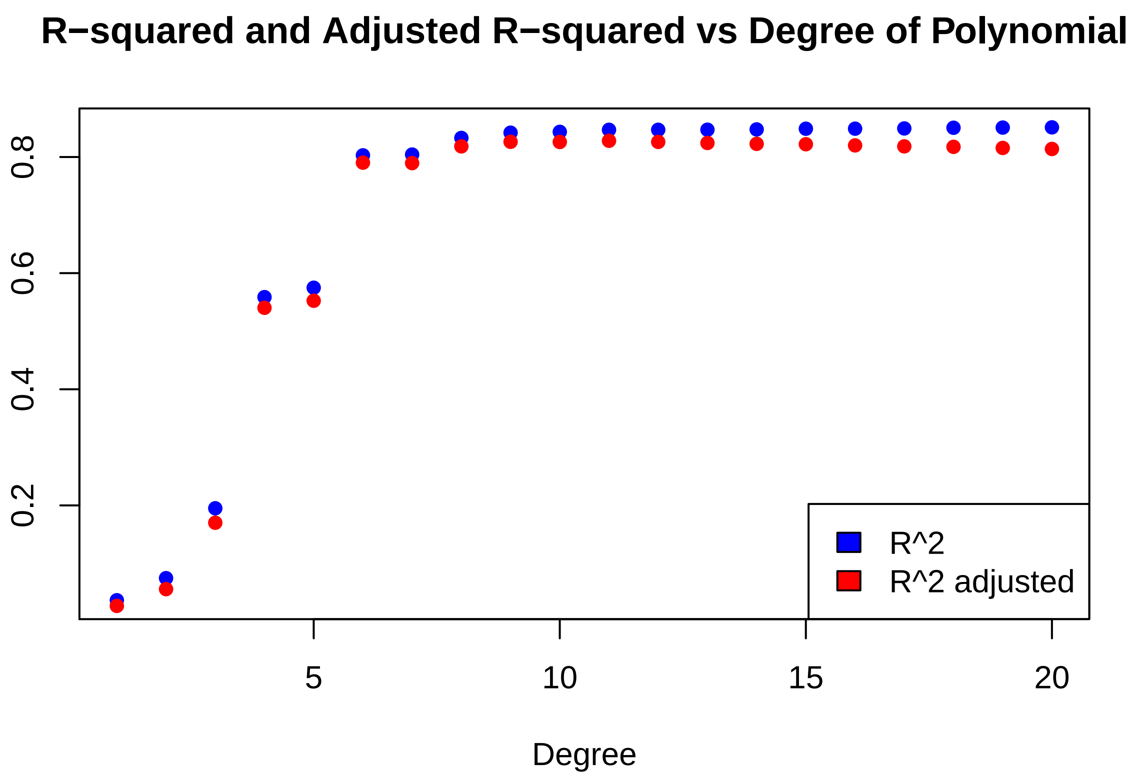
```
polynomial1 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 1))  # polynomial models of degrees 1-20
polynomial2 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 2))
polynomial3 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 3))
polynomial4 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 4))
polynomial5 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 5))
polynomial6 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 6))
polynomial7 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 7))
polynomial8 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 8))
polynomial9 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 9))
polynomial10 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 10))
polynomial11 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 11))
polynomial12 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 12))
polynomial13 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 13))
polynomial14 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 14))
polynomial15 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 15))
polynomial16 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 16))
polynomial17 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 17))
polynomial18 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 18))
polynomial19 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 19))
polynomial20 <- lm(formula = elbow$y ~ poly(elbow$x, degree = 20))

x <- elbow$x
y <- elbow$y
```

```
plot.new()
plot(y ~ x, main = "Overlaid Polynomial Regression lines\nof different degrees", ylab = "y", xlab = "x"
  lines(x, polynomial1$fitted.values, col = "green", lwd = 2)
  lines(x, polynomial3$fitted.values, col = "red", lwd = 2)
  lines(x, polynomial5$fitted.values, col = "blue", lwd = 2)
  lines(x, polynomial10$fitted.values, col = "yellow", lwd = 2)
  lines(x, polynomial15$fitted.values, col = "salmon", lwd = 2)
  lines(x, polynomial20$fitted.values, col = "purple", lwd = 2)
  legend("bottomright", legend = c("degee = 1","degree = 3", "degree = 5", "degree = 10", "degree = 15"
         fill = c("green", "red", "blue", "yellow", "salmon", "purple"))
```




```
# plot of r-squared and r-squared adj. values for different degrees
plot.new()
plot(y = c(
  summary(polynomial1)$r.squared,
  summary(polynomial2)$r.squared,
  summary(polynomial3)$r.squared,
  summary(polynomial4)$r.squared,
  summary(polynomial5)$r.squared,
  summary(polynomial6)$r.squared,
  summary(polynomial7)$r.squared,
  summary(polynomial8)$r.squared,
  summary(polynomial9)$r.squared,
  summary(polynomial10)$r.squared,
  summary(polynomial11)$r.squared,
  summary(polynomial12)$r.squared,
  summary(polynomial13)$r.squared,
  summary(polynomial14)$r.squared,
  summary(polynomial15)$r.squared,
```

```r
  summary(polynomial16)$r.squared,
  summary(polynomial17)$r.squared,
  summary(polynomial18)$r.squared,
  summary(polynomial19)$r.squared,
  summary(polynomial20)$r.squared),
  x = 1:20, ylab = "",
  xlab = "Degree",
  main ="R-squared and Adjusted R-squared vs Degree of Polynomial",
  pch = 16, col = "blue")


points(x = 1:20, y = c(summary(polynomial1)$adj.r.squared,
                       summary(polynomial2)$adj.r.squared,
                       summary(polynomial3)$adj.r.squared,
                       summary(polynomial4)$adj.r.squared,
                       summary(polynomial5)$adj.r.squared,
                       summary(polynomial6)$adj.r.squared,
                       summary(polynomial7)$adj.r.squared,
                       summary(polynomial8)$adj.r.squared,
                       summary(polynomial9)$adj.r.squared,
                       summary(polynomial10)$adj.r.squared,
                       summary(polynomial11)$adj.r.squared,
                       summary(polynomial12)$adj.r.squared,
                       summary(polynomial13)$adj.r.squared,
                       summary(polynomial14)$adj.r.squared,
                       summary(polynomial15)$adj.r.squared,
                       summary(polynomial16)$adj.r.squared,
                       summary(polynomial17)$adj.r.squared,
                       summary(polynomial18)$adj.r.squared,
                       summary(polynomial19)$adj.r.squared,
                       summary(polynomial20)$adj.r.squared),
                         col = "red", pch = 16)
       legend("bottomright", legend = c("R^2", "R^2 adjusted"),
       fill = c("blue", "red"))
```



(ii) Piecewise Regression

```
#install.packages("segmented")
library(segmented)

x <- elbow$x
y <- elbow$y
m <- lm(y ~ x)

# segm <- segmented(m, seg.Z =~x, c(10, 15, 20))      # partial regression model, 1st order
# plot(x, y, pch = 20, main = "Piecewise Regression\nOrder 1")
# lines(x, segm$fitted.values, col = "green", lwd = 2)
```

```
x2 <- elbow$x^2
m2 <- lm(y ~ x + x2)
#segm2 <- segmented(m2, seg.Z=~x+x2,
#                    psi = list(x=c(10, 15, 20),
#                    x2=c(10, 15, 20)^2))

# partial regression model, 2nd order

# plot(x, y, xlab = "x", ylab = "y",
#      main = "Piecewise Regression\nOrder 2",
#       pch = 20)

# lines(x, segm2$fitted.values, col = "red", lwd = 2)
```

```
x3 <- elbow$x^3
m3 <- lm(y ~ x + x2 + x3)
```

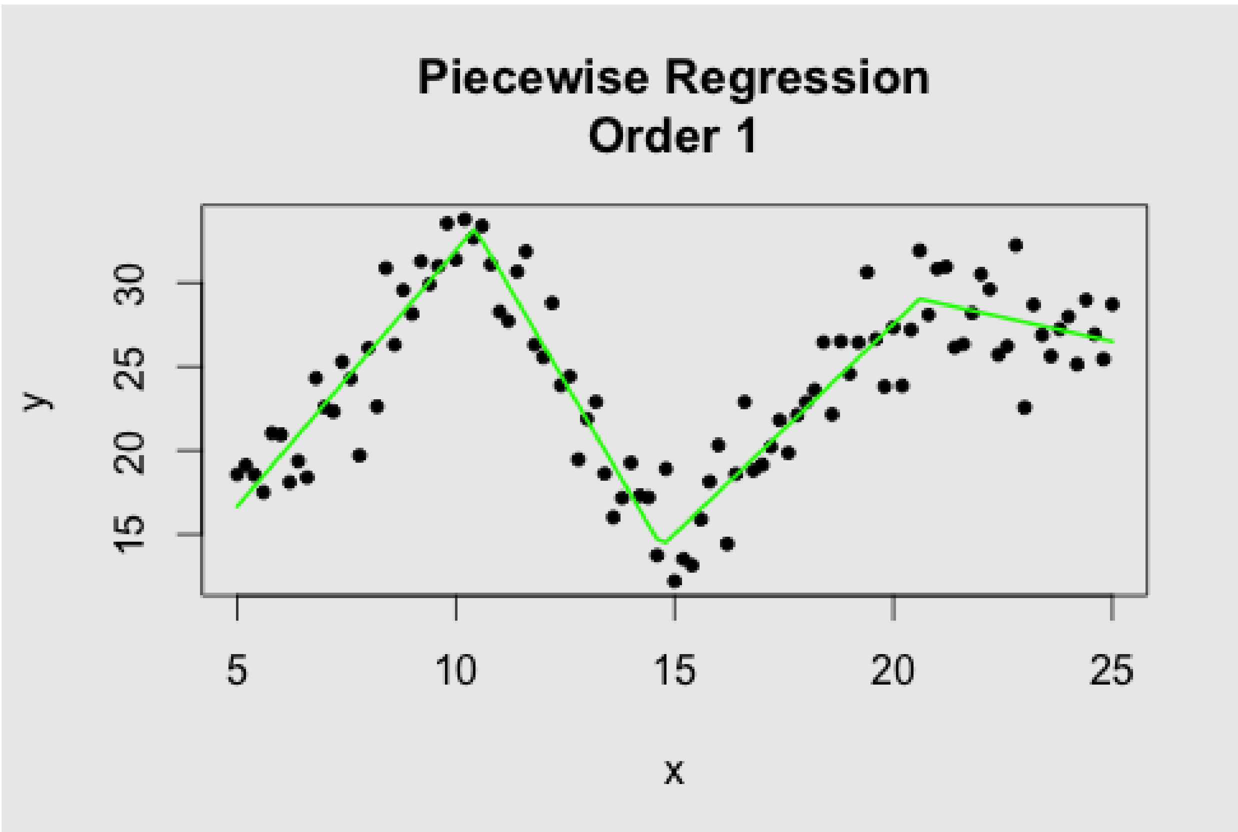


Figure 1:

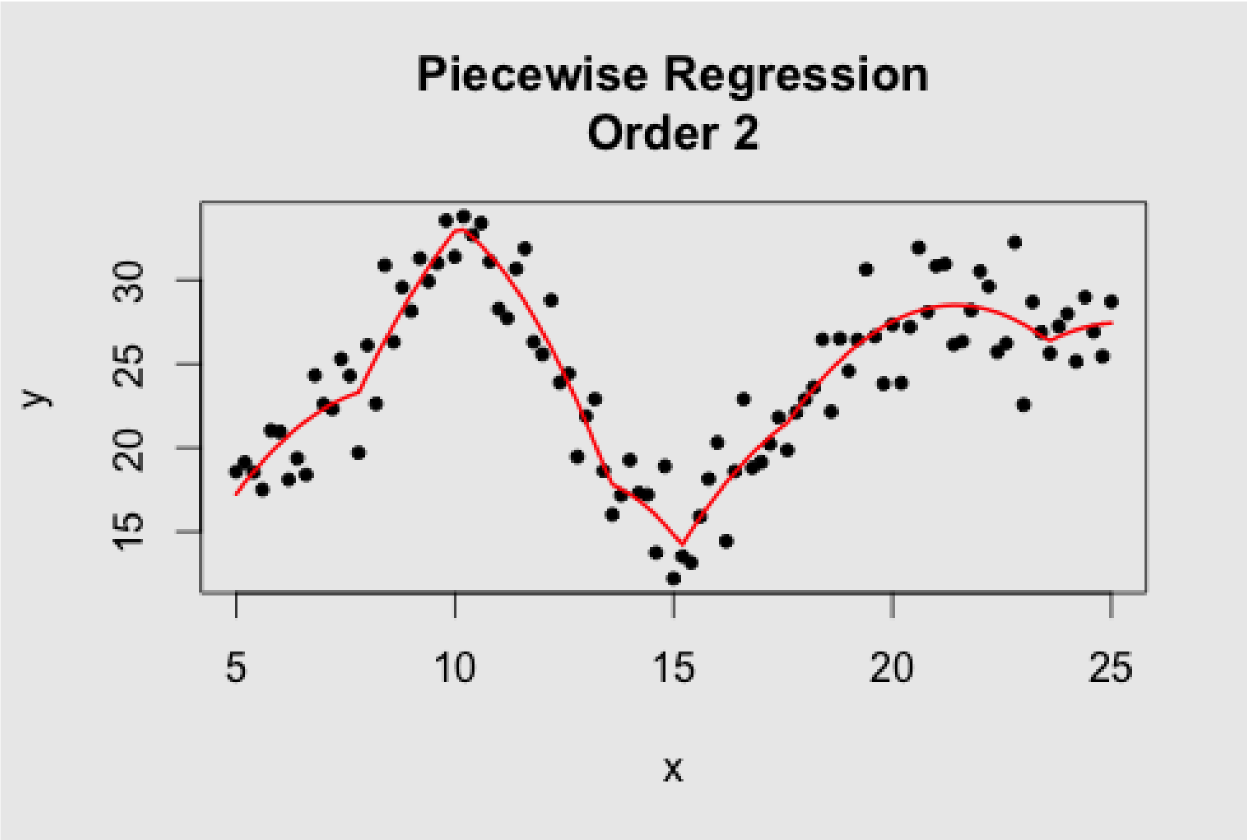


Figure 2:

```
#segm3 <- segmented(m3, seg.Z=~x+x2+x3,
#                     psi = list(x=c(10, 15, 20),
#                     x2=c(10, 15, 20)^2,
#                     x3=c(10, 15, 20)^3))  # partial regression model, 3rd order
#

#plot(x, y, xlab = "x", ylab = "y",
#     main = "Piecewise Regression\nOrder 3", pch = 20)

#lines(x, segm3$fitted.values, col = "blue", lwd = 2)

# plot(x, y, pch = 20,
#      main = "Piecewise Regressions\nof Different Orders",    #overlaid models for comparison
#      ylab = "y")

# lines(x, segm$fitted.values, col = "green", lwd = 2)
# lines(x, segm2$fitted.values, col = "red", lwd = 2)
# lines(x, segm3$fitted.values, col = "blue", lwd = 2)
# legend("bottomright", legend = c("Order 1", "Order 2", "Order 3"),
#         fill = c("green", "red", "blue"))
```

Figure 3:

```
  # plot of r-squared for piecewise regression of different orders

# plot(y = c(summary(segm)$r.squared,
```

```r
#               summary(segm2)$r.squared,
#               summary(segm3)$r.squared),
#               x = 1:3, ylab = "",
#               xlab = "Degree",
#               main ="R-Squared and Adjusted R-Squared vs Degree of Segmentation",
#               pch = 16, col = "blue",
#               ylim = c(.8, .88),
#               cex.main = 1.1)


# points(x = 1:3, y = c(summary(segm)$adj.r.squared,
#                          summary(segm2)$adj.r.squared,
#                          summary(segm3)$adj.r.squared),
#                          pch = 16, col = "red")

# legend("bottomright", legend = c("R^2", "R^2 adjusted"),
#          fill = c("blue", "red"))
```

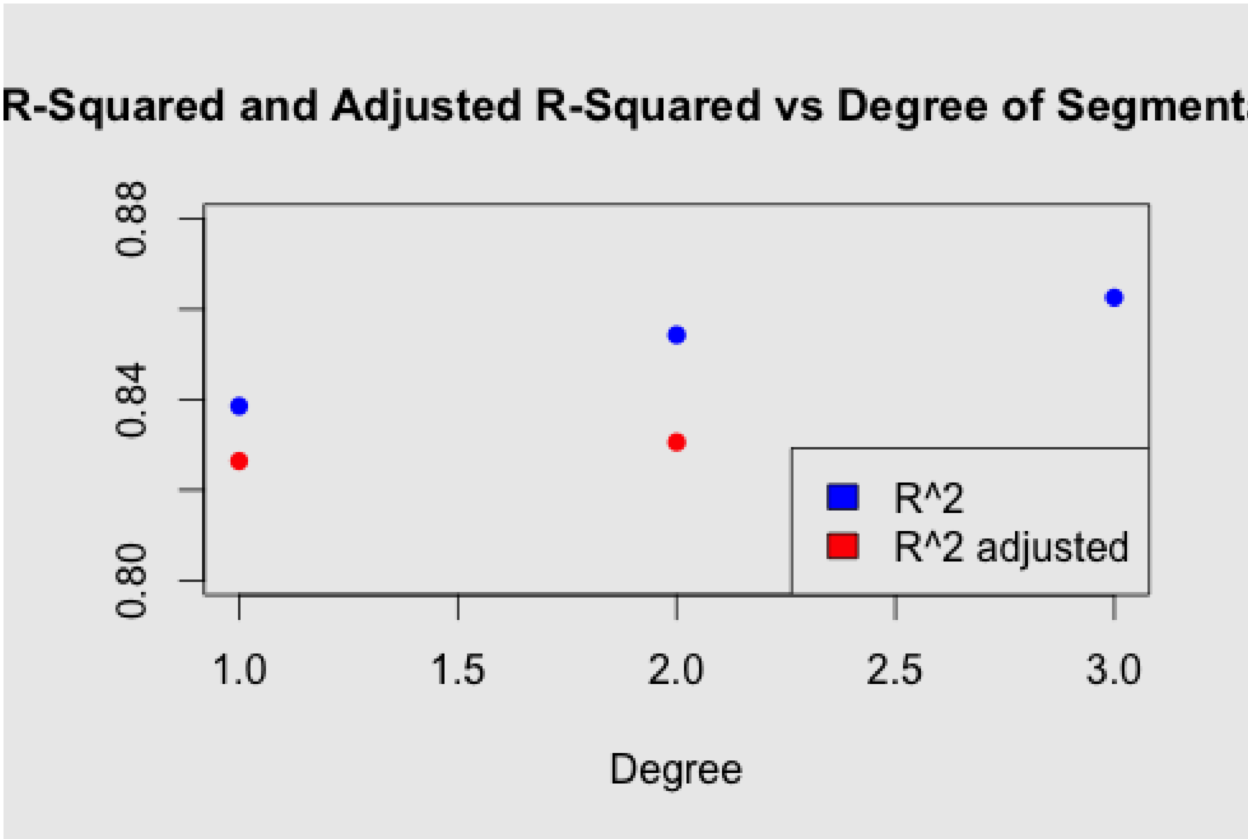
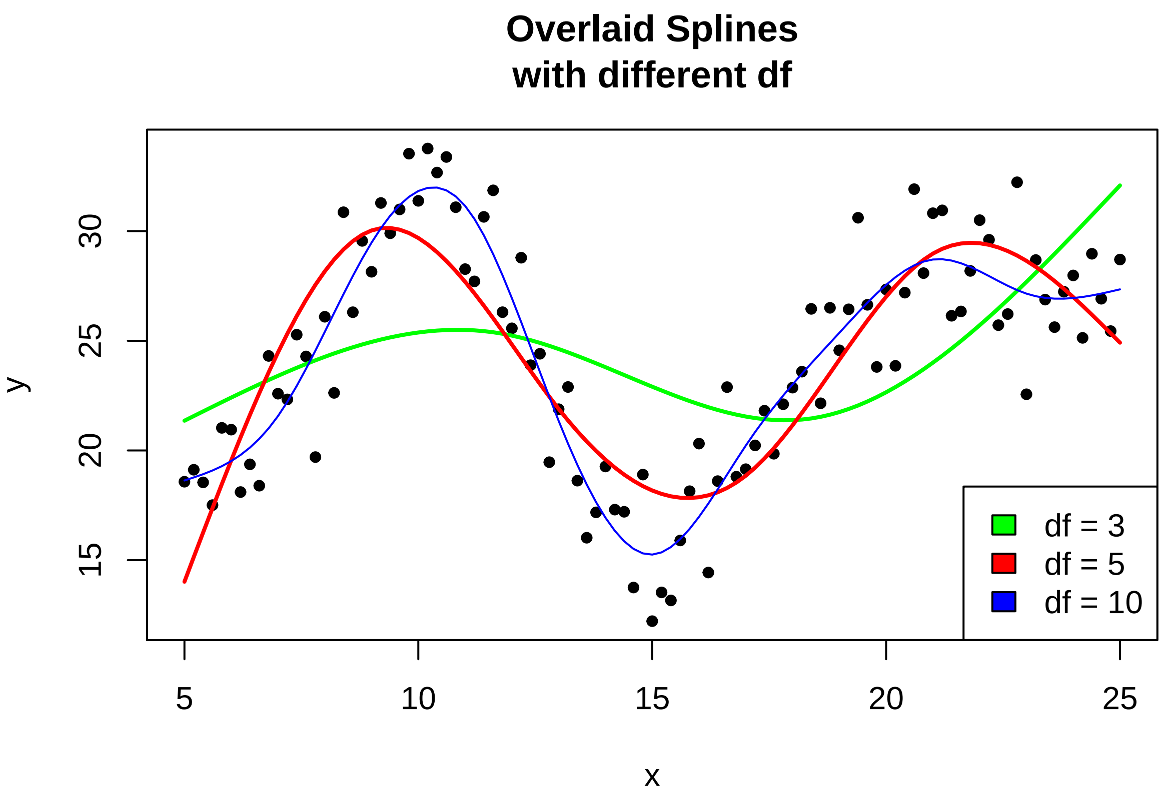


Figure 4:

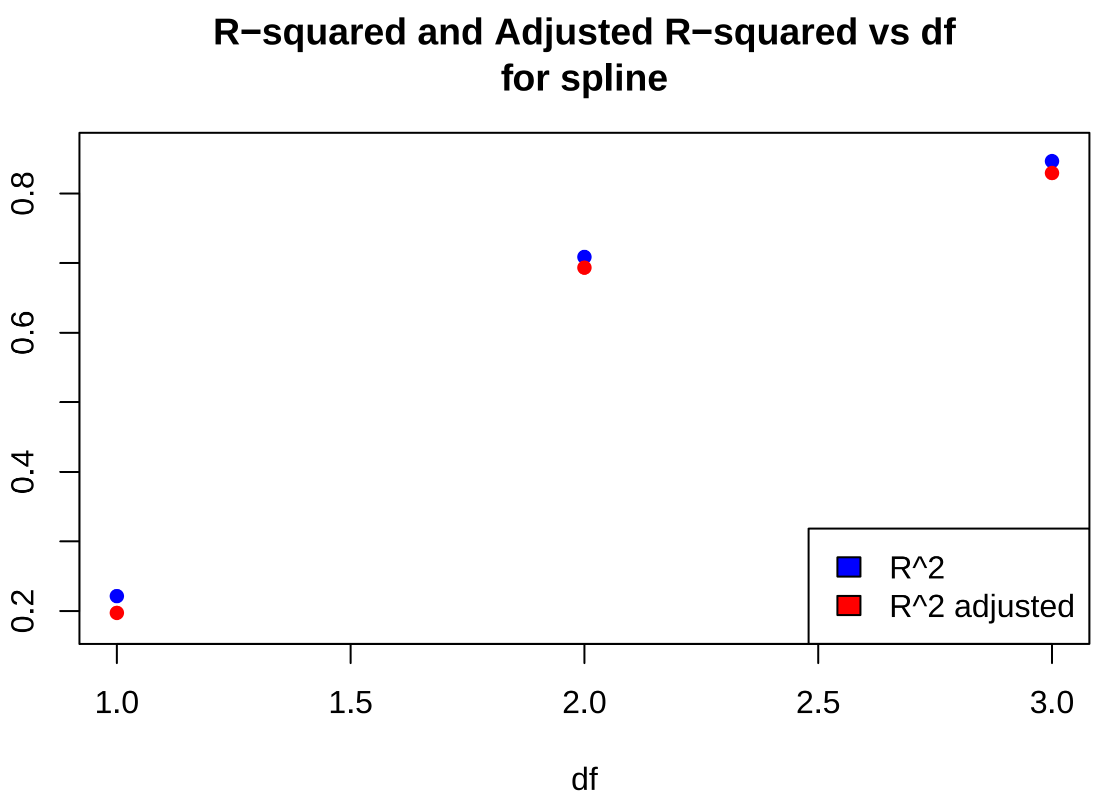
```r
####

#### (iii) Splines ####

#install.packages("splines")
library(splines)
```

```r
#install.packages("stats")
library(stats)
plot.new()

plot(elbow$y ~ elbow$x,
     main = "Overlaid Splines\nwith different df",
     ylab = "y", xlab = "x", pch = 20)


nsp1 <- ns(elbow$x, df = 3)           #fits a spline with df = 3
fm1 <- lm(y ~ nsp1, data = elbow)
ht <- seq(5, 25, length.out = 101)
lines(ht, predict(fm1, data.frame(height = ht)), col = "green", lwd = 2)

nsp2 <- ns(elbow$x, df = 5)           #fits a spline with df = 5
fm2 <- lm(y ~ nsp2, data = elbow)
lines(ht, predict(fm2, data.frame(height = ht)), col = "red", lwd = 2)

nsp3 <- ns(elbow$x, df = 10)          #fits a spline with df = 10
fm3 <- lm(y ~ nsp3, data = elbow)
lines(ht, predict(fm3, data.frame(height = ht)), col = "blue")

legend("bottomright", legend = c("df = 3", "df = 5", "df = 10"),
       fill = c("green", "red", "blue"))
```




```r
# r-squared
plot.new()
plot(y = c(summary(fm1)$r.squared,
           summary(fm2)$r.squared,
```

```r
           summary(fm3)$r.squared),
           x = 1:3,
           ylab = "",
           xlab = "df",
           main ="R-squared and Adjusted R-squared vs df\nfor spline",
           pch = 16,
           col = "blue",
           ylim = c(0.18, .86))


points(x = 1:3, y = c(summary(fm1)$adj.r.squared,
                      summary(fm2)$adj.r.squared,
                      summary(fm3)$adj.r.squared),
                      pch = 16, col = "red")

legend("bottomright",
       legend = c("R^2", "R^2 adjusted"),
       fill = c("blue", "red"))
```



Best Models Part 1

```r
# overlaid plot of best model from each section
# plot(elbow$y ~ elbow$x,
#      main = "Elbows Data",
#      ylab = "y",
#      xlab = "x",
#      pch = 20)
```

```
# lines(x, polynomial6$fitted.values, col = "blue", lwd = 2)
# lines(x, segm$fitted.values, col = "green", lwd = 2)
# lines(ht, predict(fm3, data.frame(height = ht)), col = "red", lwd = 2)
# legend("bottomright",
#         legend = c("Polynomial degree 6", "Piecewise order 1", "Spline df=10"),
#         fill = c("blue", "green", "red"))
```

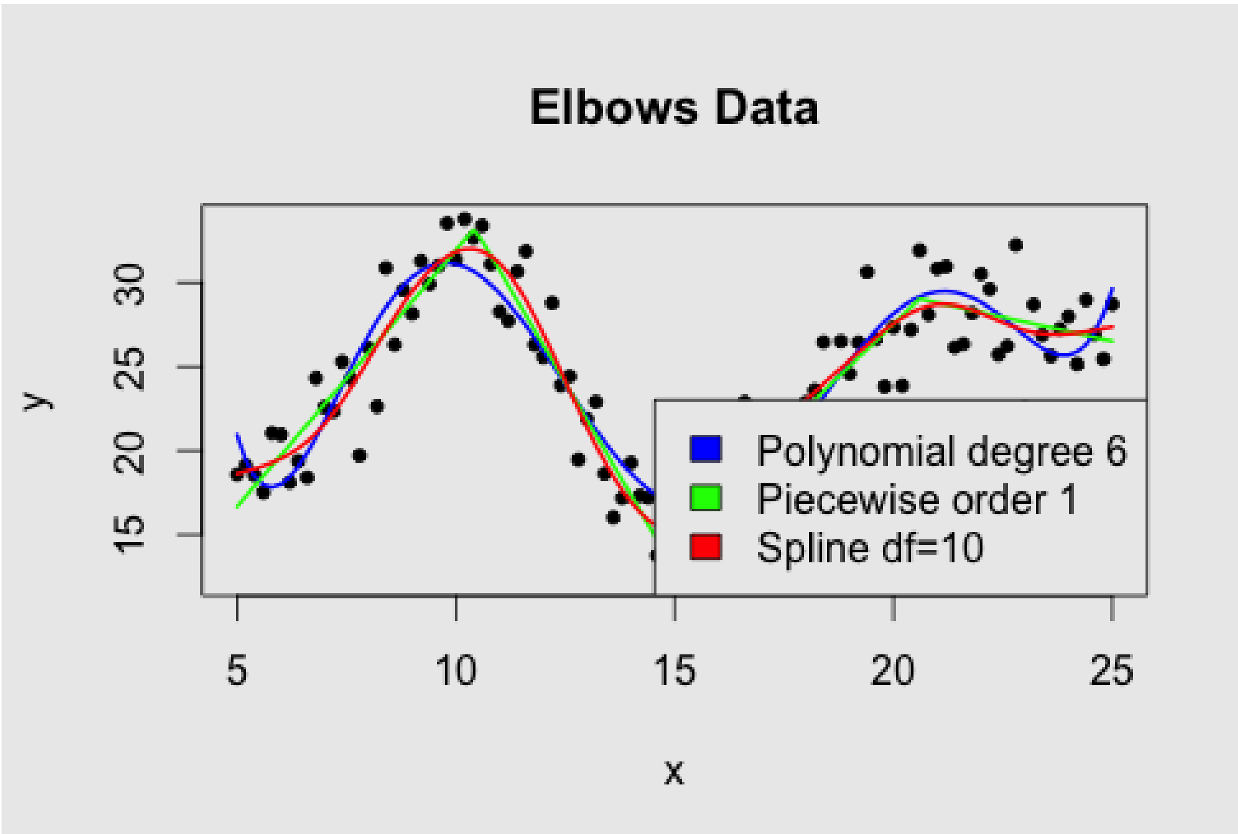
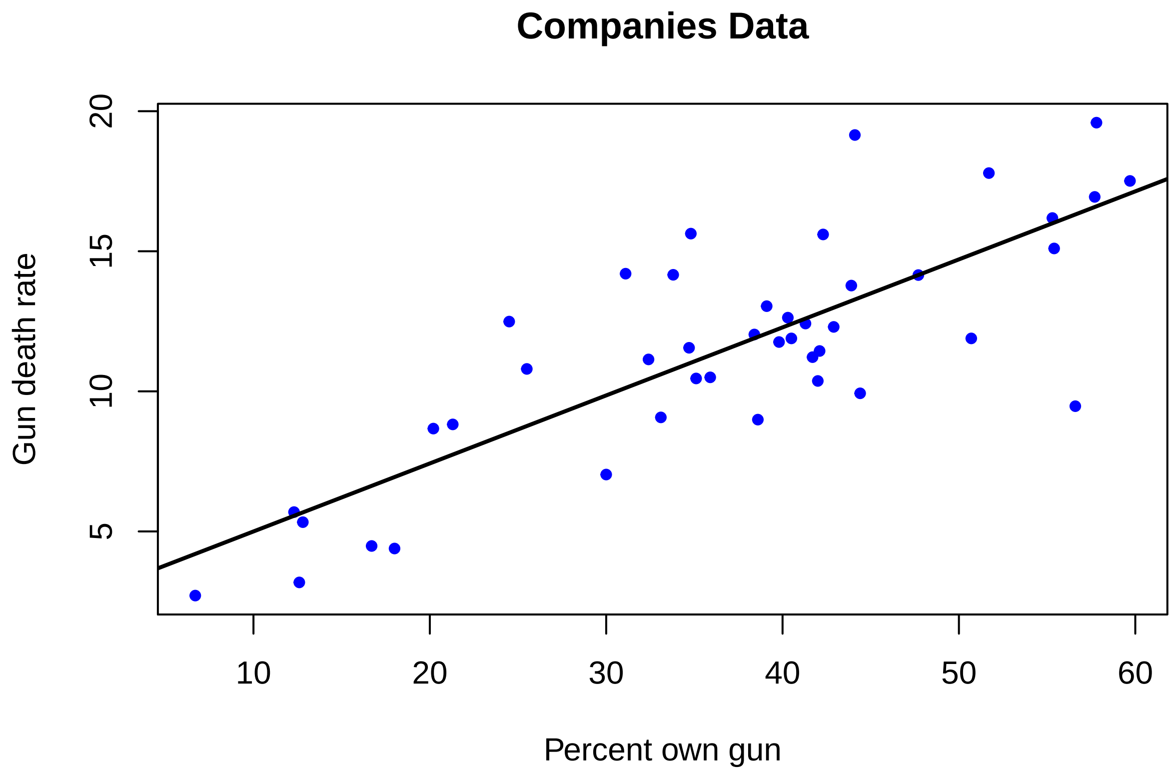


Figure 5:

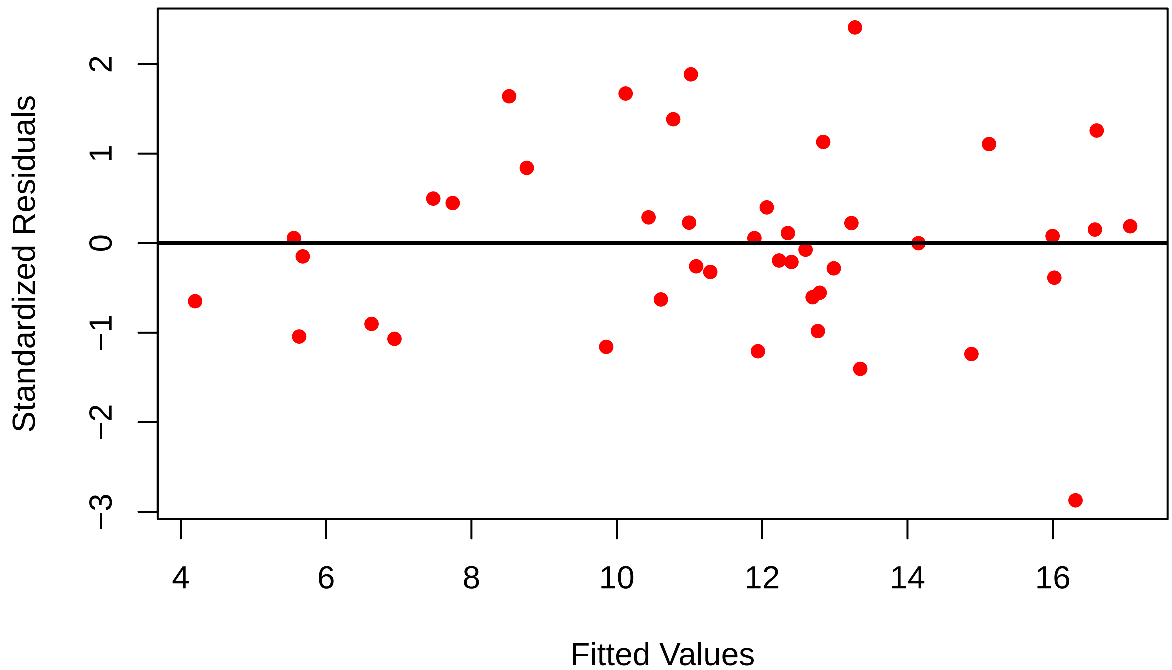
```
####

#### New data (i) ####

new <- read.csv("Gun states simple.csv",header=T)
new$x<-new$Percent.Own.Gun
new$y<-new$Gun.Death.Rate
new<-new[,-(1:7)]
head(new)
```

```
##      x     y
## 1 51.7 17.79
## 2 57.8 19.59
## 3 31.1 14.20
## 4 55.3 16.93
## 5 21.3  7.89
## 6 34.7 11.75
```

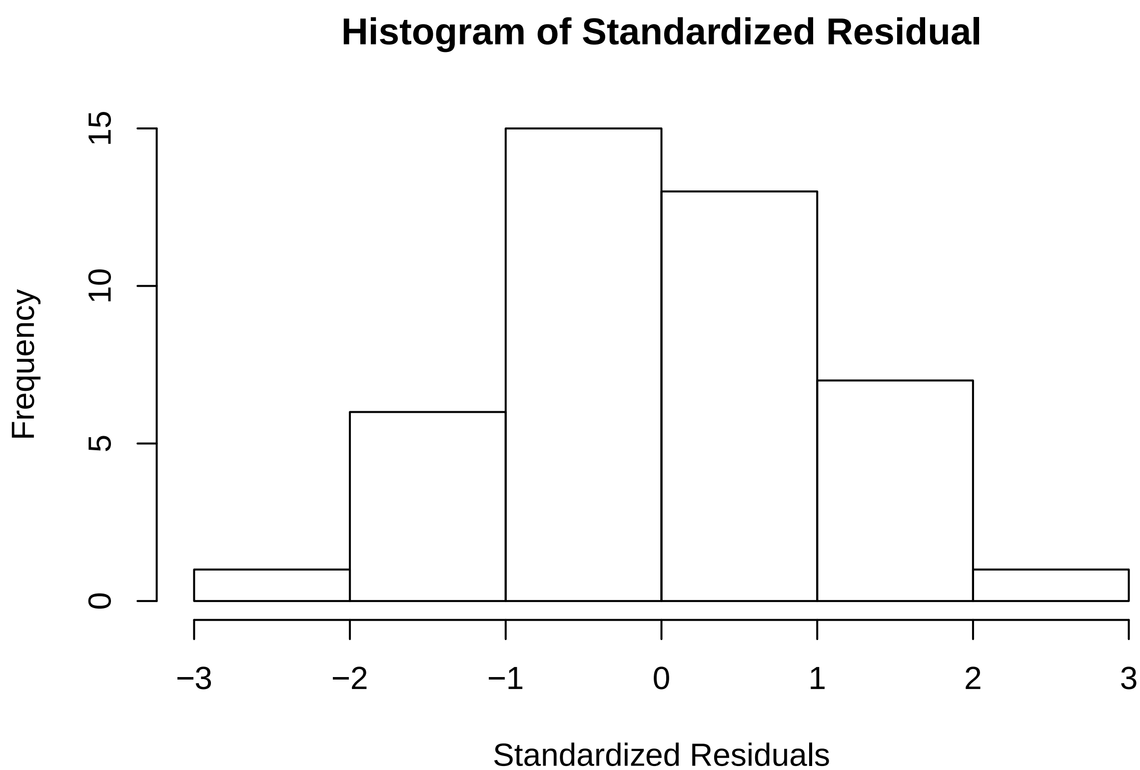
```r
new <- sortedXyData(new$x,new$y,new)       # puts x values in order

# simple linear model
plot.new()
plot(new$y ~ new$x, main = "Companies Data",
    ylab = "Gun death rate",
    xlab = "Percent own gun",
    pch = 20, col = "blue")

newSimpleLinear <- lm(new$y ~ new$x)
abline(newSimpleLinear, lwd = 2)
```




```r
newstanResidSL <- rstandard(newSimpleLinear)

plot(newstanResidSL ~ newSimpleLinear$fitted.values,
     data = new,
     ylab = "Standardized Residuals",
     xlab="Fitted Values",
     pch = 16, col = "red")

abline(h=0, lwd = 2)
```



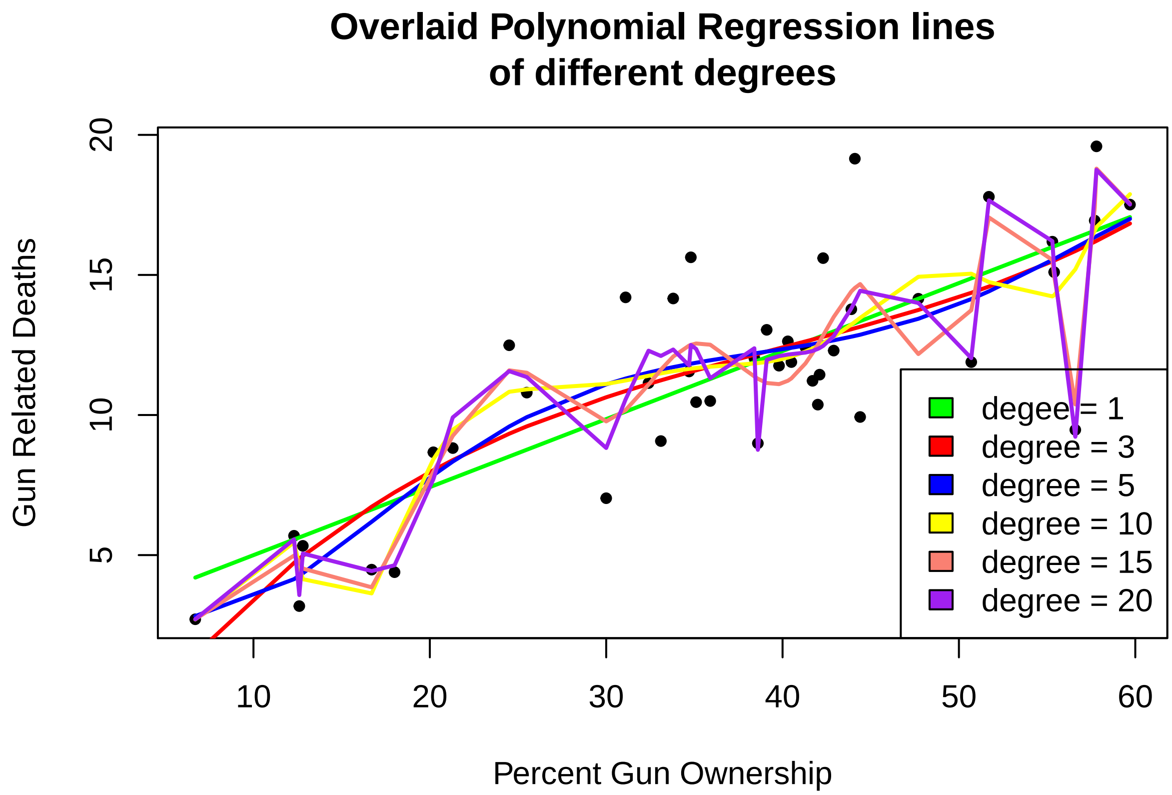

```
hist(newstanResidSL, xlab="Standardized Residuals", main="Histogram of Standardized Residual")
```



The simple linear regression model fits the data pretty well. The linearity and constant variance assumptions should be OK after checking the studentized residuals vs fitted values. It looks like it could fit better though.

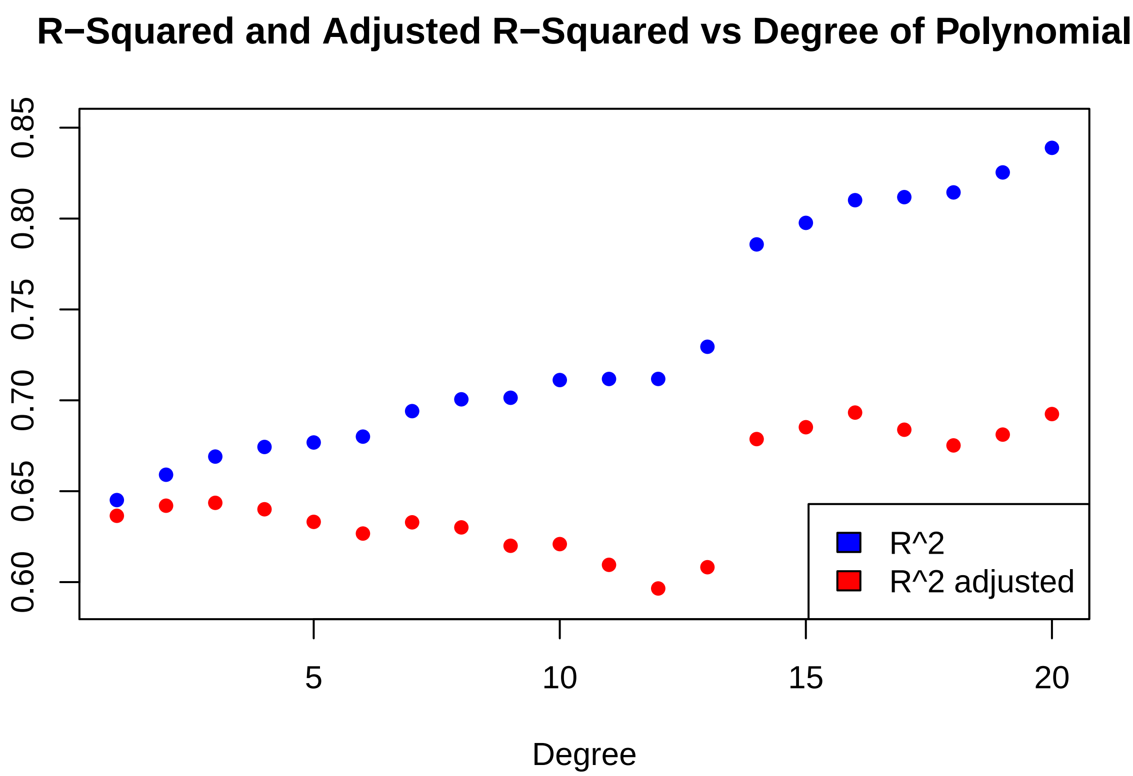
```
newpolynomial1 <- lm(formula = new$y ~ poly(new$x, degree = 1))
newpolynomial2 <- lm(formula = new$y ~ poly(new$x, degree = 2))
newpolynomial3 <- lm(formula = new$y ~ poly(new$x, degree = 3))
newpolynomial4 <- lm(formula = new$y ~ poly(new$x, degree = 4))
```

```r
newpolynomial5 <- lm(formula = new$y ~ poly(new$x, degree = 5))
newpolynomial6 <- lm(formula = new$y ~ poly(new$x, degree = 6))
newpolynomial7 <- lm(formula = new$y ~ poly(new$x, degree = 7))
newpolynomial8 <- lm(formula = new$y ~ poly(new$x, degree = 8))
newpolynomial9 <- lm(formula = new$y ~ poly(new$x, degree = 9))
newpolynomial10 <- lm(formula = new$y ~ poly(new$x, degree = 10))
newpolynomial11 <- lm(formula = new$y ~ poly(new$x, degree = 11))
newpolynomial12 <- lm(formula = new$y ~ poly(new$x, degree = 12))
newpolynomial13 <- lm(formula = new$y ~ poly(new$x, degree = 13))
newpolynomial14 <- lm(formula = new$y ~ poly(new$x, degree = 14))
newpolynomial15 <- lm(formula = new$y ~ poly(new$x, degree = 15))
newpolynomial16 <- lm(formula = new$y ~ poly(new$x, degree = 16))
newpolynomial17 <- lm(formula = new$y ~ poly(new$x, degree = 17))
newpolynomial18 <- lm(formula = new$y ~ poly(new$x, degree = 18))
newpolynomial19 <- lm(formula = new$y ~ poly(new$x, degree = 19))
newpolynomial20 <- lm(formula = new$y ~ poly(new$x, degree = 20))


plot.new()
plot(new$y ~ new$x, main = "Overlaid Polynomial Regression lines\nof different degrees",
     ylab = "Gun Related Deaths",
     xlab = "Percent Gun Ownership",
     pch = 20)

lines(new$x, newpolynomial1$fitted.values, col = "green", lwd = 2)
lines(new$x, newpolynomial3$fitted.values, col = "red", lwd = 2)
lines(new$x, newpolynomial5$fitted.values, col = "blue", lwd = 2)
lines(new$x, newpolynomial10$fitted.values, col = "yellow", lwd = 2)
lines(new$x, newpolynomial15$fitted.values, col = "salmon", lwd = 2)
lines(new$x, newpolynomial20$fitted.values, col = "purple", lwd = 2)
legend("bottomright", legend = c("degee = 1", "degree = 3",
                                 "degree = 5", "degree = 10",
                                 "degree = 15", "degree = 20"),
                                  fill = c("green", "red", "blue", "yellow", "salmon", "purple"))
```




```r
plot.new()
plot(y = c(summary(newpolynomial1)$r.squared,
           summary(newpolynomial2)$r.squared,
           summary(newpolynomial3)$r.squared,
           summary(newpolynomial4)$r.squared,
           summary(newpolynomial5)$r.squared,
           summary(newpolynomial6)$r.squared,
           summary(newpolynomial7)$r.squared,
           summary(newpolynomial8)$r.squared,
           summary(newpolynomial9)$r.squared,
           summary(newpolynomial10)$r.squared,
           summary(newpolynomial11)$r.squared,
           summary(newpolynomial12)$r.squared,
           summary(newpolynomial13)$r.squared,
           summary(newpolynomial14)$r.squared,
           summary(newpolynomial15)$r.squared,
           summary(newpolynomial16)$r.squared,
           summary(newpolynomial17)$r.squared,
           summary(newpolynomial18)$r.squared,
           summary(newpolynomial19)$r.squared,
           summary(newpolynomial20)$r.squared),
           x = 1:20, ylab = "", xlab = "Degree",
           main ="R-Squared and Adjusted R-Squared vs Degree of Polynomial",
           pch = 16, col = "blue", ylim = c(0.59, .85))


points(x = 1:20, y = c(summary(newpolynomial1)$adj.r.squared,
                       summary(newpolynomial2)$adj.r.squared,
                       summary(newpolynomial3)$adj.r.squared,
```

```
                       summary(newpolynomial4)$adj.r.squared,
                       summary(newpolynomial5)$adj.r.squared,
                       summary(newpolynomial6)$adj.r.squared,
                       summary(newpolynomial7)$adj.r.squared,
                       summary(newpolynomial8)$adj.r.squared,
                       summary(newpolynomial9)$adj.r.squared,
                       summary(newpolynomial10)$adj.r.squared,
                       summary(newpolynomial11)$adj.r.squared,
                       summary(newpolynomial12)$adj.r.squared,
                       summary(newpolynomial13)$adj.r.squared,
                       summary(newpolynomial14)$adj.r.squared,
                       summary(newpolynomial15)$adj.r.squared,
                       summary(newpolynomial16)$adj.r.squared,
                       summary(newpolynomial17)$adj.r.squared,
                       summary(newpolynomial18)$adj.r.squared,
                       summary(newpolynomial19)$adj.r.squared,
                       summary(newpolynomial20)$adj.r.squared),
                       pch = 16, col = "red")
                            legend("bottomright",
                                    legend = c("R^2", "R^2 adjusted"),
                                    fill = c("blue", "red"))
```




```
####

#### New data (ii) ####

xx <- new$x
yy <- new$y
mm <- lm(yy ~ xx)
```

```r
#newsegm <- segmented(mm, seg.Z =~xx,
#                       c(20, 35, 49),
#                       xlab = "Percent Gun Ownership",
#                       ylab = "Gun Related deaths")

# plot(xx, yy, pch = 20, main = "Piecewise Regression\nOrder 1")
# lines(xx, newsegm$fitted.values, col = "green", lwd = 2)
```

Figure 6:

```r
xx2 <- new$x^2
mm2 <- lm(yy ~ xx + xx2)
#newsegm2 <- segmented(mm2, seg.Z=~xx+xx2,
#                        psi = list(xx=c(20, 35, 49),
#                                   xx2=c(20, 35, 49)^2))

# plot(xx, yy, xlab = "Percent Gun Ownership",
#      ylab = "Gun Related deaths",
#      main = "Piecewise Regression\nOrder 2",
#      pch = 20)

# lines(xx, newsegm2$fitted.values, col = "red", lwd = 2)
```

```r
a <-30
b <- 36
c <- 47
```

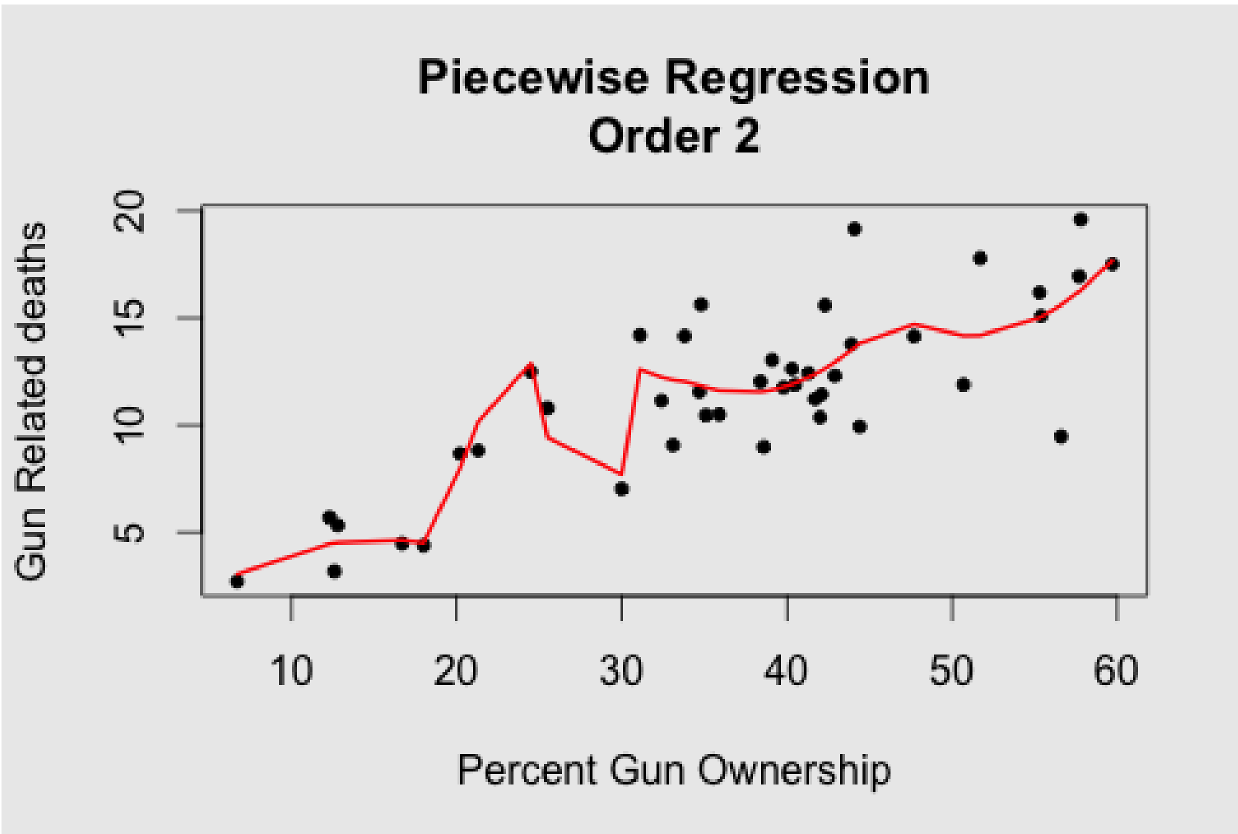

Piecewise Regression
Order 2
Gun Related deaths
5
10
15
20
10
20
30
40
50
60
Percent Gun Ownership

Figure 7:

```r
xx3 <- new$x^3
mm3 <- lm(yy ~ xx + xx2 + xx3)
# newsegm3 <- segmented(mm3, seg.Z=~xx+xx2+xx3,
#                        psi = list(xx=c(a, b, c),
#                                   xx2=c(a, b, c)^2,
#                                   xx3=c(27, 34, )^3))

# plot(xx, yy, xlab = "Percent Gun Ownership",
#      ylab = "Gun Related deaths",
#      main = "Piecewise Regression\nOrder 3",
#      pch = 20)

# lines(xx, newsegm3$fitted.values, col = "blue", lwd = 2)
```

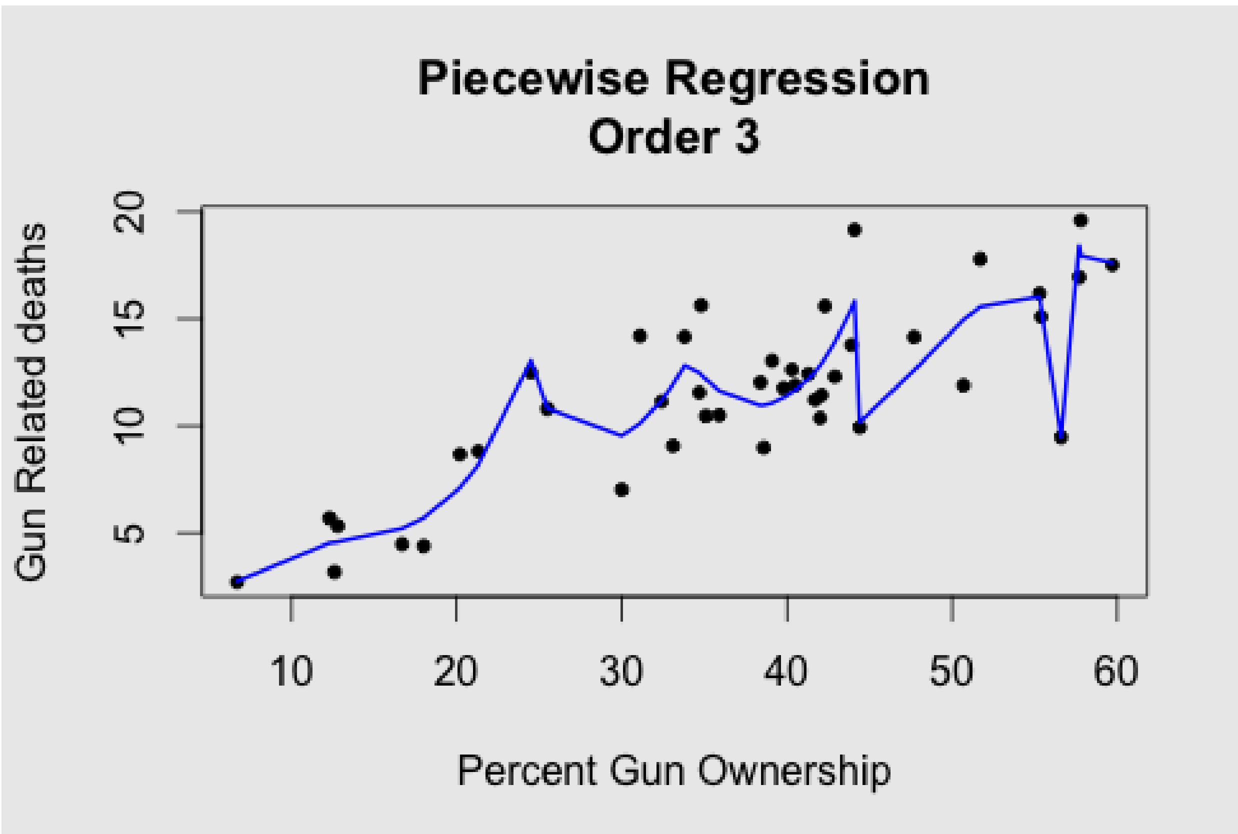


Figure 8:

```r
# plot(xx, yy, pch = 20,
#      main = "Piecewise Resgressions\nof Different Orders",    #overlaid models for comparison
#      ylab = "y")

# lines(xx, newsegm$fitted.values, col = "green", lwd = 2)
# lines(xx, newsegm2$fitted.values, col = "red", lwd = 2)
# lines(xx, newsegm3$fitted.values, col = "blue", lwd = 2)

# legend("bottomright",
#        legend = c("Order 1", "Order 2", "Order 3"),
```

```
#         fill = c("green", "red", "blue"))
```

Figure 9:

```
# plot of r-squared for piecewise regression of different orders
# plot(y = c(summary(newsegm)$r.squared,
#             summary(newsegm2)$r.squared,
#             summary(newsegm3)$r.squared),
#             x = 1:3,
#             ylab = "",
#             xlab = "Degree",
#             main ="R-squared and Adjusted R-squared vs Degree of Segmentation",
#             pch = 16, col = "blue",
#             ylim = c(.60, .83), cex.main = 1.1)

# points(x = 1:3, y = c(summary(newsegm)$adj.r.squared,
#                        summary(newsegm2)$adj.r.squared,
#                        summary(newsegm3)$adj.r.squared),
#                        pch = 16, col = "red")

# legend("bottomright",
#          legend = c("R^2", "R^2 adjusted"),
#          fill = c("blue", "red"))
```

```
####
```

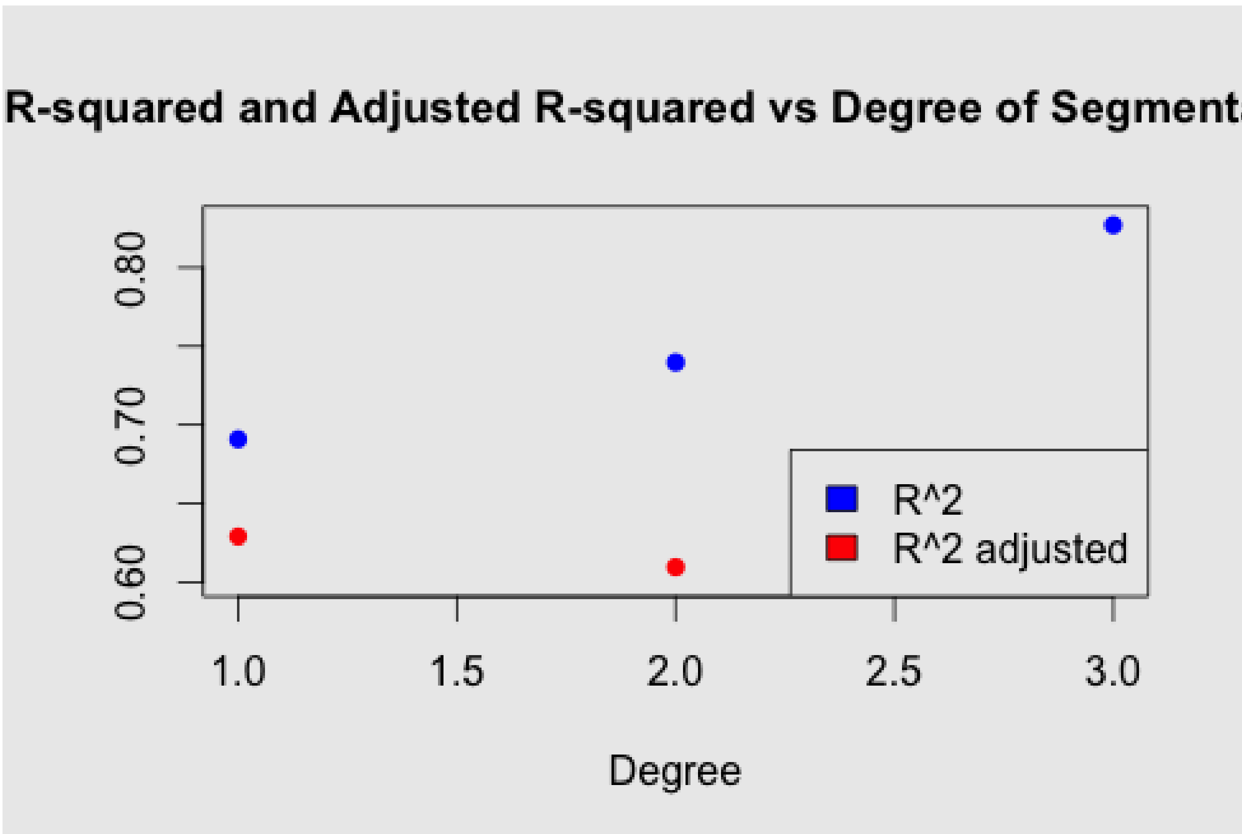
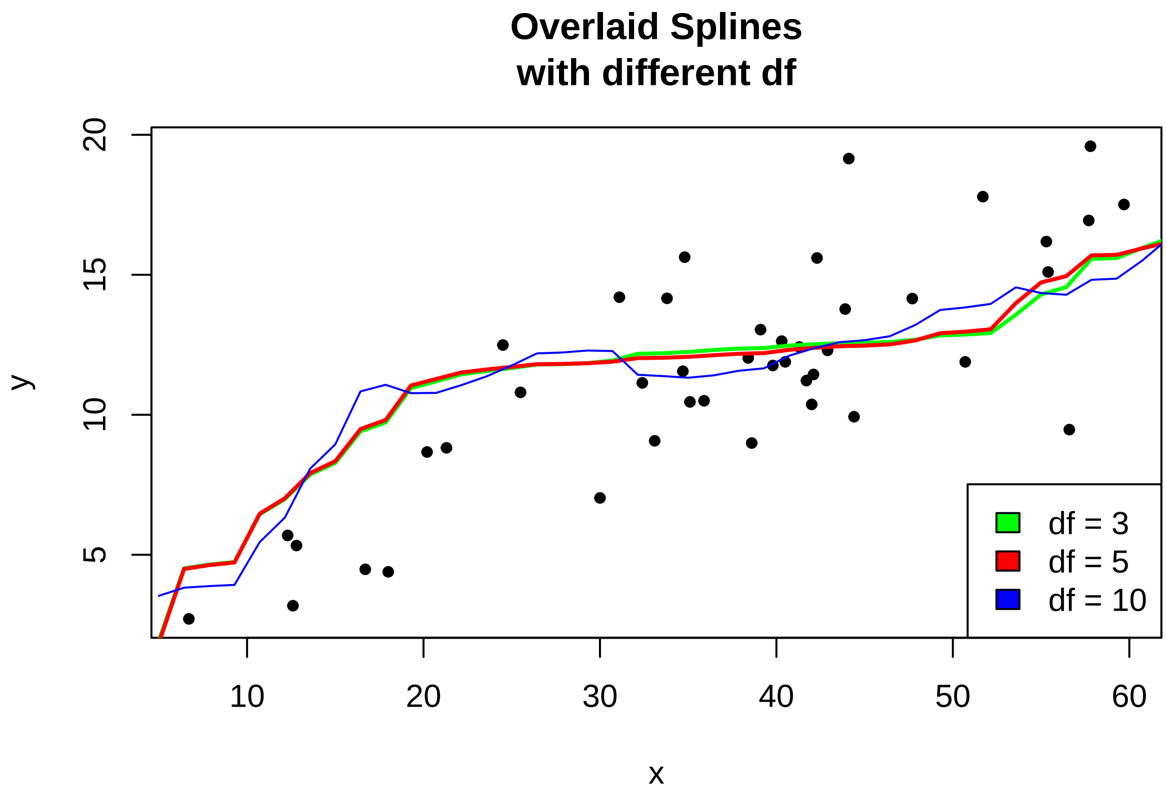
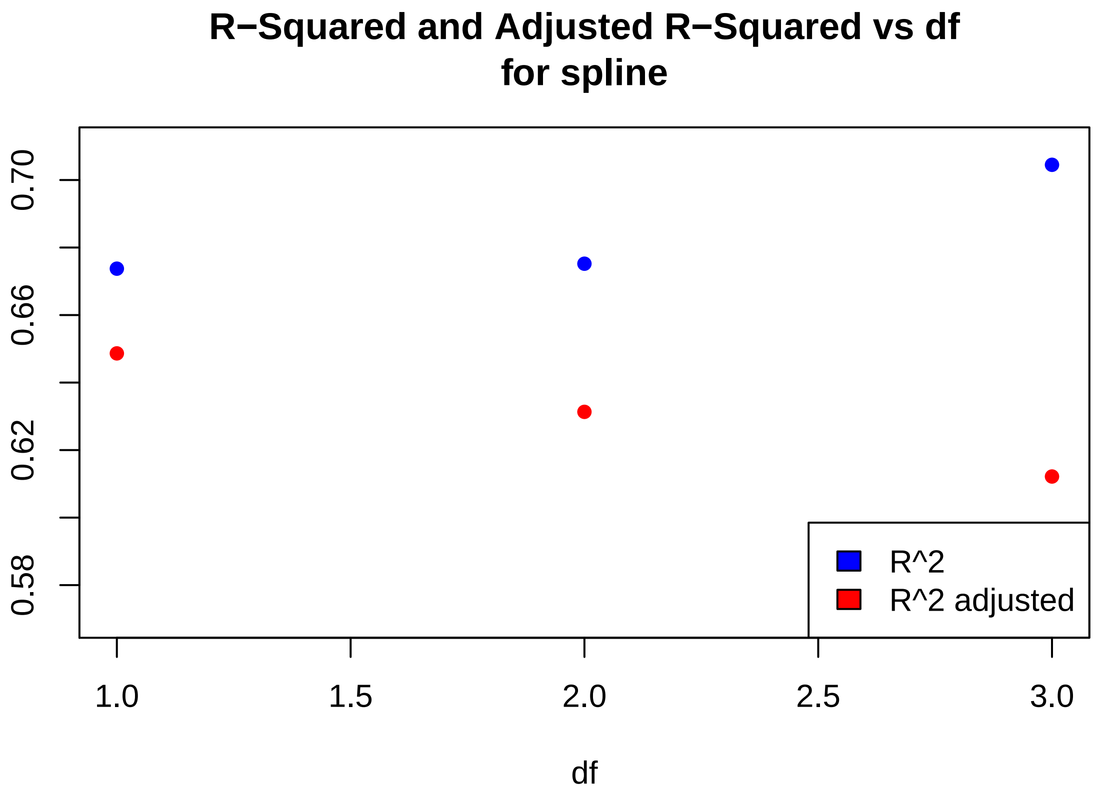

R-squared and Adjusted R-squared vs Degree of Segmenta
0.60
0.70
0.80
1.0
1.5
2.0
2.5
3.0
Degree
R^2
R^2 adjusted

Figure 10:

```r
####New data (iii) Splines####

plot.new()
plot(new$y ~ new$x,
     main = "Overlaid Splines\nwith different df",
     ylab = "y", xlab = "x",
     pch = 20)


nnsp1 <- ns(new$x, df = 3)            #fits a spline with df = 3
ffm1 <- lm(yy ~ nnsp1, data = new)
hht <- seq(5, 65, length.out = 43)

lines(hht,
      predict(ffm1,data.frame(height = hht)),
      col = "green",
      lwd = 2)

nnsp2 <- ns(new$x, df = 5)           #fits a spline with df = 5
ffm2 <- lm(yy ~ nnsp2, data = new)
lines(hht,
      predict(ffm2, data.frame(height = hht)),
      col = "red",
      lwd = 2)

nnsp3 <- ns(new$x, df = 10)          #fits a spline with df = 10
ffm3 <- lm(yy ~ nnsp3, data = new)
lines(hht,
      predict(ffm3, data.frame(height = hht)),
      col = "blue")

legend("bottomright",
       legend = c("df = 3", "df = 5", "df = 10"),
       fill = c("green", "red", "blue"))
```




```
# r-squared
plot.new()
plot(y = c(summary(ffm1)$r.squared,
           summary(ffm2)$r.squared,
           summary(ffm3)$r.squared),
     x = 1:3,
     ylab = "",
     xlab = "df",
     main ="R-Squared and Adjusted R-Squared vs df\nfor spline",
     pch = 16,
     col = "blue",
     ylim = c(.57, .71))


points(x = 1:3,
       y = c(summary(ffm1)$adj.r.squared,
             summary(ffm2)$adj.r.squared,
             summary(ffm3)$adj.r.squared),
       pch = 16,
       col = "red")

legend("bottomright",
       legend = c("R^2", "R^2 adjusted"),
       fill = c("blue", "red"))
```




```
####Best Models 2####

# plot(new$y ~ new$x,
#      main = "Gun Data",
#      ylab = "Gun Related Deaths",
#      xlab = "Percent Gun Ownership",
#      pch = 20)
#
# lines(new$x, newpolynomial1$fitted.values, col = "blue", lwd = 2)
# lines(xx, newsegm$fitted.values, col = "green", lwd = 2)
# lines(hht, predict(ffm3, data.frame(height = hht)), col = "red")
#
# legend("bottomright",
#        legend = c("Polynomial degree 1",
#                   "Piecewise order 1",
#                   "Spline df=10"),
#        fill = c("blue", "green", "red"))

####
```

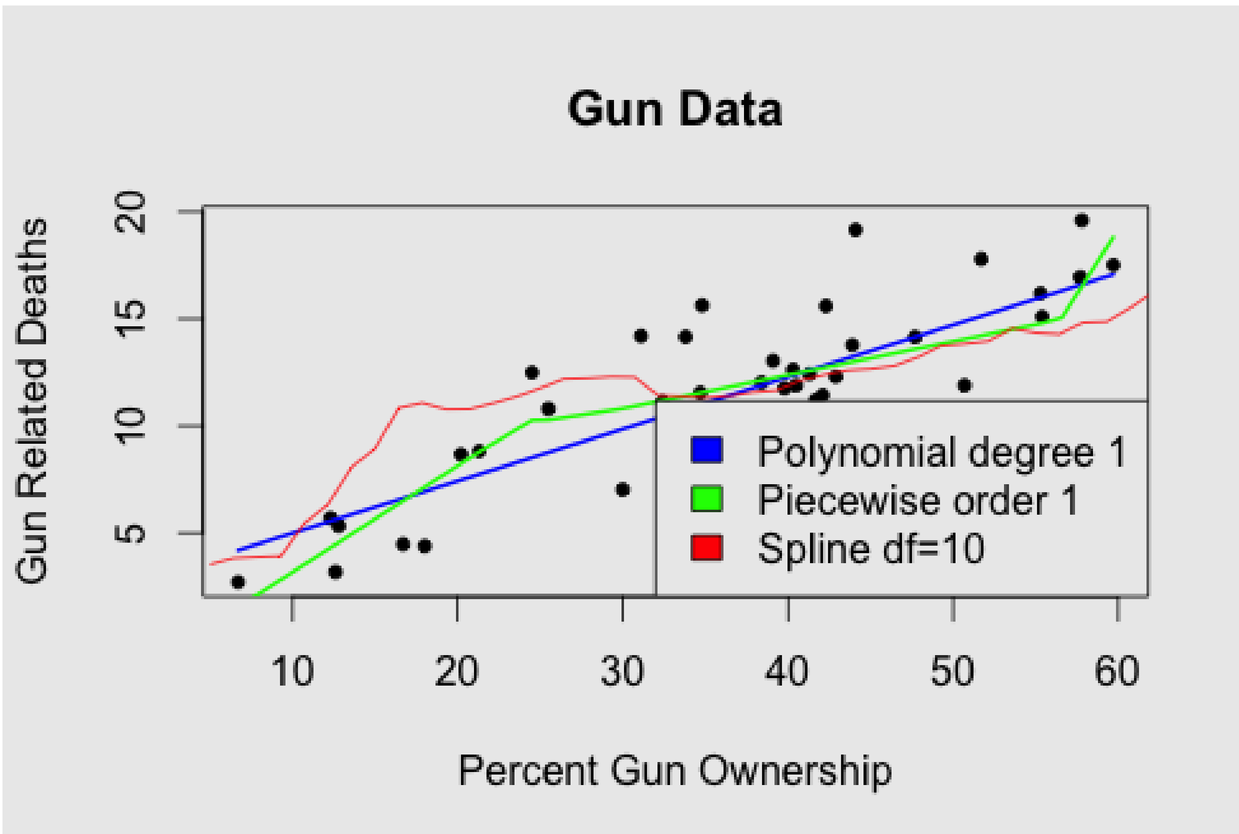

Gun Data
Gun Related Deaths
Percent Gun Ownership
5
10
15
20
10
20
30
40
50
60
Polynomial degree 1
Piecewise order 1
Spline df=10

Figure 11: