Week 2 Homework

Ben Arancibia

June 20, 2015

HA 4.1 Electricity consumption was recorded for a small town on 12 randomly chosen days. The following maximum temperatures (degrees Celsius) and consumption (megawatt-hours) were recorded for each day.

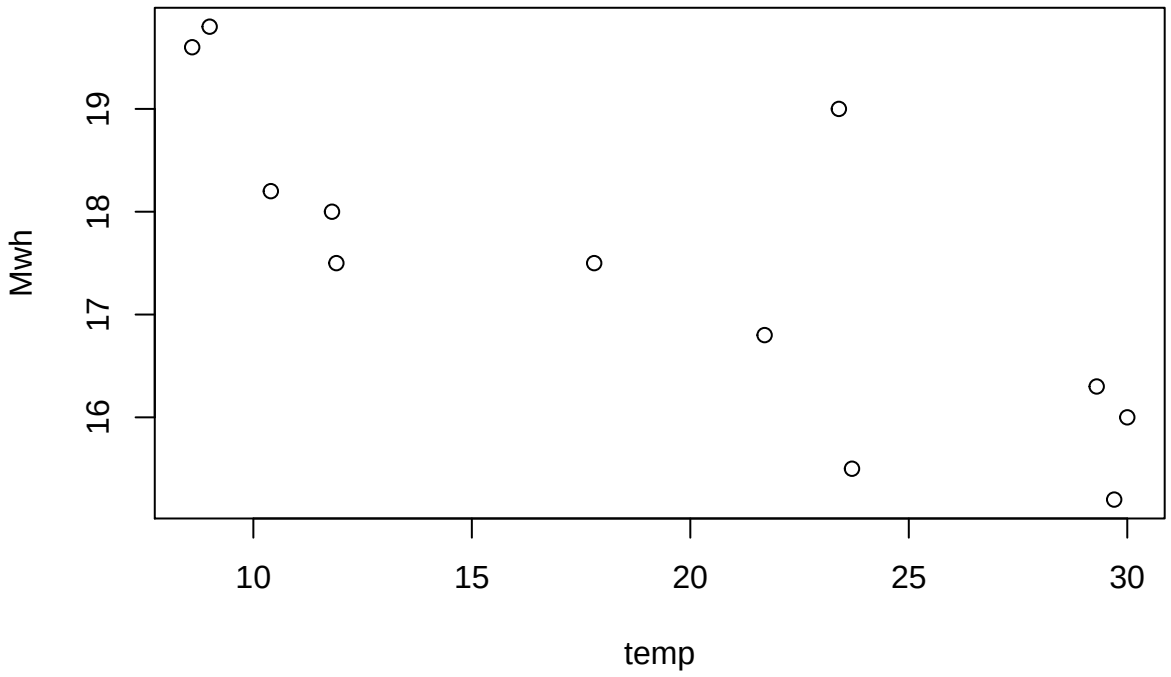
```
library(fpp)
```

```
knitr::kable(econsumption)
```

Mwh	temp
16.3	29.3
16.8	21.7
15.5	23.7
18.2	10.4
15.2	29.7
17.5	11.9
19.8	9.0
19.0	23.4
17.5	17.8
16.0	30.0
19.6	8.6
18.0	11.8

a) Plot the data and find the regression model for Mwh with temperature as an explanatory variable. Why is there a negative relationship?

```
plot(Mwh ~ temp, data=econsumption)
```



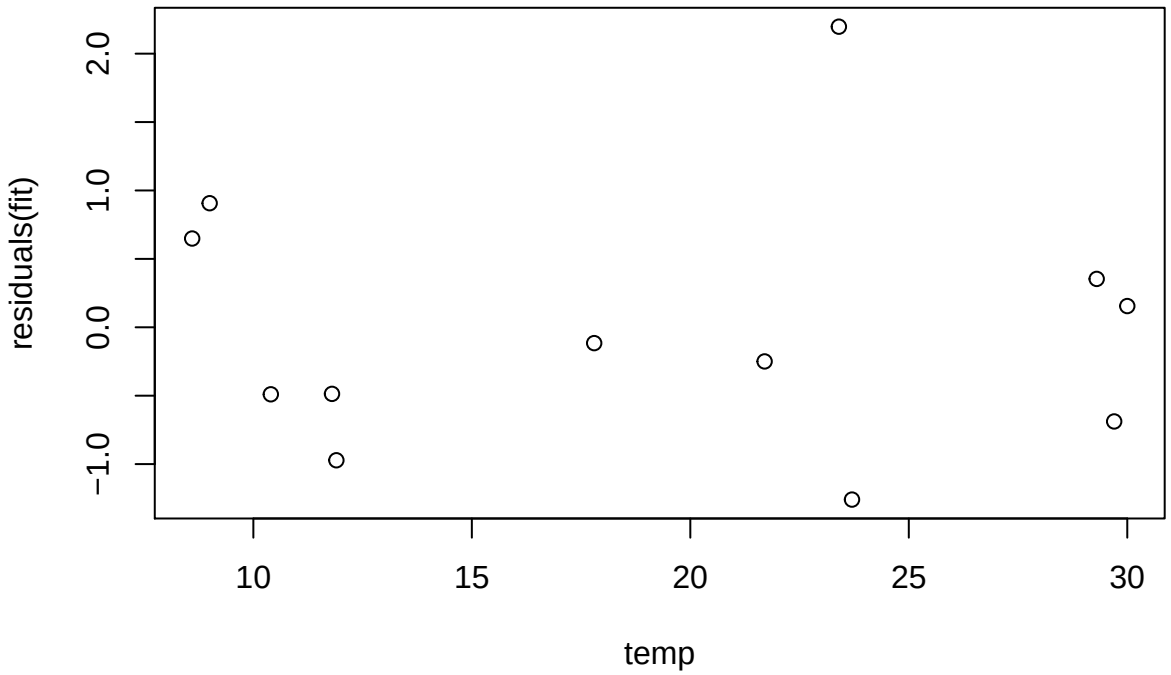

```
fit  <- lm(Mwh ~ temp, data=econsumption)
summary(fit)
```

```
##
## Call:
## lm(formula = Mwh ~ temp, data = econsumption)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2593 -0.5395 -0.1827  0.4274  2.1972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.19952    0.73040   27.66 8.86e-11 ***
## temp        -0.14516    0.03549   -4.09  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9888 on 10 degrees of freedom
## Multiple R-squared:  0.6258, Adjusted R-squared:  0.5884
## F-statistic: 16.73 on 1 and 10 DF,  p-value: 0.00218
```

It seems that there is a negative relationship because a simple linear model is not appropriate. A non-linear model will be necessary for the data.

b) Produce a residual plot. Is the model adequate? Are there any outliers or influential observations?

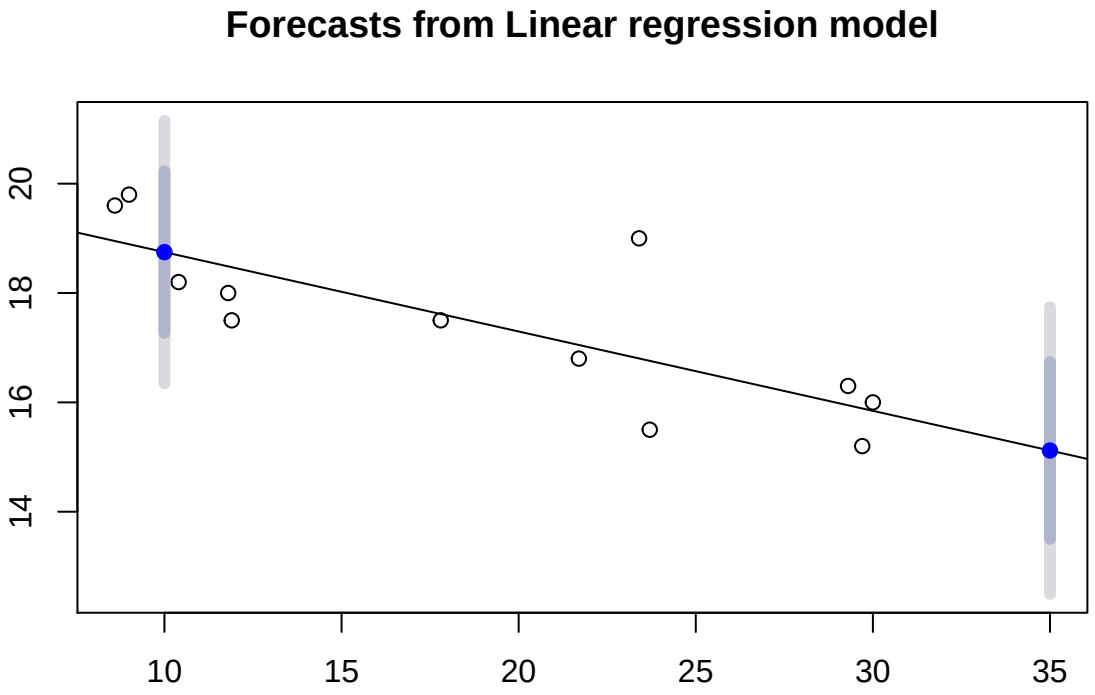
```
plot(residuals(fit)~temp, data=econsumption)
```



The model appears to be adequate, there is no apparent pattern. The only thing that might cause an issues is 1 outlier at temperature 23.4.

c) Use the model to predict the electricity consumption that you would expect for a day with maximum temperature 10 and a day with maximum temperature 35. Do you believe these predictions?

```
fcast <- forecast(fit, newdata=data.frame(temp=c(10,35)))
plot(fcast)
```



Based on the predictive intervals and the nearby data points, the predictions are believable.

d) Give prediction intervals for your forecasts. The following R code will get you started:

```
summary(fit)
```

```
##
## Call:
## lm(formula = Mwh ~ temp, data = econsumption)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2593 -0.5395 -0.1827  0.4274  2.1972
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.19952    0.73040   27.66 8.86e-11 ***
## temp        -0.14516    0.03549   -4.09  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9888 on 10 degrees of freedom
## Multiple R-squared:  0.6258, Adjusted R-squared:  0.5884
## F-statistic: 16.73 on 1 and 10 DF,  p-value: 0.00218
```

```
forecast(fit, newdata=data.frame(temp=c(60)))
```

```
##   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 1       11.49008 9.041979 13.93819 7.514874 15.46529
```

HA 5.2 The data below (data set texasgas) shows the demand for natural gas and the price of natural gas for 20 towns in Texas in 1969.

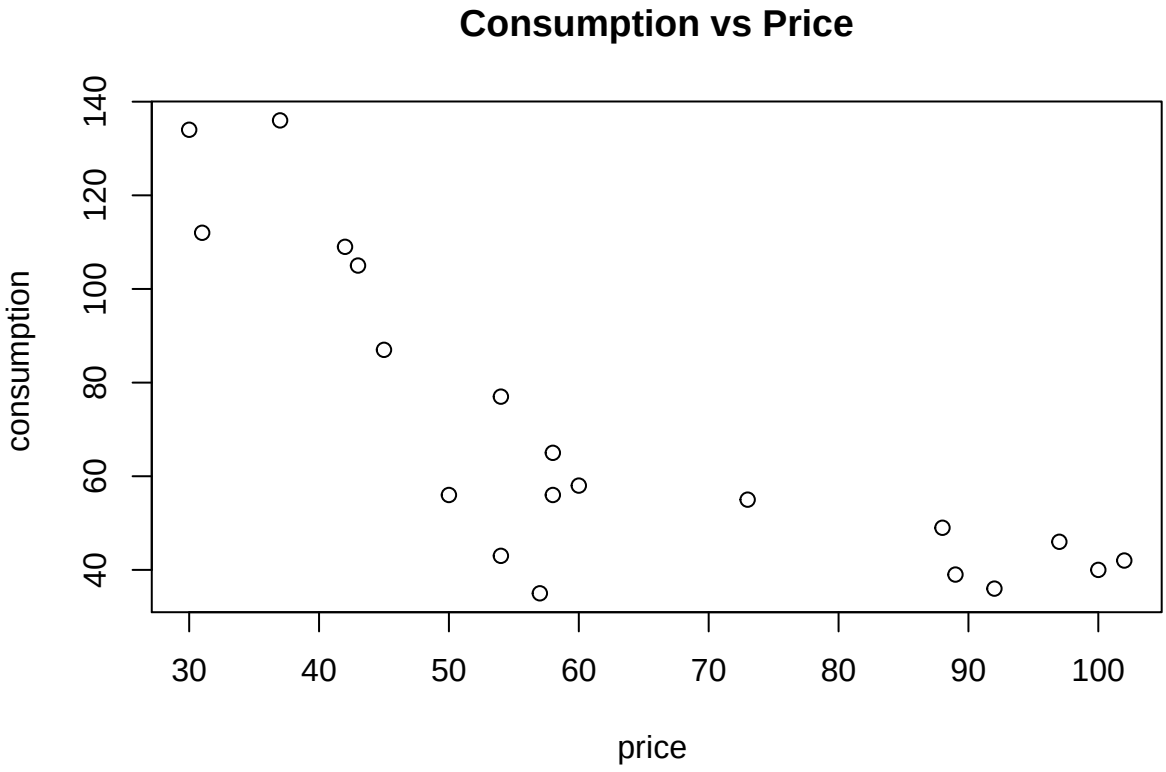
```
knitr::kable(texasgas)
```

price	consumption
30	134
31	112
37	136
42	109
43	105
45	87
50	56
54	43
54	77
57	35
58	65
58	56

price	consumption
60	58
73	55
88	49
89	39
92	36
97	46
100	40
102	42

a) Do a scatterplot of consumption against price.

```
plot(consumption ~ price, data=texasgas,main="Consumption vs Price", xlab="price ", ylab="consumption "
```

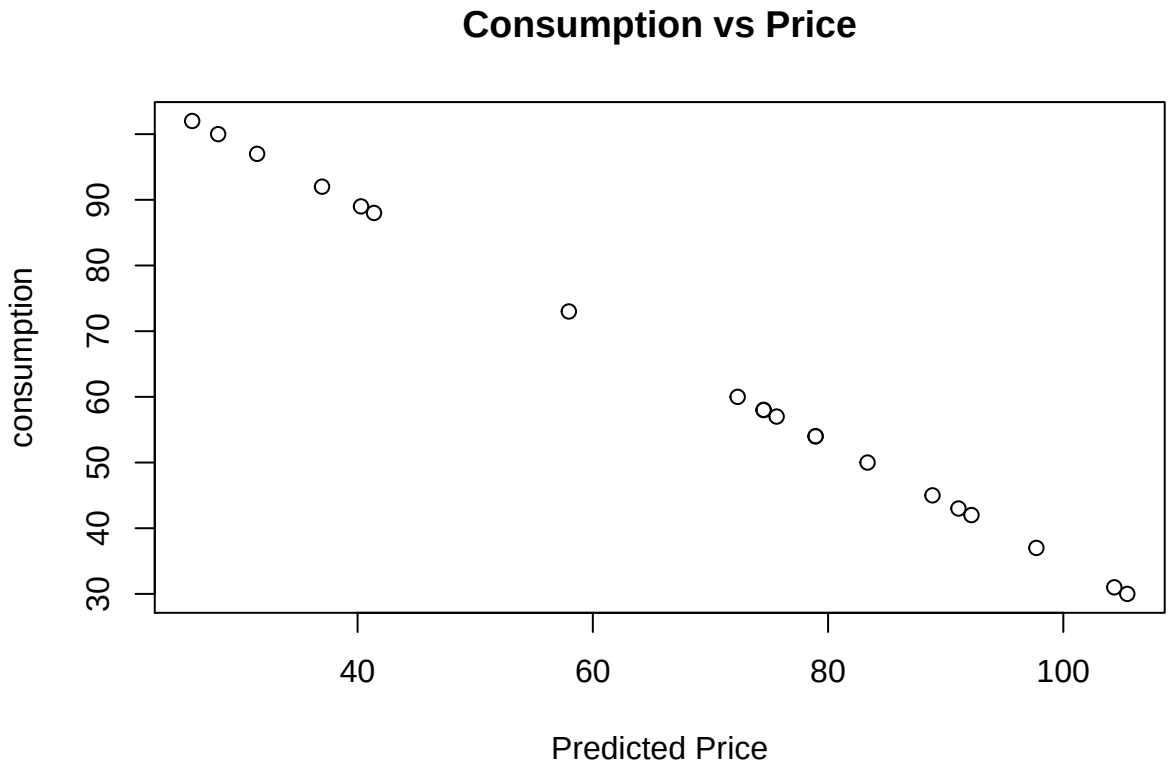


b) The slope of the fitted line should change with P, because this is not a linear relationship. It is a non-linear relationship since it is a negatative relationship.

c) Fit the three models and find the coefficients, and residual variance in each case.

Model 1 - Log

```r
fit1 <- lm(consumption~price, data=texasgas)
plot(fitted(fit1), texasgas$price,  main="Consumption vs Price",ylab="consumption", xlab="Predicted Pri
```

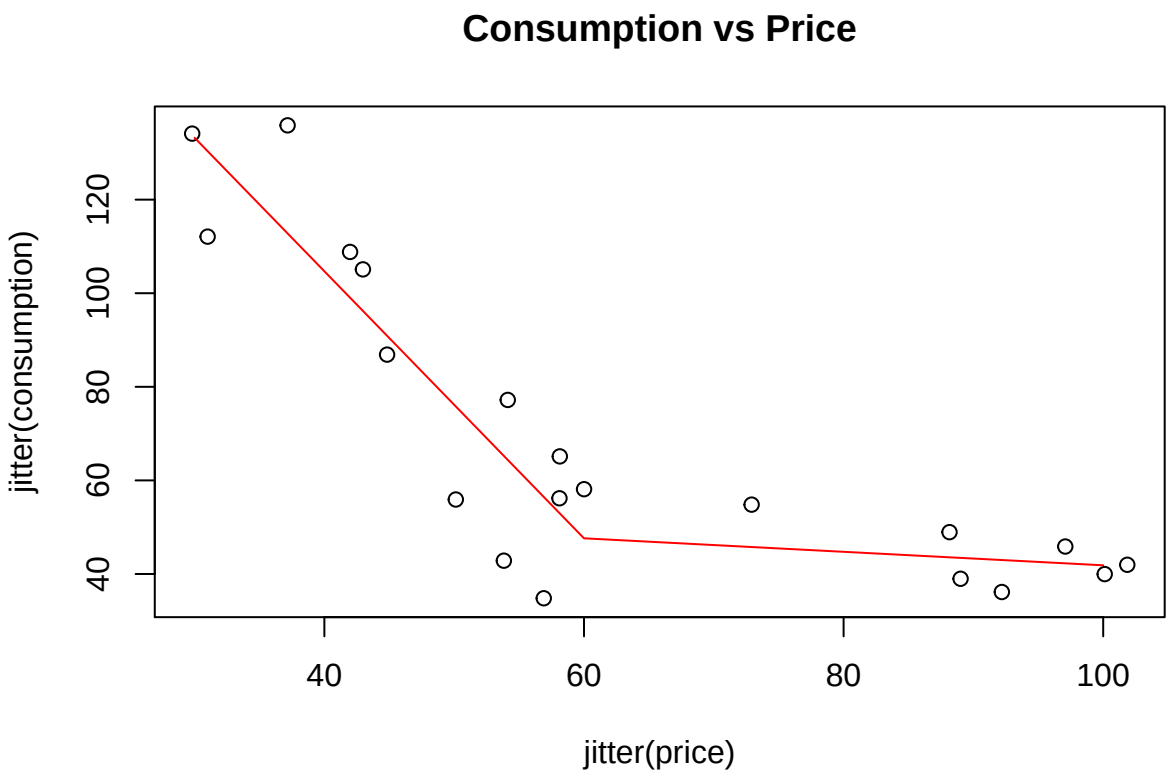



```r
summary(fit1)
```

```
##
## Call:
## lm(formula = consumption ~ price, data = texasgas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.625 -10.719  -1.136  14.073  38.292
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  138.561     13.552  10.225 6.34e-09 ***
## price         -1.104      0.202  -5.467 3.42e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.86 on 18 degrees of freedom
## Multiple R-squared:  0.6241, Adjusted R-squared:  0.6033
## F-statistic: 29.89 on 1 and 18 DF,  p-value: 3.417e-05
```

Model 2 - Piecewise Linear

```r
pricep <- pmax(texasgas$price-60,0)
fit2 <- lm(consumption~price+pricep,data=texasgas)
x <- 30:100; z <- pmax(x-60,0)
fcast2 <- forecast(fit2, newdata=data.frame(price=x,pricep=z))
plot(jitter(consumption)~jitter(price), data=texasgas, main="Consumption vs Price" )
lines(x, fcast2$mean,col="red")
```

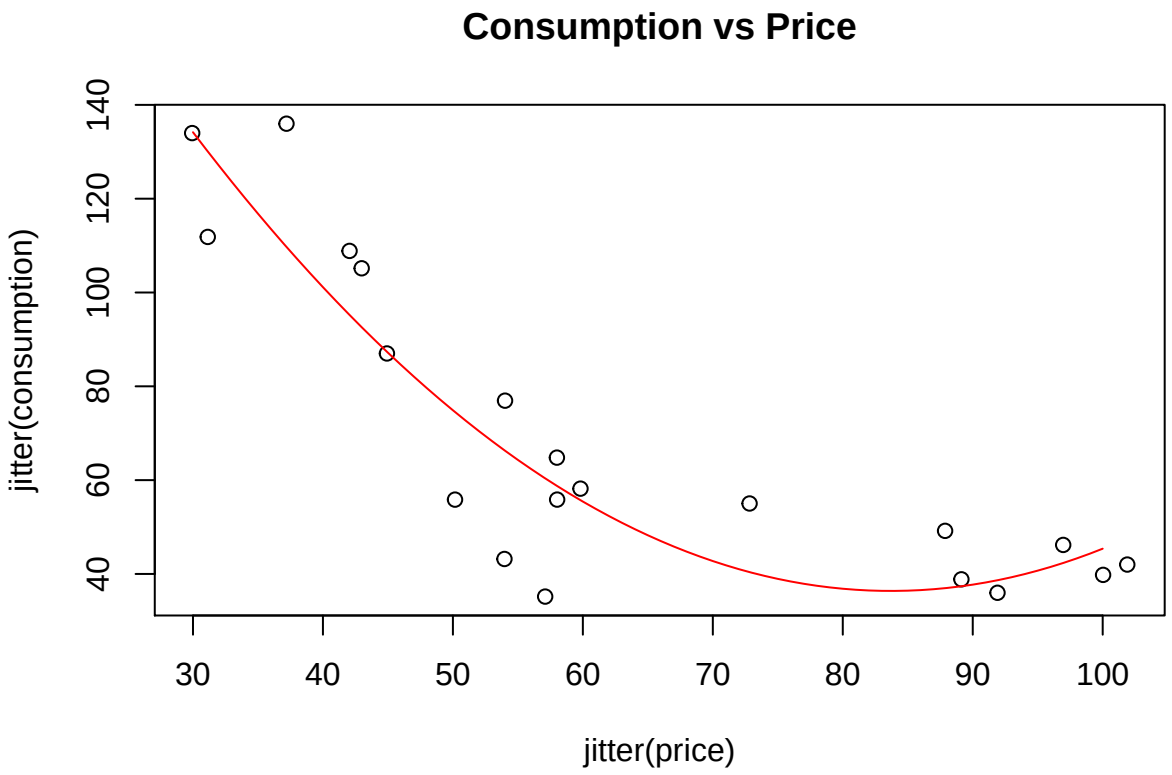



```r
summary(fit2)
```

```
##
## Call:
## lm(formula = consumption ~ price + pricep, data = texasgas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.744  -5.084   1.722   9.442  22.749
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 218.8263    17.4986  12.505 5.34e-10 ***
## price        -2.8534     0.3560  -8.015 3.56e-07 ***
## pricep        2.7092     0.5144   5.266 6.30e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.23 on 17 degrees of freedom
## Multiple R-squared:  0.8572, Adjusted R-squared:  0.8404
## F-statistic: 51.01 on 2 and 17 DF,  p-value: 6.55e-08
```

Model 3 - Polynomial nonlinear regression

```
fit3 <- lm(consumption ~ price + I(price^2), texasgas)
fcast3 <- forecast(fit3, newdata=data.frame(price=x,pricep=z))
plot(jitter(consumption)~jitter(price), data=texasgas, main="Consumption vs Price" )
lines(x, fcast3$mean,col="red")
```




```
summary(fit3)
```

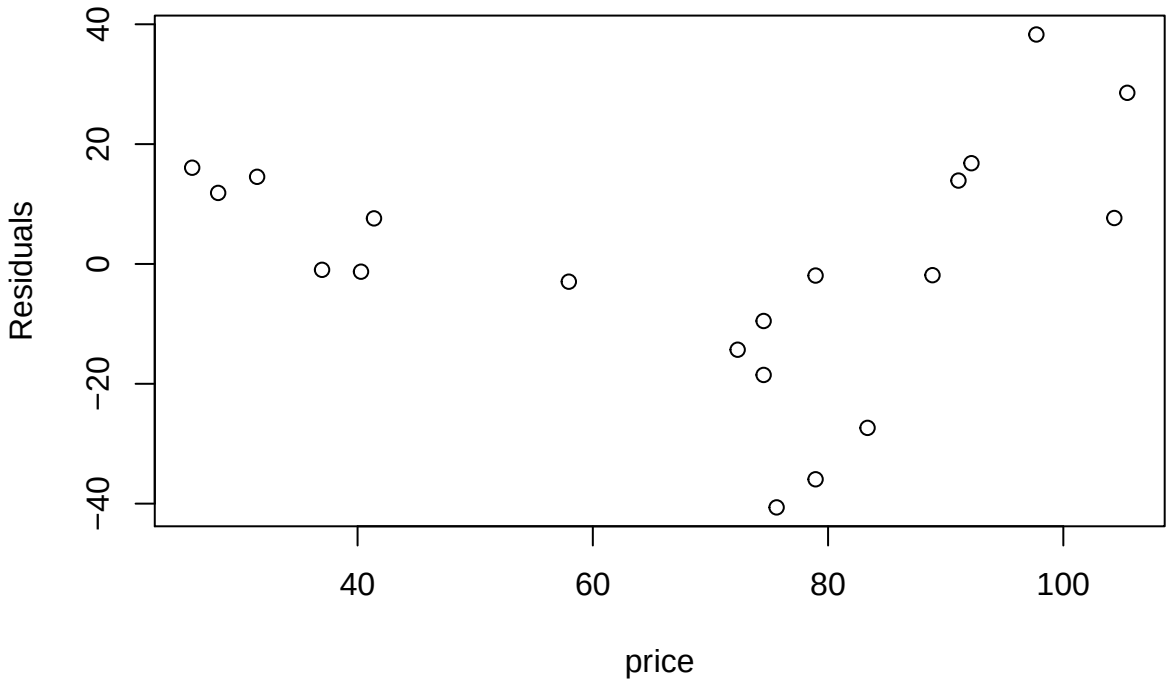
```
##
## Call:
## lm(formula = consumption ~ price + I(price^2), data = texasgas)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.5601  -5.4693   0.7502  11.0252  25.6619
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 273.930628  31.031614   8.827 9.32e-08 ***
## price        -5.675863   1.009086  -5.625 3.03e-05 ***
## I(price^2)    0.033904   0.007412   4.574 0.000269 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.37 on 17 degrees of freedom
## Multiple R-squared:  0.8315, Adjusted R-squared:  0.8117
## F-statistic: 41.95 on 2 and 17 DF,  p-value: 2.666e-07
```

d) For each model, find the value of R2 and AIC, and produce a residual plot. Comment on the adequacy of the three models.

```
CV(fit1)
```

```
##          CV         AIC        AICc         BIC       AdjR2
## 475.7952381 125.4055763 126.9055763 128.3927731   0.6032525
```

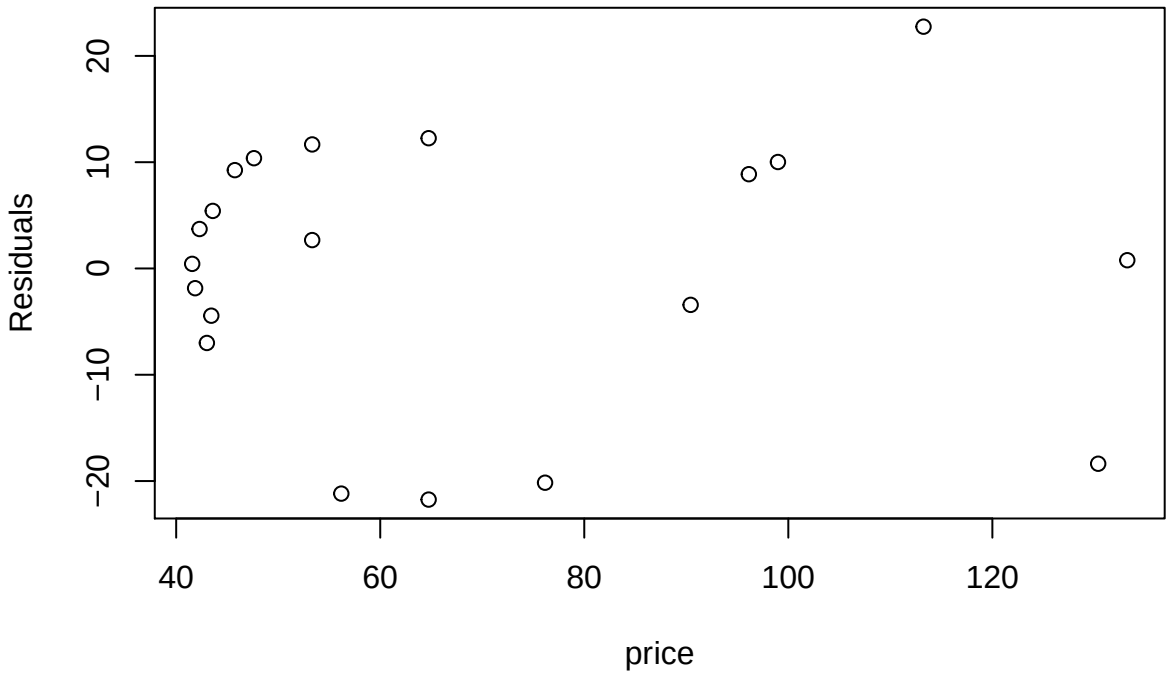
```
plot(fitted(fit1), residuals(fit1), xlab="price", ylab="Residuals")
```




```
CV(fit2)
```

```
##          CV         AIC        AICc         BIC       AdjR2
## 204.3115937 108.0556267 110.7222933 112.0385558   0.8403537
```

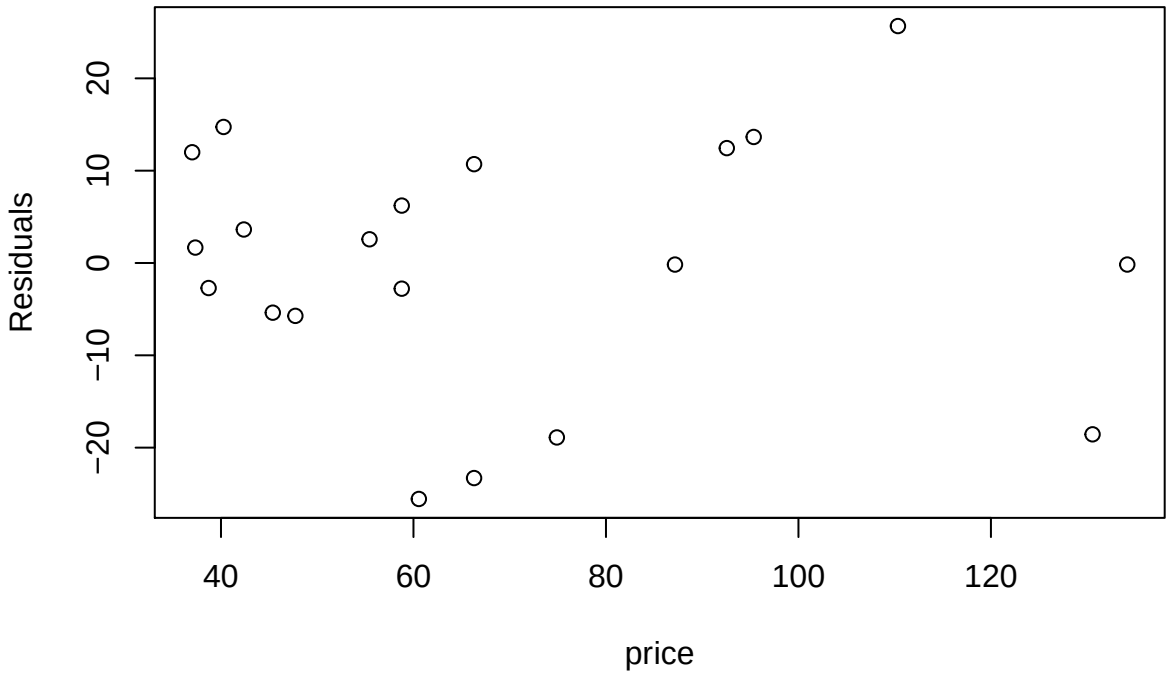
```
plot(fitted(fit2), residuals(fit2), xlab="price", ylab="Residuals")
```




```
CV(fit3)
```

```
##         CV        AIC       AICc        BIC      AdjR2
## 238.565929 111.358304 114.024971 115.341233   0.811689
```

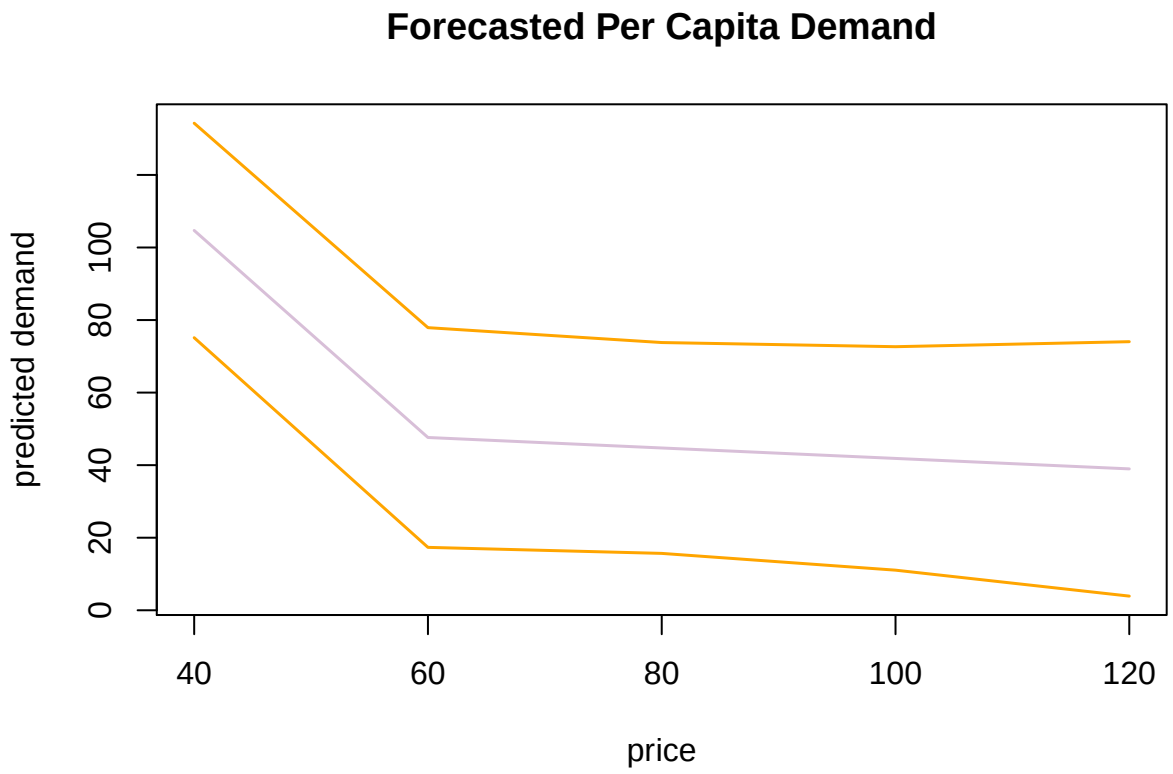
```
plot(fitted(fit3), residuals(fit3), xlab="price", ylab="Residuals")
```



The piecewise linear and polynomial nonlinear regression models have the "best" adjusted R-square values and AIC. It seems that the piecewise linear model does the best explaining the relation of the two variables.

e) For prices 40, 60, 80, 100, and 120 cents per 1,000 cubic feet, compute the forecasted per capita demand using the best model of the three above.

```
x <- seq(40,120, 20); z <- pmax(x-60,0)
predicted <- predict(fit2,newdata=data.frame(price=x,pricep=z),interval="prediction")
matplot(x,predicted,type="l",lty=1,lwd=1.5,col=c("thistle","orange","orange"), xlab="price", ylab="predi
```



f) Compute 95% prediction intervals. Make a graph of these prediction intervals and discuss their interpretation. ####Model 1

```
confint(fit1,level=0.95)
```

```
##                  2.5 %      97.5 %
## (Intercept) 110.090281 167.0317658
## price        -1.528447  -0.6798396
```

```
confint(fit2,level=0.95)
```

Model 2

```
##                  2.5 %     97.5 %
## (Intercept) 181.907467 255.745183
## price        -3.604495  -2.102259
## pricep        1.623794   3.794565
```

```
confint(fit3,level=0.95)
```

Model 3

```
##                     2.5 %      97.5 %
## (Intercept) 208.45964560 339.4016112
## price        -7.80484861  -3.5468767
## I(price^2)    0.01826611   0.0495416
```

g) What is the correlation between P and P2? Does this suggest any general problem to be considered in dealing with polynomial regressions—especially of higher orders?

```
cor(texasgas$price,I(texasgas$price^2))
```

```
## [1] 0.9904481
```

Since it is almost 1, this suggests an issue of multicollinearity. This means it can be difficult to estimate the regression model, uncertainty associated with individual regression coeffiecients will be large, and forecasts will be unreliable if the values of the future predictors are outside the range of the historical values of the predictors.

KJ 6.2

a) Load the necessary data and take a look at it

```
library(AppliedPredictiveModeling)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.1.3

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.1.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
library(pls)
```

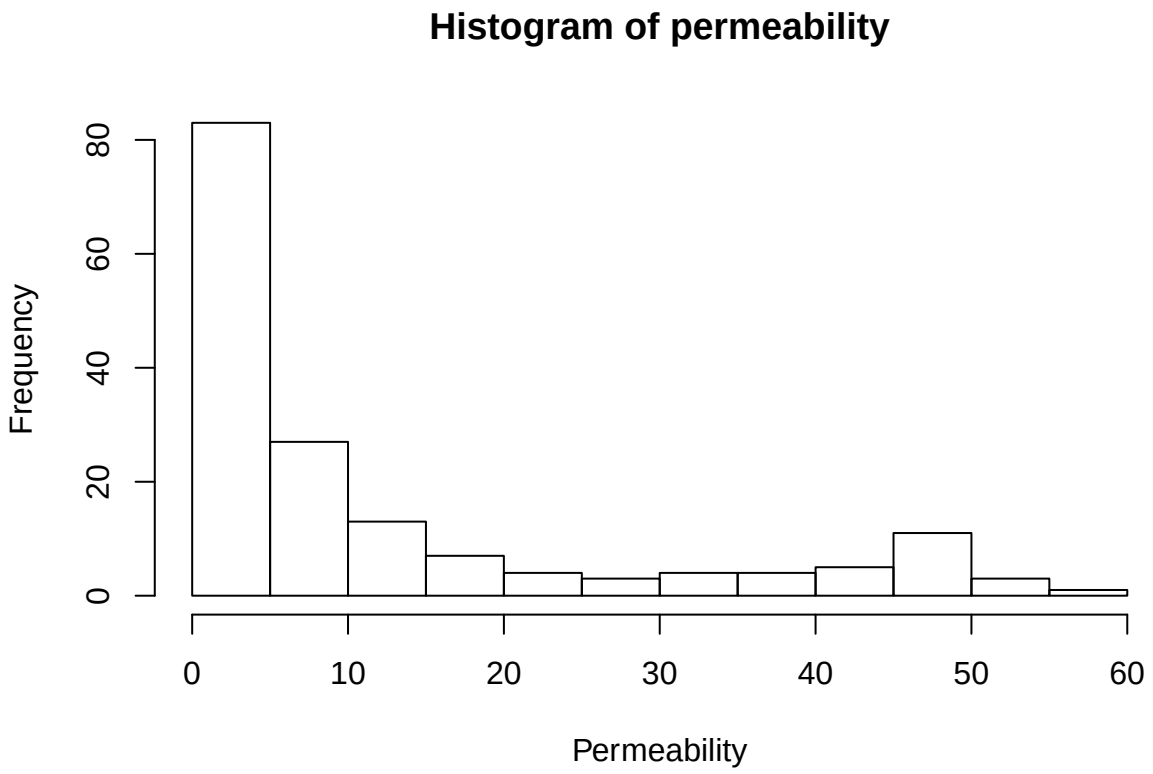
```
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:caret':
##
##     R2
##
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(elasticnet)
```

```
## Loading required package: lars
## Loaded lars 1.2
```

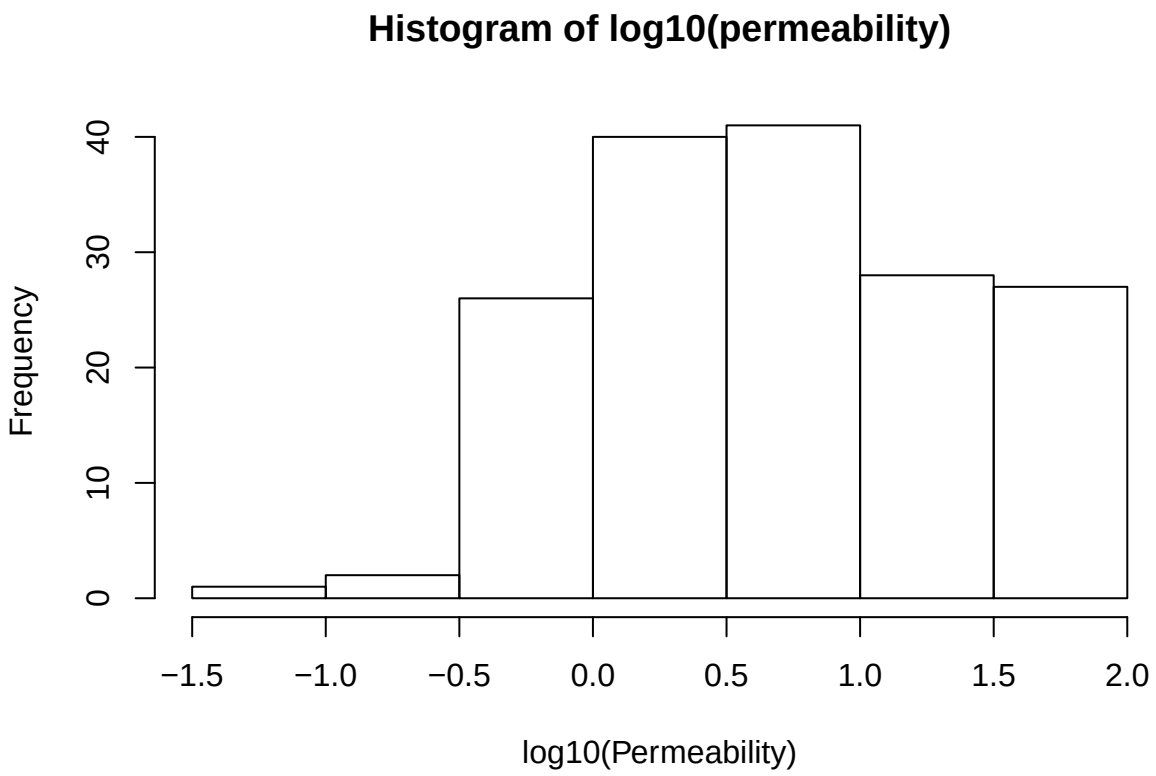
```
data(permeability)

hist(permeability, xlab="Permeability")
```



One thing to notice is the data is skewed, so log-transform the data.

```
hist(log10(permeability), xlab="log10(Permeability)")
```



b) The fingerprint predictors indicate the presence or absence of substructures of a molecule and are often sparse meaning that relatively few of the molecules contain each substructure. Filter out the predictors that have low frequencies using the nearZeroVar function from the caret package. How many predictors are left for modeling?

First pre-process the fingerprint predictors remove near-zero variance fingerprint predictors.

```
nzvfingerprints <- nearZeroVar(fingerprints)
nonzvfingerprints <- fingerprints[,-nzvfingerprints]
length(nzvfingerprints)
```

```
## [1] 719
```

```
ncol(nonzvfingerprints)
```

```
## [1] 388
```

There are 719 near-zero variance fingerprints, which means there is 388 left for modeling. This is a significant reduction from the original and indicates that many of the fingerprints are describing unique features of very small subsets of molecules.

c) Split the data into a training and a test set, pre-process the data, and tune a PLS model. How many latent variables are optimal and what is the corresponding resampled estimate of R2?

```
set.seed(614)
trainingRows <- createDataPartition(permeability,
                                    p = 0.75,
                                    list = FALSE)

trainFingerprints <- nonzvfingerprints[trainingRows,]
trainPermeability <- permeability[trainingRows,]

testFingerprints <- nonzvfingerprints[-trainingRows,]
testPermeability <- permeability[-trainingRows,]
```

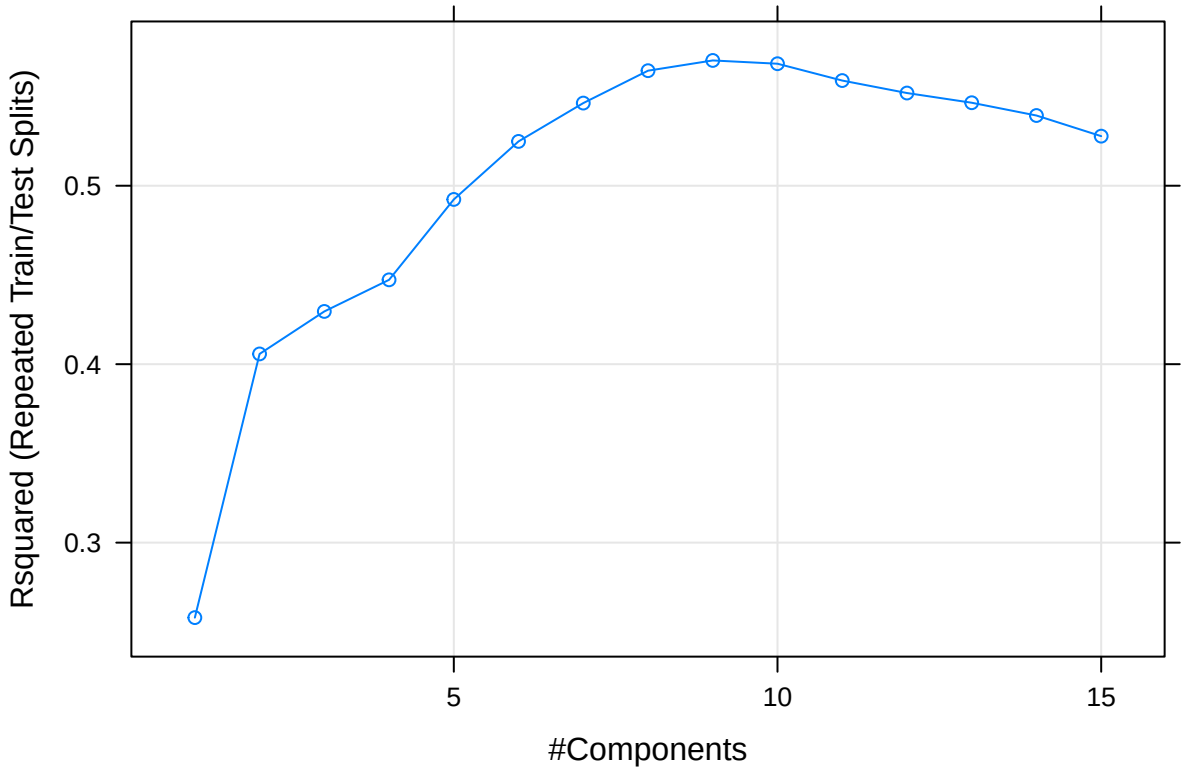
```
set.seed(614)
ctrl <- trainControl(method = "LGOCV")

plsTune <- train(x = trainFingerprints, y = log10(trainPermeability),
                 method = "pls",
                 tuneGrid = expand.grid(ncomp = 1:15),
                 trControl = ctrl)

plsTune
```

```
## Partial Least Squares
##
## 125 samples
## 388 predictors
##
## No pre-processing
## Resampling: Repeated Train/Test Splits Estimated (25 reps, 0.75%)
##
## Summary of sample sizes: 96, 96, 96, 96, 96, 96, ...
##
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   RMSE SD     Rsquared SD
##    1     0.5570643  0.2579863  0.06890454  0.1712800
##    2     0.4986610  0.4057383  0.05352057  0.1276443
##    3     0.4878360  0.4295600  0.05356515  0.1163711
##    4     0.4816227  0.4472692  0.05216970  0.1169232
##    5     0.4636363  0.4923307  0.05027547  0.1131329
##    6     0.4477109  0.5248432  0.05932397  0.1235872
##    7     0.4376023  0.5463110  0.06237137  0.1315045
##    8     0.4301039  0.5644676  0.06340413  0.1264695
##    9     0.4294608  0.5702027  0.05938280  0.1122135
##   10     0.4348420  0.5683697  0.06075378  0.1088045
##   11     0.4422510  0.5589378  0.06537601  0.1165392
##   12     0.4485103  0.5519423  0.07011156  0.1212039
##   13     0.4526875  0.5465051  0.07041862  0.1187694
##   14     0.4578244  0.5393518  0.07448052  0.1254865
##   15     0.4648311  0.5277832  0.07677469  0.1289097
##
## RMSE was used to select the optimal model using  the smallest value.
## The final value used for the model was ncomp = 9.
```

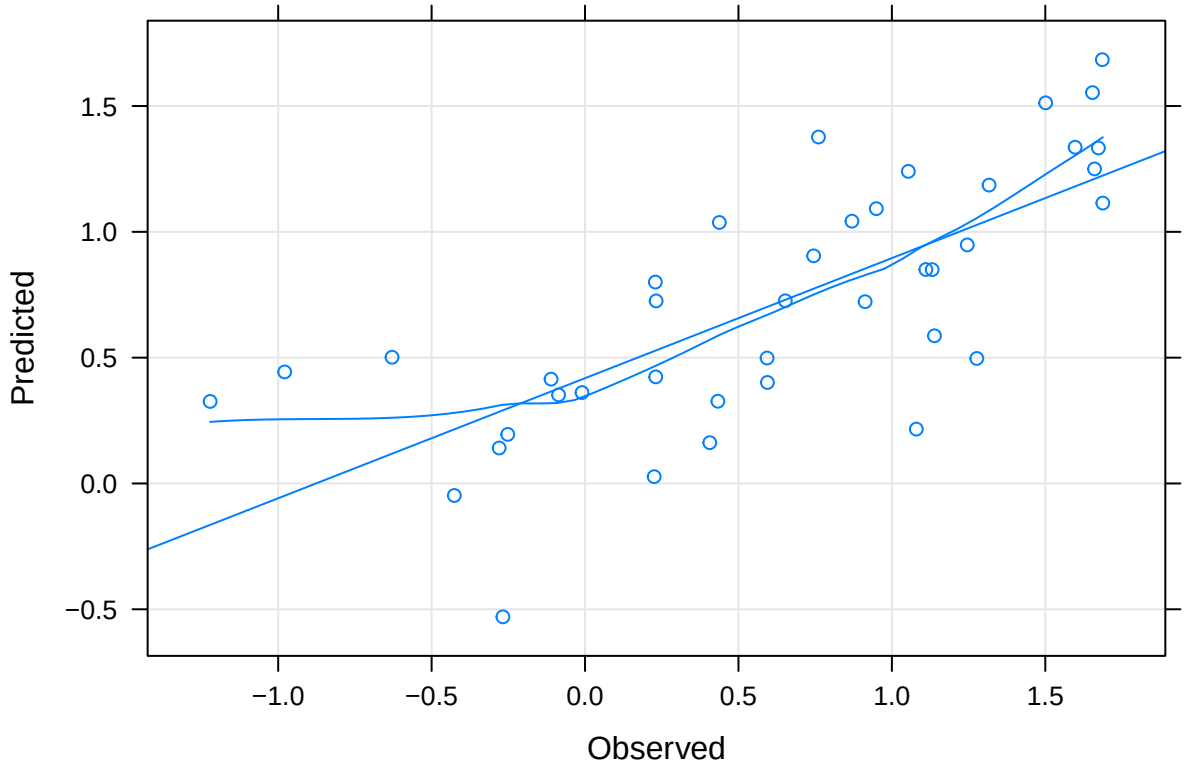
```
plot(plsTune,metric="Rsquared")
```



The above plot indicates that the optimal number of latent variables that maximizes R^2 is 9 so the R^2 value is .57.

d) Predict the response for the test set. What is the test set estimate of R2?

```
plsTest <- data.frame(Observed=log10(testPermeability),Predicted=predict(plsTune,testFingerprints))
xyplot(Predicted ~ Observed,
       plsTest, panel = function(...) {
         theDots <- list(...)
         panel.xyplot(..., type = c("p", "g","r","smooth"))
         corr <- round(cor(theDots$x, theDots$y), 2)
         panel.text(44, min(theDots$y), paste("corr:", corr))
         },
       ylab = "Predicted",
       xlab = "Observed")
```



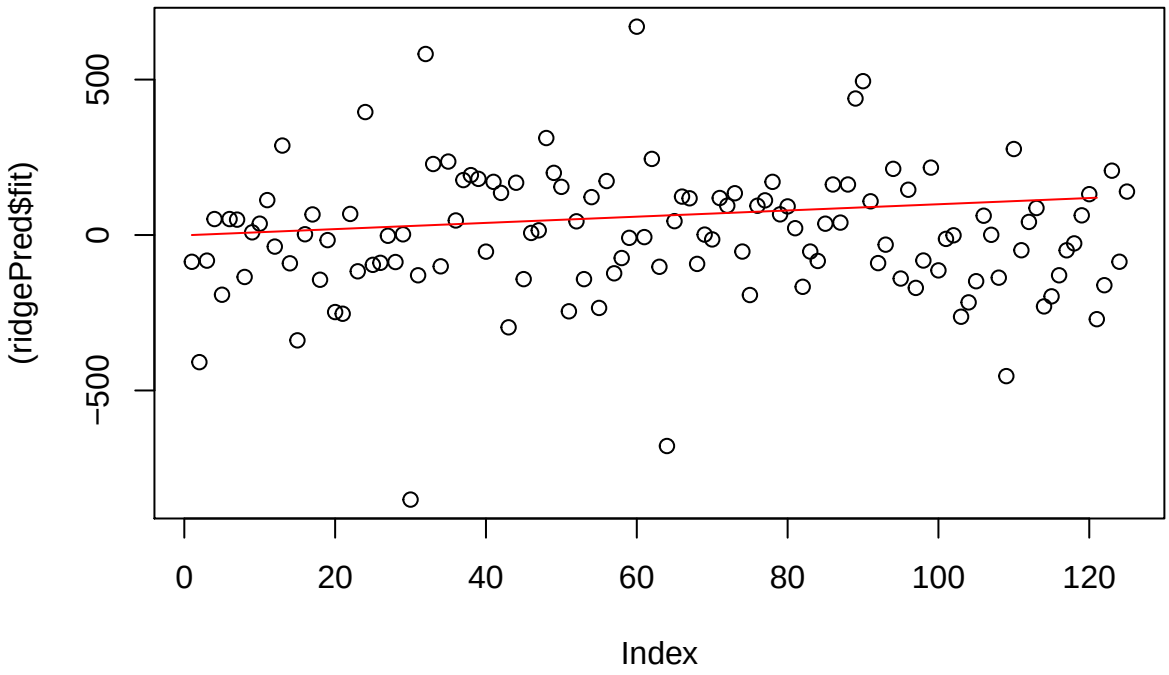
The R^2 value is 0.541

e) Try building other models discussed in this chapter. Do any have better predictive performance?

```
ridgeModel <- enet(x = trainFingerprints, y=log10(trainPermeability),lambda = 0.001)
ridgePred <- predict(ridgeModel, newx =trainFingerprints,s = 1, mode = "fraction",type = "fit")
head(ridgePred$fit)
```

```
##          1          2          3          4          5          6
##  -86.15364 -408.96311  -82.34307   51.40248 -192.09784   51.40248
```

```
plot((ridgePred$fit))
x <- 0:120;
lines(x, ridgePred$mean,col="red")
```




```
enetModel <- enet(x = as.matrix(trainFingerprints), y = log10(trainPermeability), lambda = 0.01, normali
enetPred <- predict(enetModel, newx = as.matrix(trainFingerprints), s = .1, mode = "fraction", type = "
names(enetPred)
```

```
## [1] "s"        "fraction" "mode"     "fit"
```

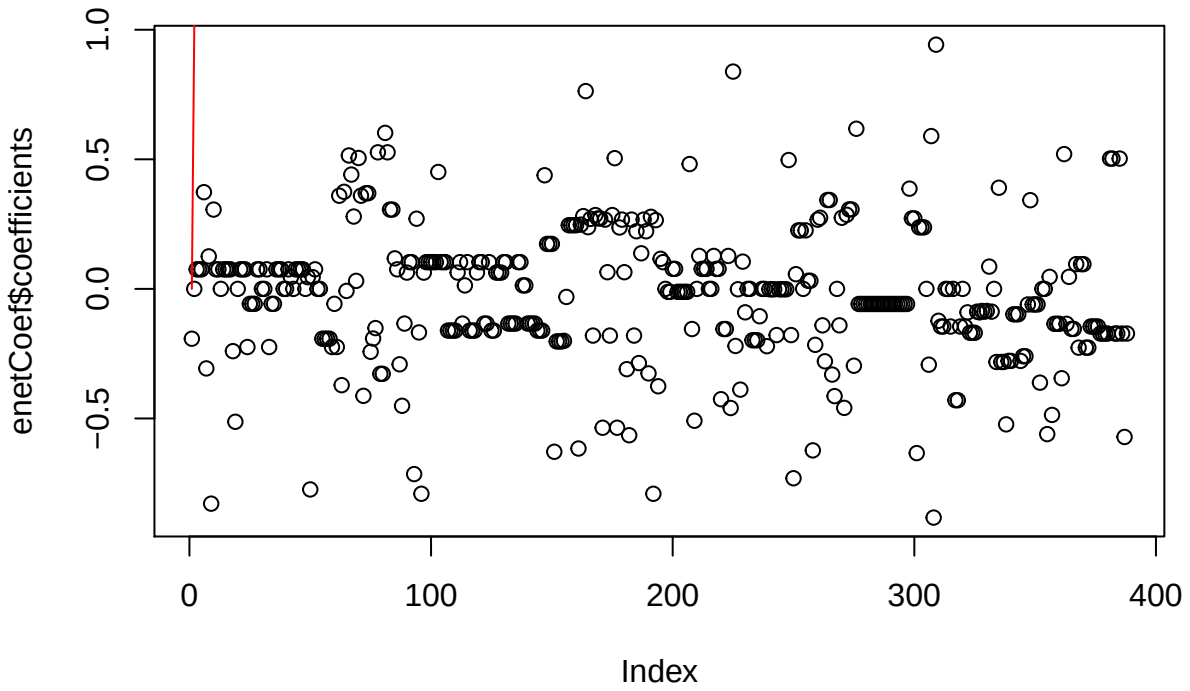
```
head(enetPred$fit)
```

```
##           1           2           3           4           5           6
##  1.08294551  0.22656381  1.04353084 -0.07331627  0.10882714 -0.07331627
```

```
enetCoef<- predict(enetModel, newx = as.matrix(trainFingerprints), s = .1, mode = "fraction", type = "c
tail(enetCoef$coefficients)
```

```
##       X800       X801       X805       X806       X812       X813
## -0.1723545 -0.1723545  0.5028157 -0.1723545 -0.5714399 -0.1723545
```

```
plot(enetCoef$coefficients)
x <- 0:400;
lines(x, enetCoef$mean,col="red")
```



f)

Would you recommend any of your models to replace the permeability laboratory experiment?

The ridge seems to be the best model because there appears to be outliers in the enet model that are not captured in the plot causing the mean to spike immediately.

Notes: I had a lot of issues trying to build the models in part *e*. My local machine kept hanging on the calculation and become unresponsive.

the below code is the code I tried to run to tune the ridge model but kept hanging my machine.

```
ridgeTune <- train(x = trainFingerprints, y = log10(trainPermeability),
+                    method = "ridge",
+                    tuneGrid = ridgeGrid,
+                    trControl = ctrl)
```