

IS624 - Assignment 1

James Quacinella

06/19/2015

Question 4.1

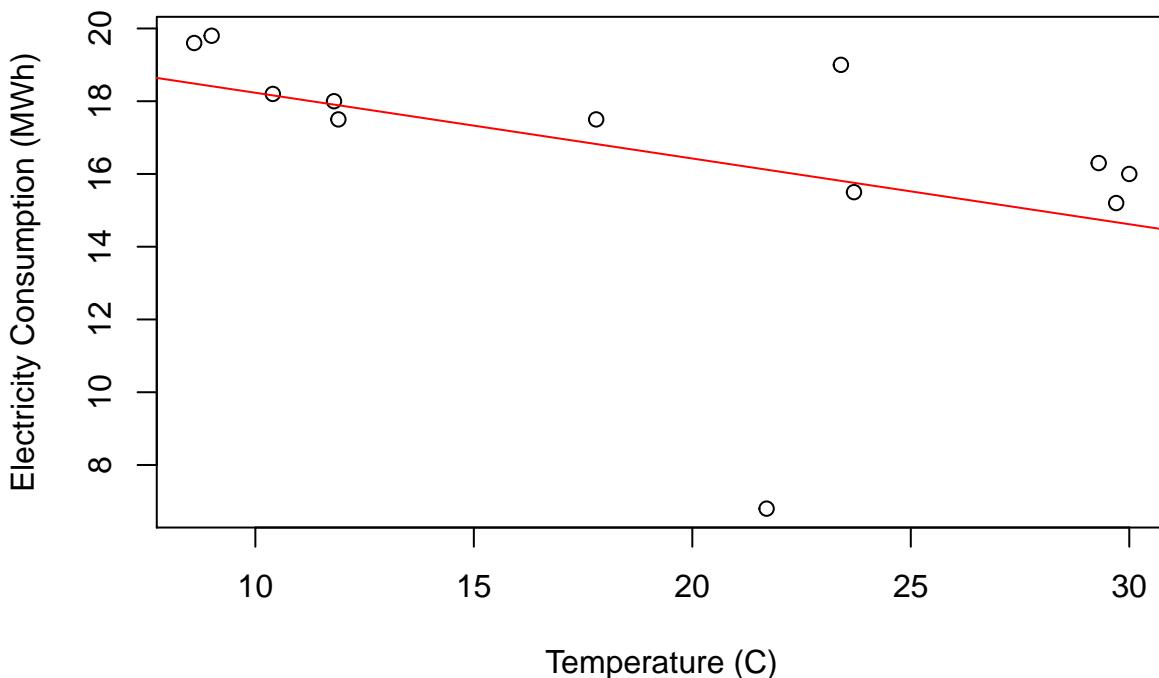
Electricity consumption was recorded for a small town on 12 randomly chosen days. The following maximum temperatures (degrees Celsius) and consumption (megawatt-hours) were recorded for each day.

```
# Day 1 2 3 4 5 6 7 8 9 10 11 12
data.mwh <- c(16.3, 6.8, 15.5, 18.2, 15.2, 17.5, 19.8,
           19, 17.5, 16, 19.6, 18)
data.temp <- c(29.3, 21.7, 23.7, 10.4, 29.7, 11.9,
             9, 23.4, 17.8, 30, 8.6, 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?

```
data4.1 <- data.frame(Mwh = data.mwh, temp = data.temp)
plot(Mwh ~ temp, data = data4.1, main = "Electricity Consumption (MWh) versus Temperature (Celsius)",
     xlab = "Temperature (C)", ylab = "Electricity Consumption (MWh)")
fit <- lm(Mwh ~ temp, data = data4.1)
abline(fit, col = "red")
```

Electricity Consumption (MWh) versus Temperature (Celsius)

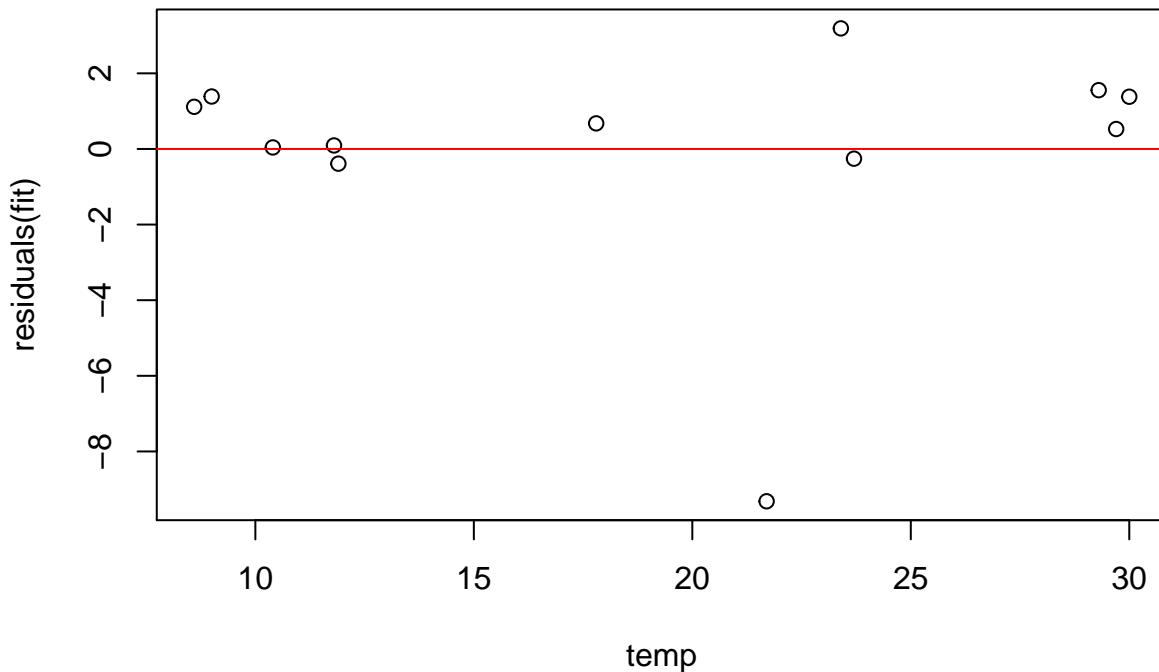


Answer: As we can see on the above plot, there is a negative relationship between consumption of energy and temperature. There is an increase in usage of electricity when the temperature is lower due to homes turning on their heat (I guess the heat systems in these homes are not based on gas, bad data, etc)

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

```
plot(residuals(fit) ~ temp, data = data4.1, main = "Residuals Plot")
abline(0, 0, col = "red")
```

Residuals Plot

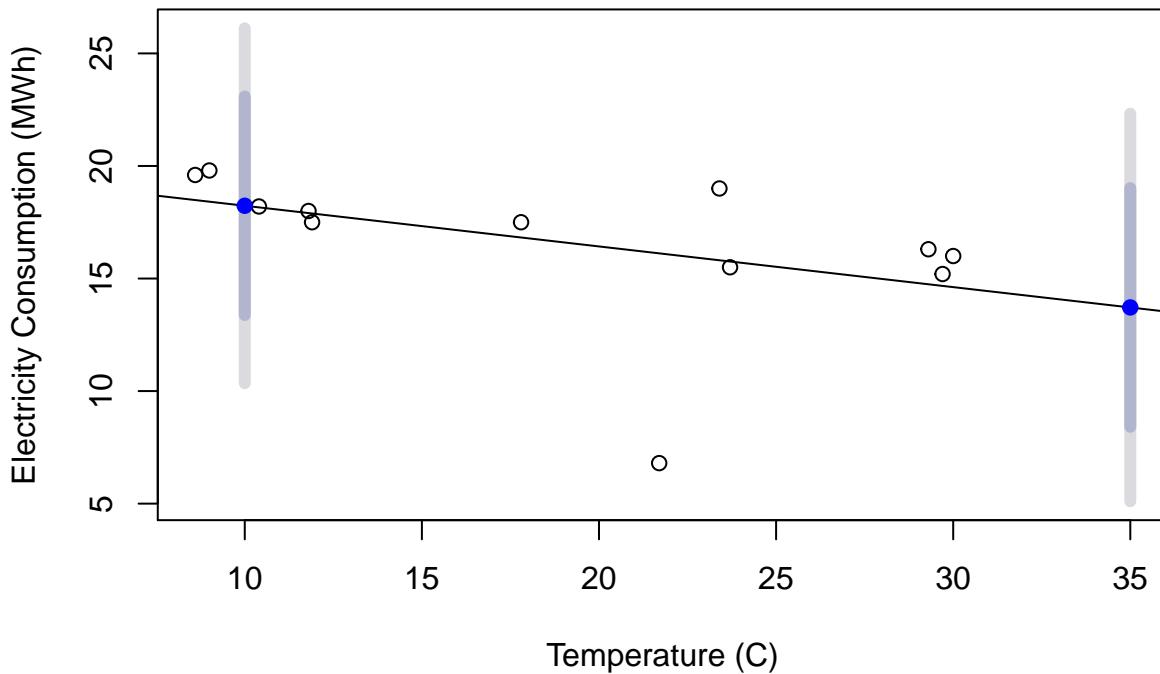


Answer: There are some outliers in the data for sure, but otherwise the graph looks like there no systemic patterns so the model should be adequate. If you look at Temperature approx 23, we get to very different numbers for consumption. This could be due to a lot of reasons (i.e, the lower consumption day could have landed on a holiday where many families travelled away, etc).

- c) Use the model to predict the electricity consumption that you would expect for a day with maximum temperature 10 degrees and a day with maximum temperature 35 degree. Do you believe these predictions?
- d) Give prediction intervals for your forecasts.

```
# Generate a prediction for temp = 10 and 35
forecast <- forecast(fit, newdata = data.frame(temp = c(10,
  35)))
plot(forecast, xlab = "Temperature (C)", ylab = "Electricity Consumption (MWh)")
```

Forecasts from Linear regression model



```
# Show the actual intervals for each prediction
forecast
```

```
##   Point Forecast      Lo 80      Hi 80      Lo 95      Hi 95
## 1     18.23241 13.378919 23.08591 10.351355 26.11347
## 2     13.71495  8.413211 19.01668  5.106039 22.32385
```

Answer: I think I would believe these predictions, but we should be leary as we get further away from the data set we have. The prediction we have for 10 is well within the historical data, but as we extrapolate further out to 35, it looks like the model might be getting inaccurate.

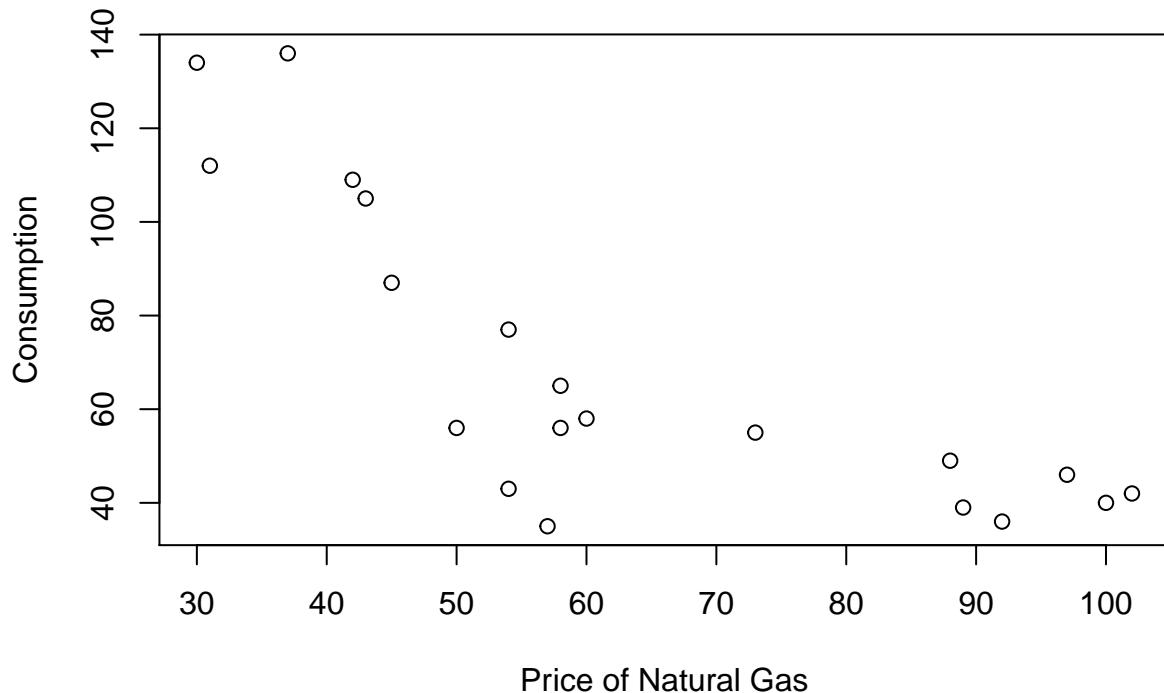
Question 5.2

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.

- Do a scatterplot of consumption against price. The data are clearly not linear. Three possible nonlinear models for the data are given below; The second model divides the data into two sections, depending on whether the price is above or below 60 cents per 1,000 cubic feet.

```
plot(consumption ~ price, data = texasgas, main = "Consumption versus Price of Natural Gas In Texas, 1969",
      xlab = "Price of Natural Gas", ylab = "Consumption")
```

Consumption versus Price of Natural Gas In Texas, 1969



- b) Can you explain why the slope of the fitted line should change with P?

Answer: The derivative of this graph is the rate of consumption, which I do not think would be constant. This means the consumption curve of our model cannot be a simple line, because its derivative should not be constant. Why? Well consumption of natural gas is probably higher when prices are low due to over consumption, and lower after hitting a threshold price where people would rather do without natural gas than pay a high price. Generally speaking, there is a relationship between a good's price and the rate at which it is consumed.

- c) Fit the three models and find the coefficients, and residual variance in each case. For the second model, the parameters a_1, a_2, b_1, b_2 can be estimated by simply fitting a regression with four regressors but no constant: (i) a dummy taking value 1 when $P \leq 60$ and 0 otherwise; (ii) $P_1 = P$ when $P \leq 60$ and 0 otherwise; (iii) a dummy taking value 0 when $P \leq 60$ and 1 otherwise; (iv) $P_2 = P$ when $P > 60$ and 0 otherwise.

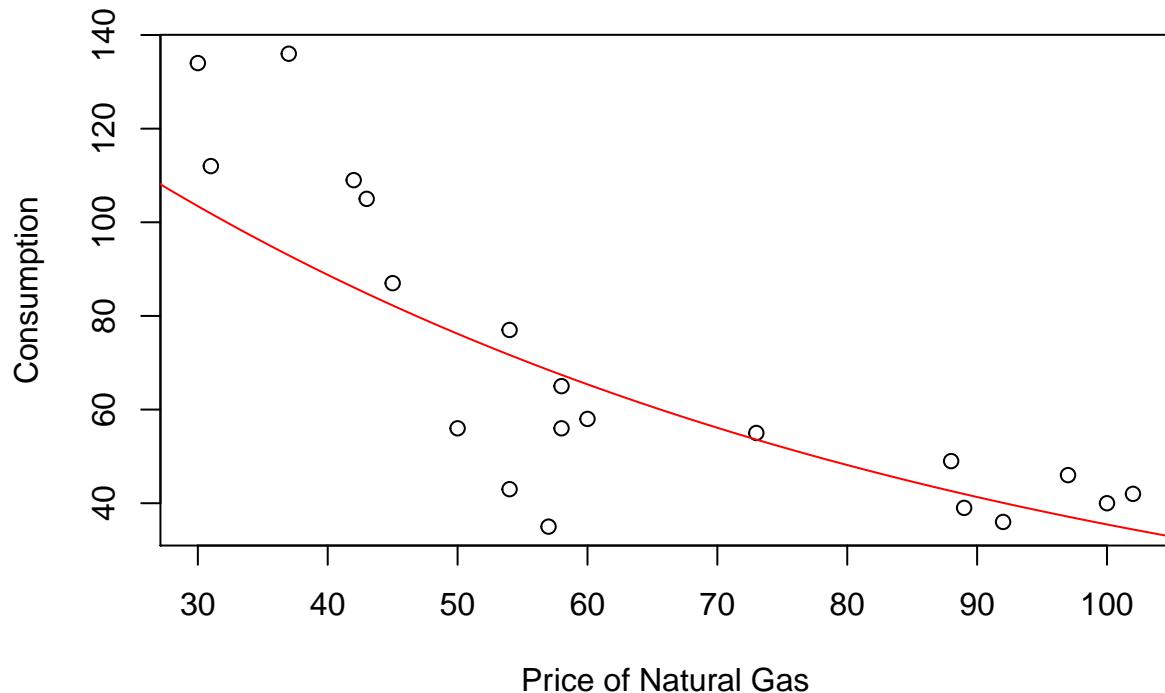
Answer: For each model, I fit them against the data (or constructed predictors) and plot the model, in red, versus the real data, in black.

```
prices <- seq(20, 110, by = 1)

# Model 1
model1 <- lm(log(consumption) ~ price, data = texasgas)
model1.predict <- function(input) {
  return(exp(model1$coef["price"] * input + model1$coef["(Intercept)"]))
}

# Plot Model1 and Data
plot(consumption ~ price, data = texasgas, main = "(Model 1) Consumption versus Price of Natural Gas In Texas, 1969",
     xlab = "Price of Natural Gas", ylab = "Consumption")
lines(prices, model1.predict(prices), col = "red")
```

(Model 1) Consumption versus Price of Natural Gas In Texas, 1969



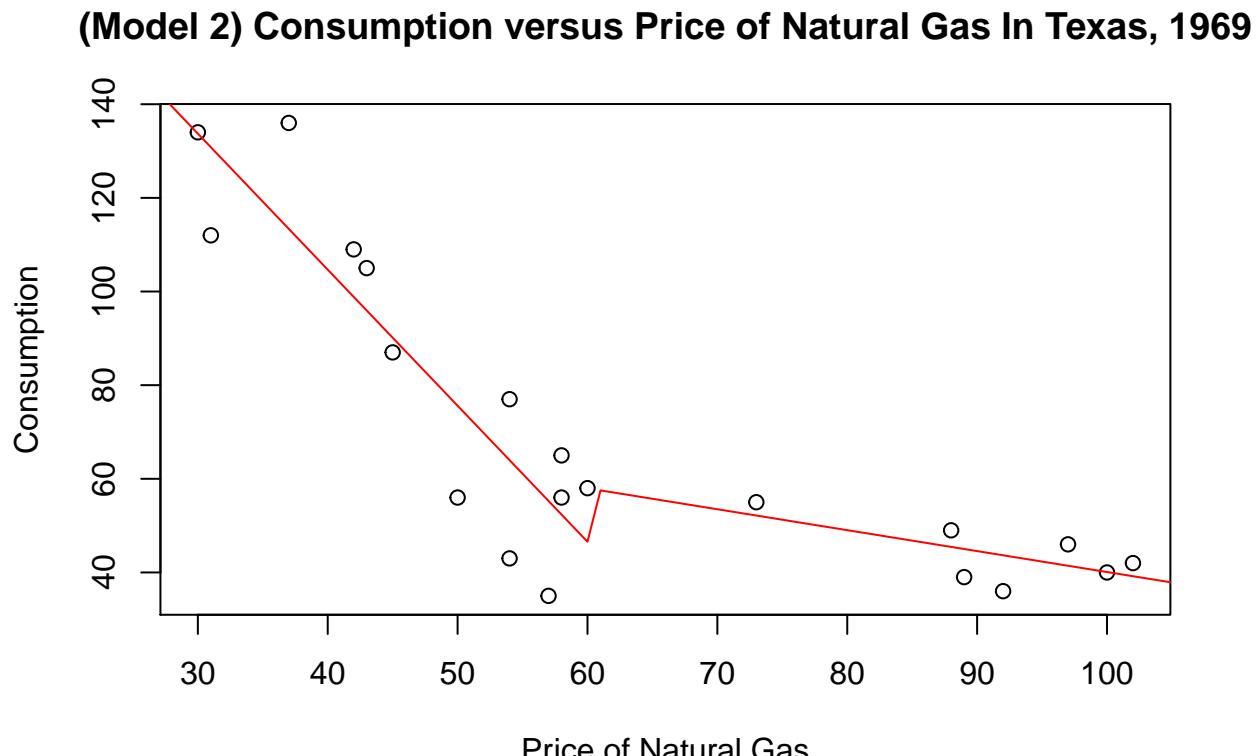
```

# Need some predictors for model 2
texasgas$priceLess60 <- ifelse(texasgas$price <= 60,
                                texasgas$price, 0)
texasgas$dummy1 <- as.numeric(texasgas$priceLess60 >
                                0)
texasgas$priceGreater60 <- ifelse(texasgas$price >
                                60, texasgas$price, 0)
texasgas$dummy2 <- as.numeric(texasgas$priceGreater60 >
                                0)

# Model 2
model2 <- lm(consumption ~ 0 + priceLess60 + dummy1 +
               priceGreater60 + dummy2, data = texasgas)
model2.predict <- function(input) {
  return(ifelse(input <= 60, model2$coef["priceLess60"] *
    input + model2$coef["dummy1"], model2$coef["priceGreater60"] *
    input + model2$coef["dummy2"]))
}

# Plot model2 and data
plot(consumption ~ price, data = texasgas, main = "(Model 2) Consumption versus Price of Natural Gas In Texas, 1969",
      xlab = "Price of Natural Gas", ylab = "Consumption")
lines(prices, model2.predict(prices), col = "red")

```



```

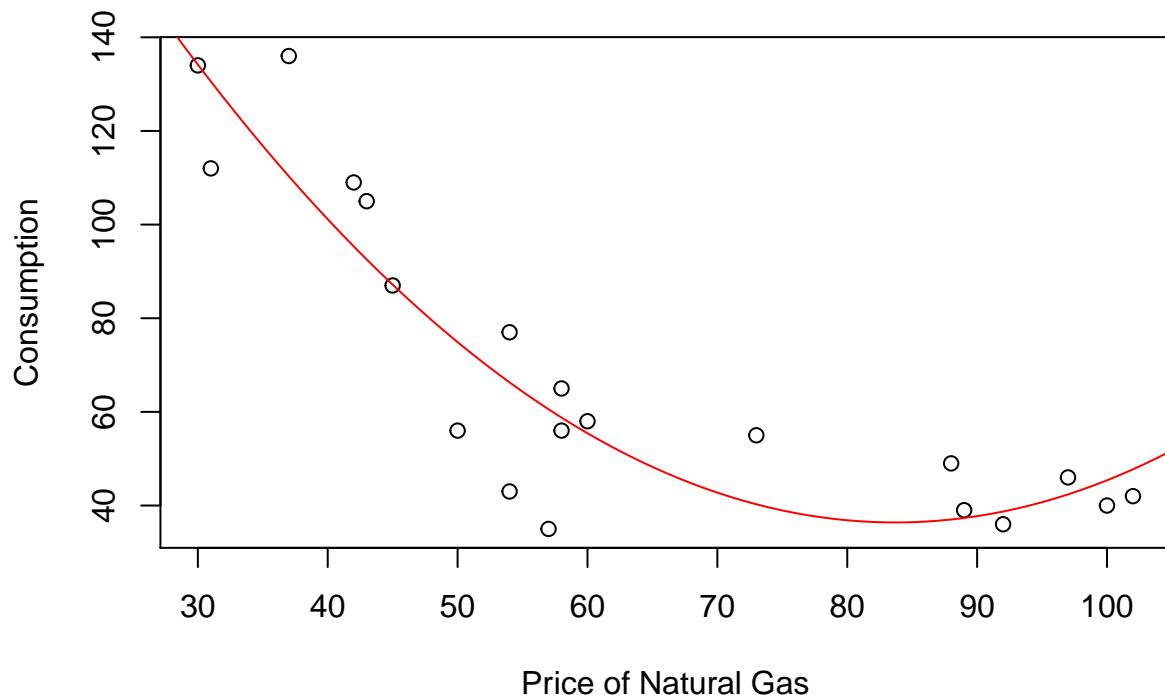
# We need a non-linear predictor for model3
texasgas$price_squared <- texasgas$price^2

# Model 3
model3 <- lm(consumption ~ price + price_squared, data = texasgas)
model3.predict <- function(input) {
  return(model3$coef["price"] * input + model3$coef["price_squared"] *
    input^2 + model3$coef[["(Intercept)"]])
}

# Plot Model and data
plot(consumption ~ price, data = texasgas, main = "(Model 3) Consumption versus Price of Natural Gas In Texas, 1969",
      xlab = "Price of Natural Gas", ylab = "Consumption")
lines(prices, model3.predict(prices), col = "red")

```

(Model 3) Consumption versus Price of Natural Gas In Texas, 1969



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

Answer: I am a bit surprised: From the R squared and AIC measures, Model 2 would be considered the best. Eyeballing it, I probably would have picked Model 3, which goes to show you why you don't eyeball these matters. Model2 is interesting because it is jagged at the inflection point of P=60. Predictions around here are going to take a discontinuous jump around this point.

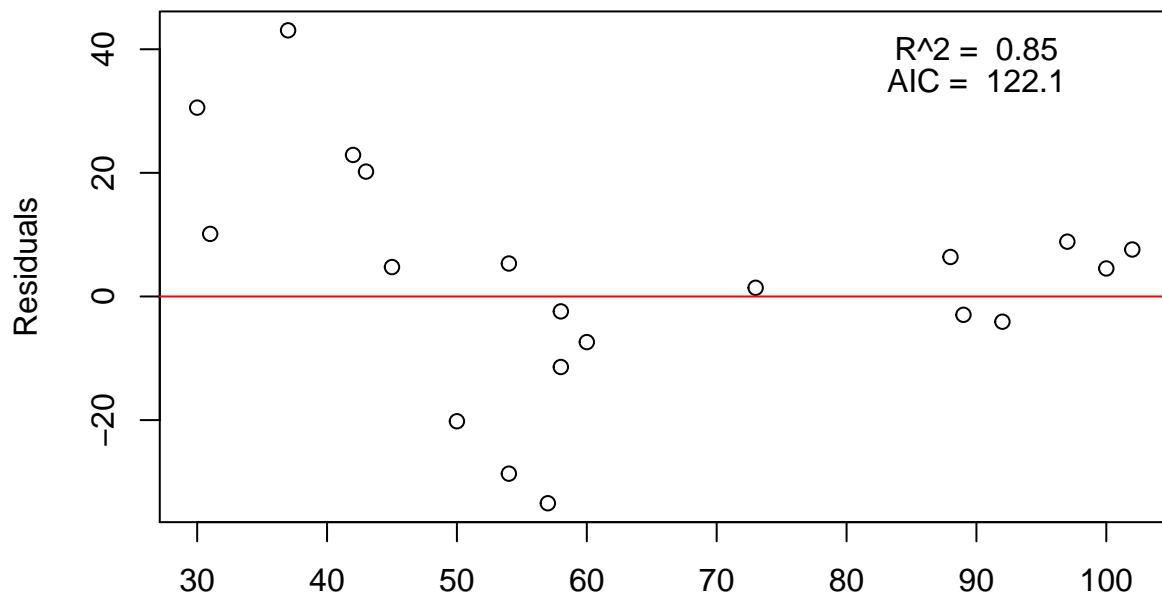
The residual plots all look random to me, so from that metric these models are adequate. Model two seems to be the best overall pick from looking at the R² and AIC metrics.

```
N <- nrow(texasgas)
model1.residuals <- texasgas$consumption - model1.predict(texasgas$price)
model1.rsquared <- cor(texasgas$consumption, model1.predict(texasgas$price))
model1.SSE <- sum(model1.residuals^2)
model1.k <- 1
model1.AIC <- N * log(model1.SSE/N) + 2 * (model1.k +
  2)

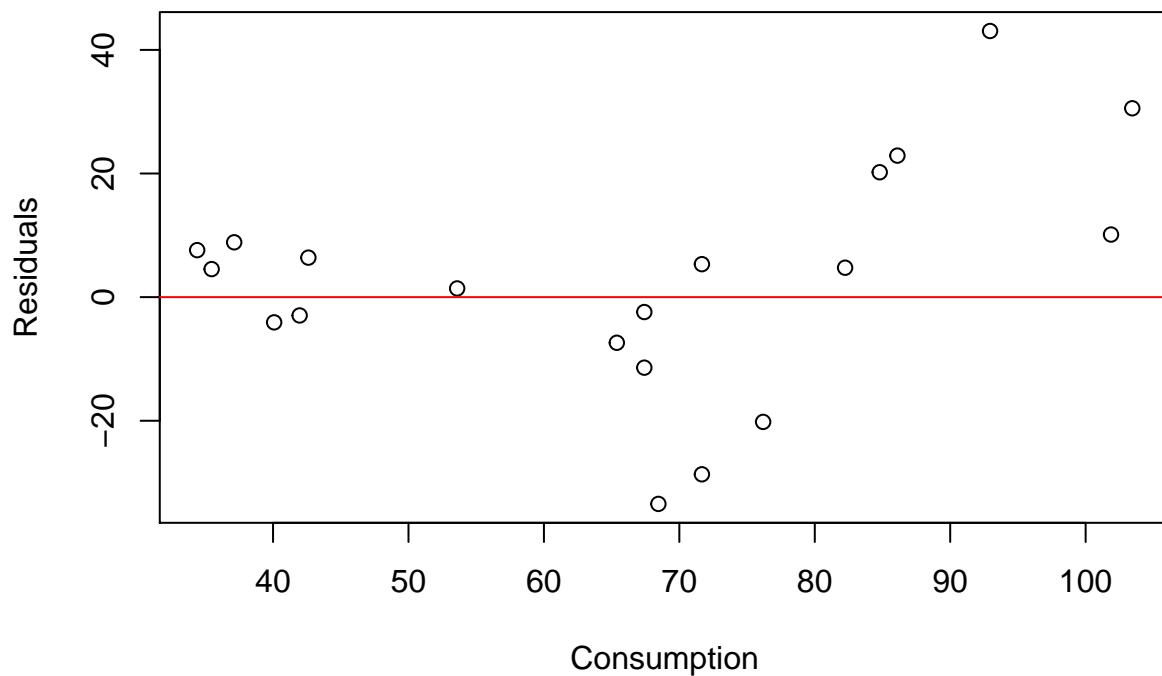
# Plot residuals versus predictor(s)
plot(texasgas$price, model1.residuals, main = "Residuals versus Price (model1)",
      xlab = "Price", ylab = "Residuals")
text(90, 40, labels = c(paste("R^2 = ", round(model1.rsquared,
      digits = 2))))
text(90, 35, labels = c(paste("AIC = ", round(model1.AIC,
      digits = 2))))
abline(0, 0, col = "red")

# Plot residuals versus predicted consumption
plot(model1.predict(texasgas$price), model1.residuals,
      main = "Residuals versus Consumption (model1)",
      xlab = "Consumption", ylab = "Residuals")
abline(0, 0, col = "red")
```

Residuals versus Price (model1)



Residuals versus Consumption (model1)



```

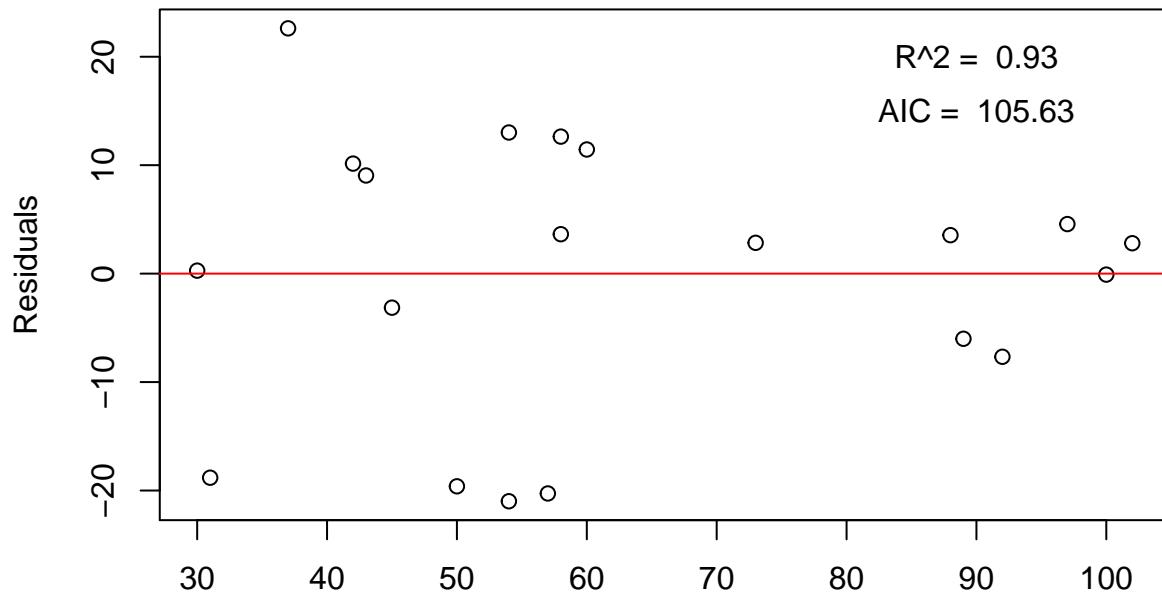
model2.residuals <- resid(model2)
model2.rsquared <- cor(texasgas$consumption, model2.predict(texasgas$price))
model2.SSE <- sum(model2.residuals^2)
model2.k <- 1
model2.AIC <- N * log(model2.SSE/N) + 2 * (model2.k +
  2)

# Plot residuals versus predictor(s)
plot(texasgas$price, model2.residuals, main = "Residuals versus Price (model2)",
  xlab = "Price", ylab = "Residuals")
text(90, 20, labels = c(paste("R^2 = ", round(model2.rsquared,
  digits = 2))))
text(90, 15, labels = c(paste("AIC = ", round(model2.AIC,
  digits = 2))))
abline(0, 0, col = "red")

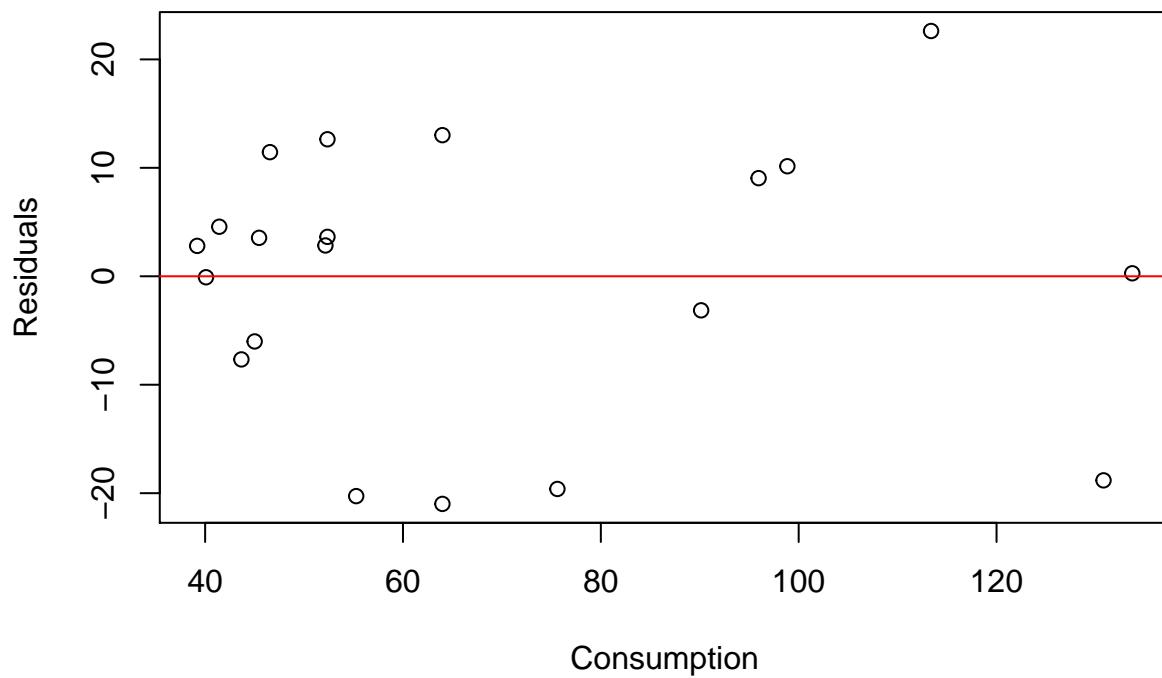
# Plot residuals versus predicted consumption
plot(model2.predict(texasgas$price), model2.residuals,
  main = "Residuals versus Consumption (model1)",
  xlab = "Consumption", ylab = "Residuals")
abline(0, 0, col = "red")

```

Residuals versus Price (model2)



Residuals versus Consumption (model1)^{Price}



```

model3.residuals <- resid(model3)
model3.SSE <- sum(model3.residuals^2)
model3.k <- 2 # Two predictors
model3.rsquared <- cor(texasgas$consumption, model3.predict(texasgas$price))
model3.AIC <- N * log(model3.SSE/N) + 2 * (model3.k +
  2)

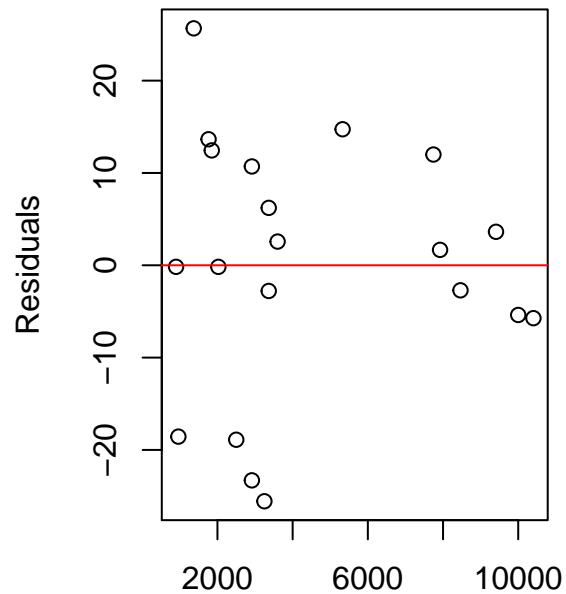
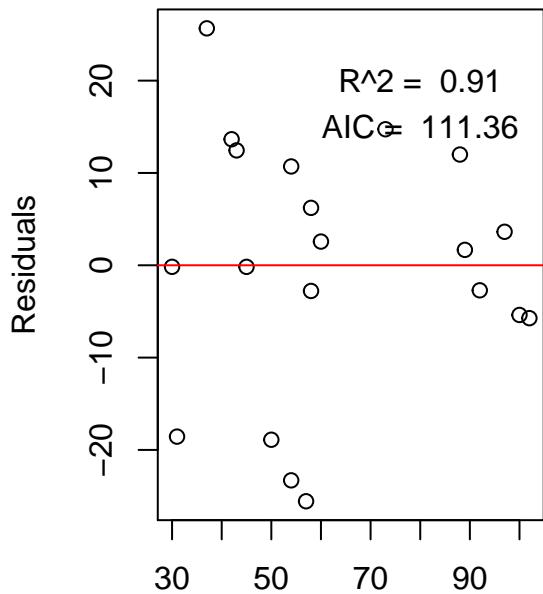
# Plot residuals versus predictor(s)
par(mfrow = c(1, 2))
plot(texasgas$price, model3.residuals, main = "Residuals versus Price (model3)",
  xlab = "Price", ylab = "Residuals")
text(80, 20, labels = c(paste("R^2 = ", round(model3.rsquared,
  digits = 2))))
text(80, 15, labels = c(paste("AIC = ", round(model3.AIC,
  digits = 2))))
abline(0, 0, col = "red")

plot(texasgas$price_squared, model3.residuals, main = "Residuals versus Price Squared (model3)",
  xlab = "Price Squared", ylab = "Residuals")
abline(0, 0, col = "red")

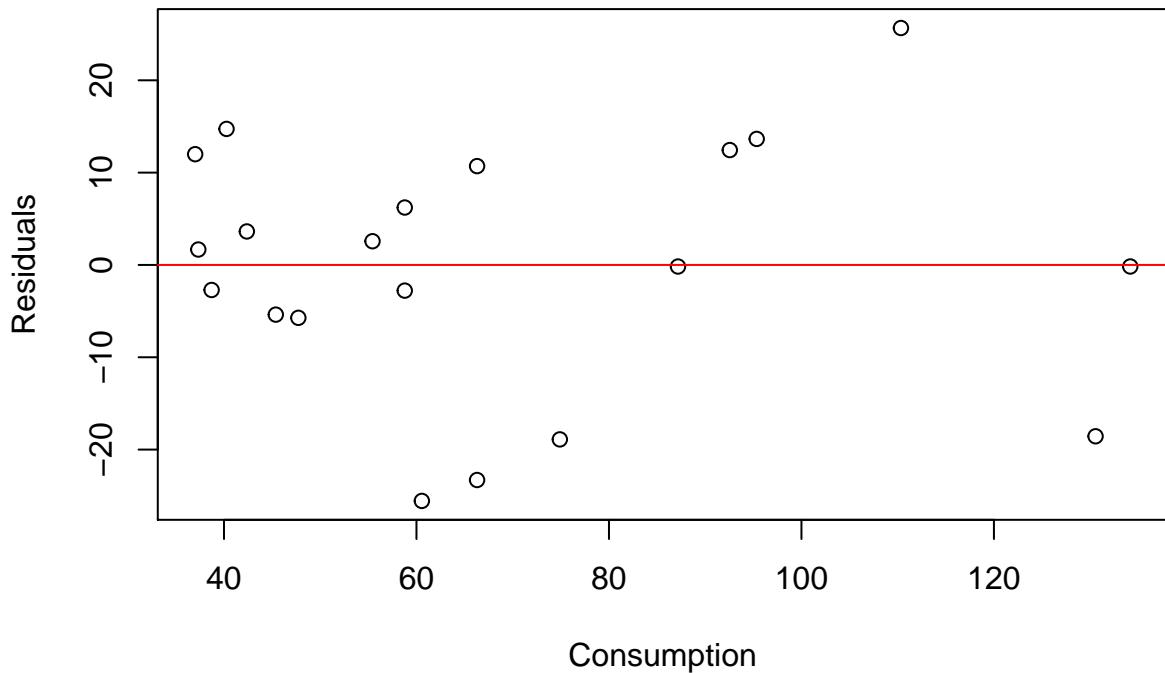
# Plot residuals versus predicted consumption
par(mfrow = c(1, 1))
plot(model3.predict(texasgas$price), model3.residuals,
  main = "Residuals versus Consumption (model3)",
  xlab = "Consumption", ylab = "Residuals")
abline(0, 0, col = "red")

```

Residuals versus Price (model3)



Residuals versus Consumption (model3)



- f) 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.

```

input_prices <- c(40, 60, 80, 100, 120)
predictions <- model2.predict(input_prices)
predictions

## [1] 104.66623 46.55289 49.02913 40.08989 31.15065

```

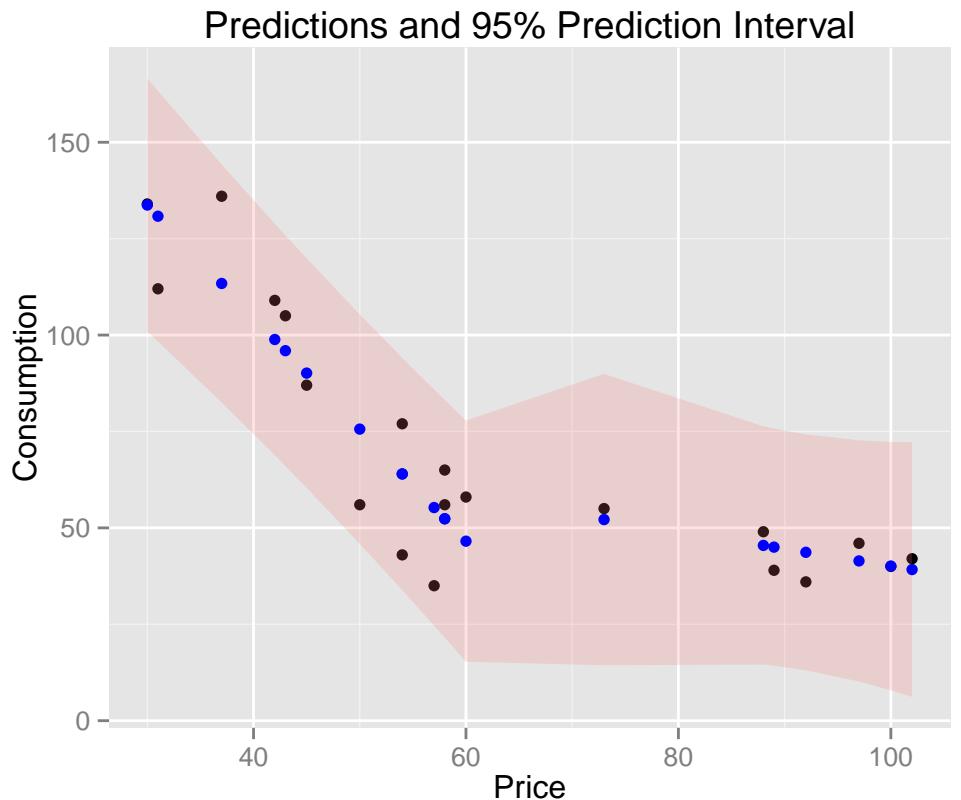
- g) Compute 95% prediction intervals. Make a graph of these prediction intervals and discuss their interpretation.

```

# Print intervals w/ predictions
texasgas_predict <- data.frame(texasgas, predict(model2,
    interval = "prediction"))

## Warning in predict.lm(model2, interval = "prediction"): predictions on current data refer to _future_
ggplot(texasgas_predict, aes(x = price, y = consumption)) +
  geom_point() + geom_ribbon(aes(y = fit, ymin = lwr,
  ymax = upr, fill = "prediction"), alpha = 0.2) +
  geom_point(aes(y = fit), colour = "blue") + ggtitle("Predictions and 95% Prediction Interval") +
  xlab("Price") + ylab("Consumption")

```



- h) 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?

Answer: As we can see below, the correlation is very high. Generally speaking, having two predictors that are highly correlated can pose problems for some machine learning models, so care must be taken when adding non-linear terms.

```
cor(texasgas$price, texasgas$price_squared)
```

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

6.2. Developing a model to predict permeability (see Sect. 1.4) could save significant resources for a pharmaceutical company, while at the same time more rapidly identifying molecules that have a sufficient permeability to become a drug:

- (a) Start R and use these commands to load the data: snip ... The matrix fingerprints contains the 1,107 binary molecular predictors for the 165 compounds, while permeability contains permeability response.
- (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?

```
nearZero <- nearZeroVar(fingerprints)
length(nearZero) # 719

## [1] 719

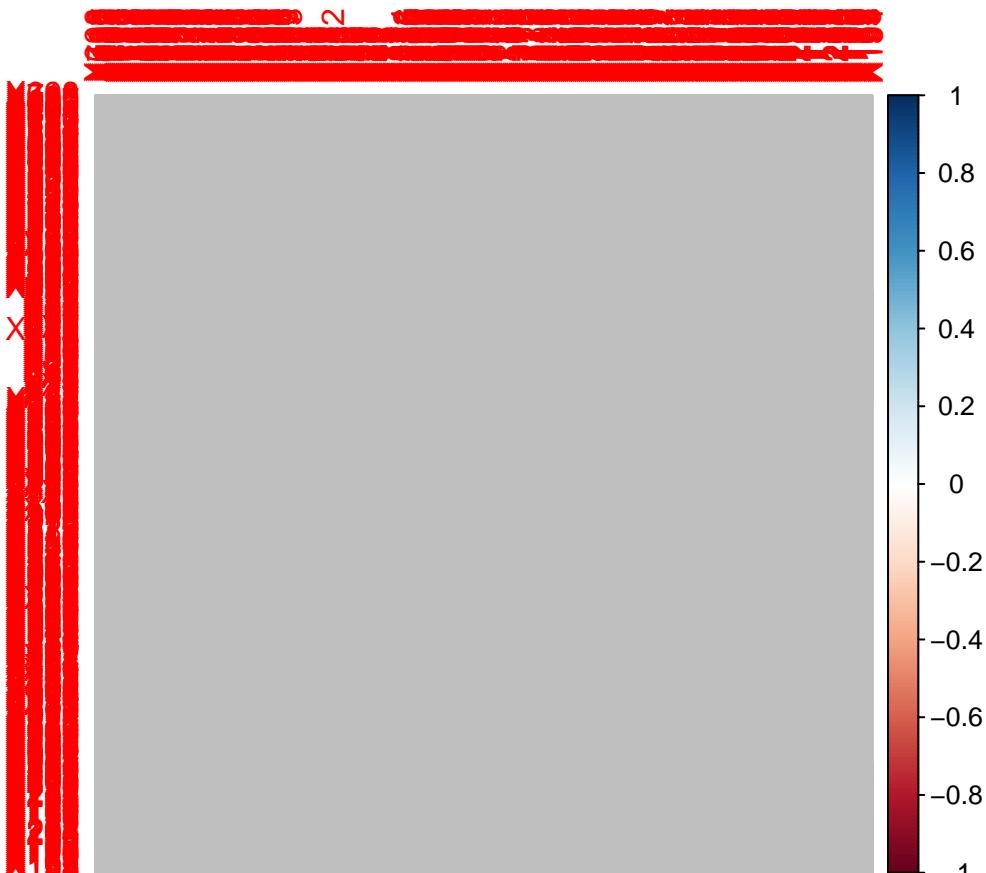
# Filter out these predictors
fingerprints.filtered <- fingerprints[, -nearZero]
```

Seems like there are 719 predictors that we can drop because they have near zero variance. This leaves us with 388 predictors.

- (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 ?

First, lets look at pairwise correlation amongst the predictors:

```
# Look at correlation matrix (looks ugly)
correlations <- cor(fingerprints.filtered)
corrplot::corrplot(correlations, order = "hclust")
```

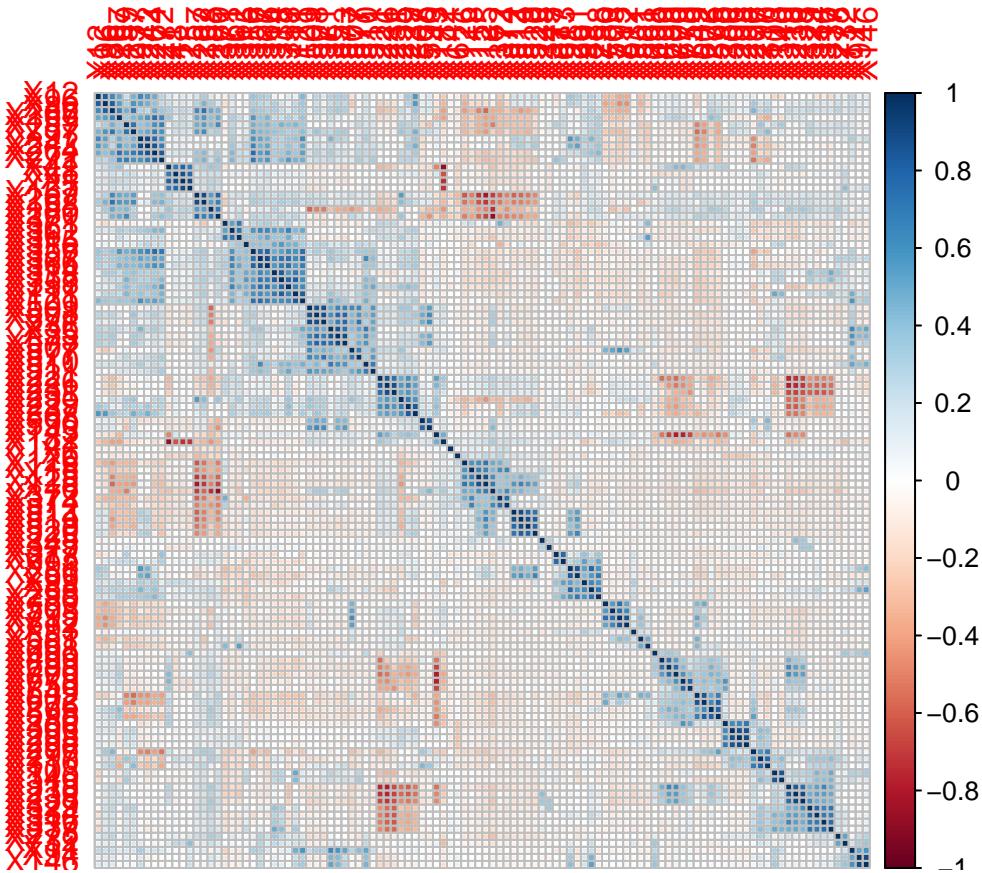


Thats a pretty awful looking chart but we do see significant number of predictors that are correlated(at least I see it in R; the PDF shows nothing). Lets remove them with threshold .90 (I played with this value and going to far with it lowers the R^2 and brings up RMSE):

```
# Find highly correlated predictors and remove
highCorr <- findCorrelation(correlations, cutoff = 0.9)
fingerprints.filtered <- fingerprints.filtered[, -highCorr]
```

The correlation matrix looks much better now:

```
# Much better
correlations <- cor(fingerprints.filtered)
corrplot::corrplot(correlations, order = "hclust")
```



```

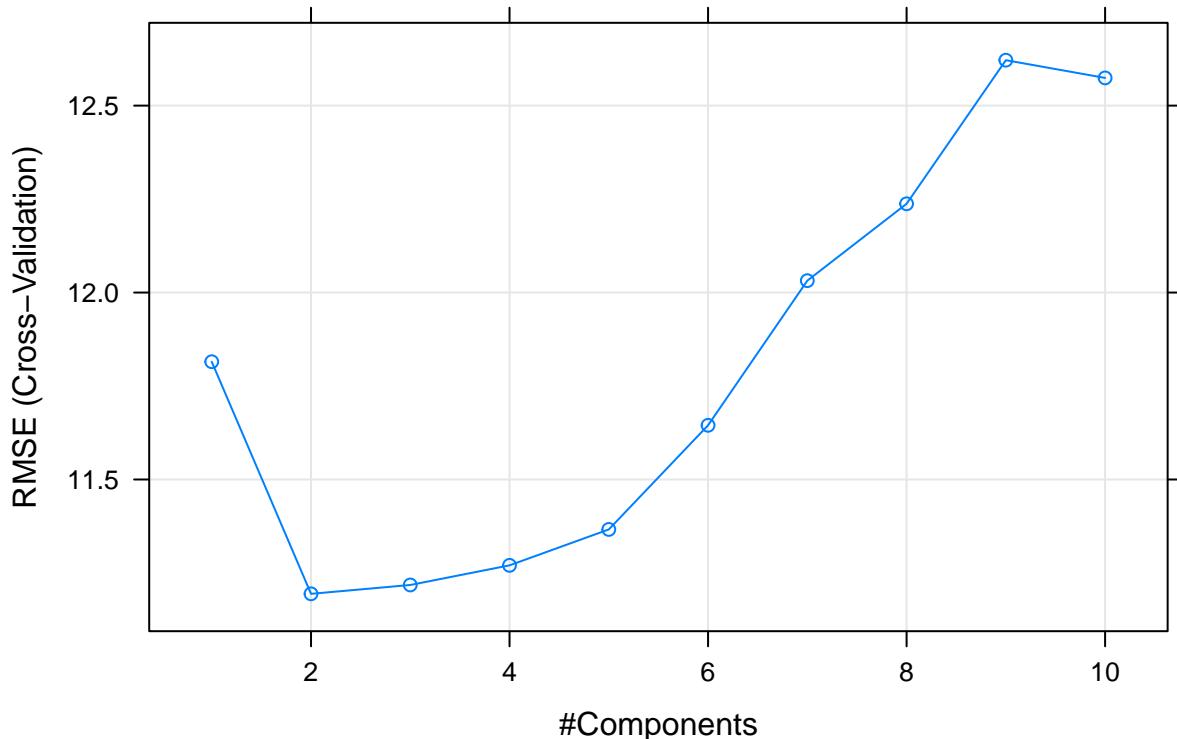
# Split the data into a training and test set
fingerprints.train <- fingerprints.filtered[1:124,
    ]
fingerprints.test <- fingerprints.filtered[1:124, ]
permeability.train <- permeability[1:124, ]
permeability.test <- permeability[1:124, ]

# Train a PLS Model
model.pls <- train(fingerprints.train, permeability.train,
    method = "pls", tuneLength = 10, trControl = trainControl(method = "cv"))

# Plot PLS Model to see at which # of components we
# minimize error
plot(model.pls, main = "RMSE Error of PLS Model vs Number of Components")

```

RMSE Error of PLS Model vs Number of Components



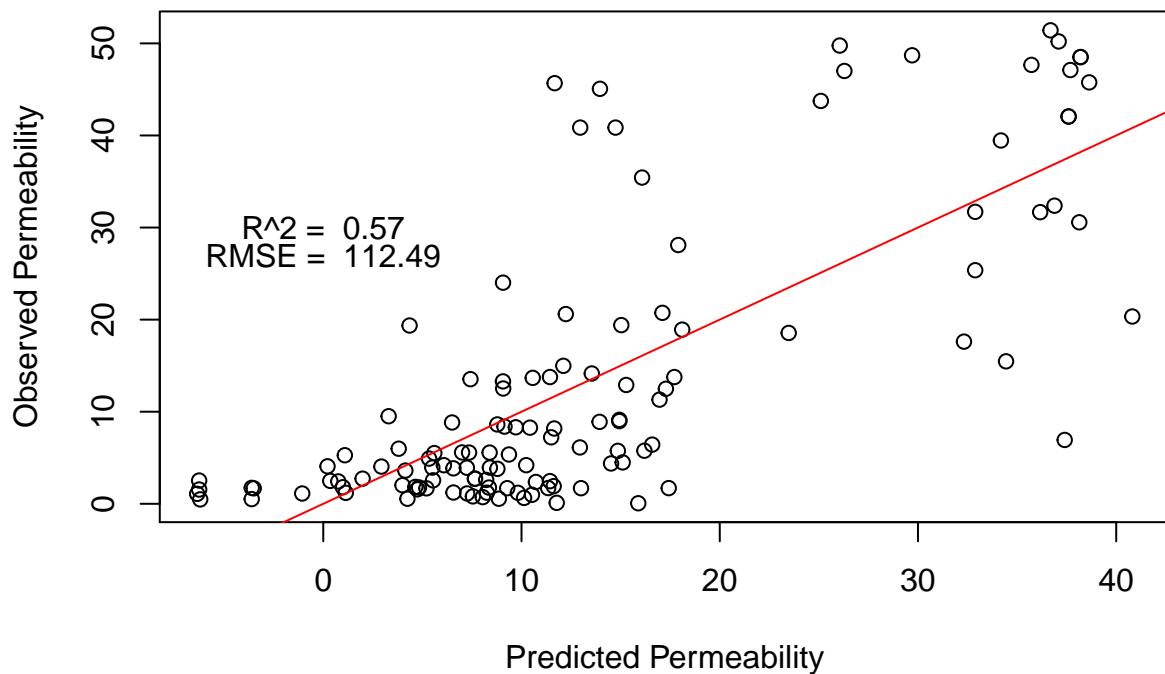
```
# What is the R^2 value?  
# model.pls$results[model.pls$results$ncomp == 5,  
# 'Rsquared'] # 0.5385791
```

We see from the above plot that the number of components is 5 when we minimize error, with an R^2 value of 0.5385791.

(d) Predict the response for the test set. What is the test set estimate of R^2 ?

```
# Predict the permeability based on test set of  
# fingerprints  
permeability.predict = predict(model.pls, fingerprints.test)  
  
# Print out predictions and outcomes  
plot(permeability.predict, permeability.test, main = "Observed versus Predicted Permeability from PLS M  
xlab = "Predicted Permeability", ylab = "Observed Permeability")  
abline(0, 1, col = "red")  
text(0, 30, paste("R^2 = ", round(cor(permeability.test,  
permeability.predict)^2, 2)))  
text(0, 27, paste("RMSE = ", round(sqrt(sum((permeability.test -  
permeability.predict)^2)), 2)))
```

Observed versus Predicted Permeability from PLS Model (n=3)



The graph above shows a plot of observed values versus predicted values for permeability. The red line shown is the $y = x$ line: the closer the values are to the red line, the better. R^2 is only 0.5830192

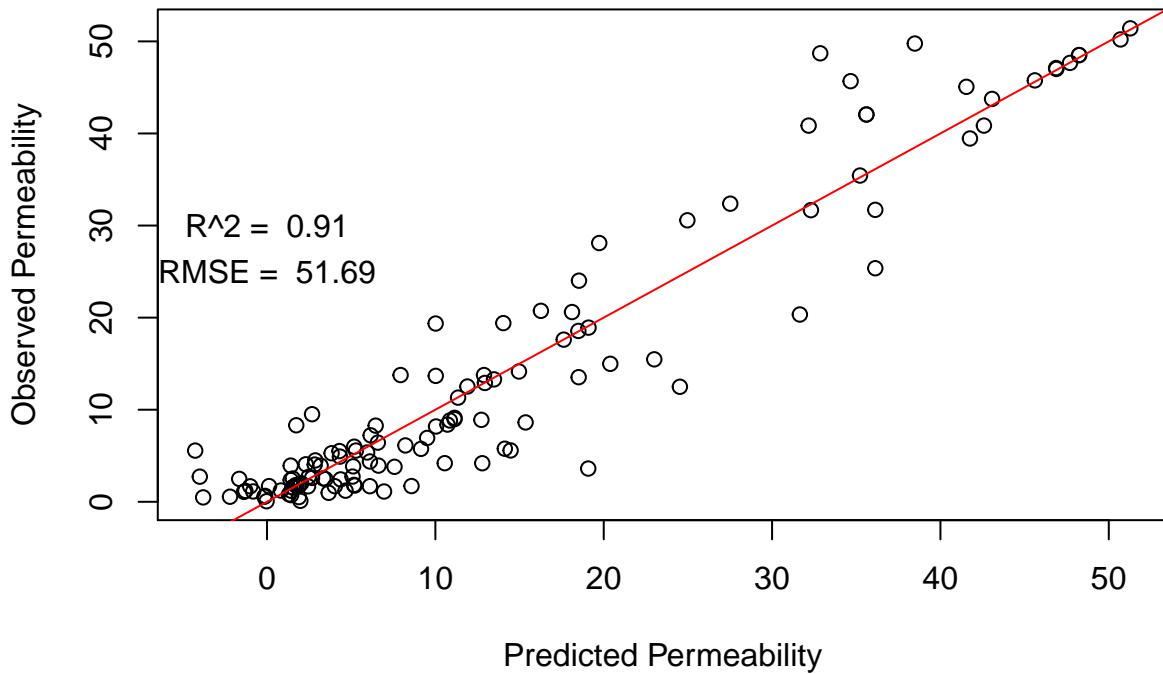
- (e) Try building other models discussed in this chapter. Do any have better predictive performance?

Using some ideas from (this link)[\[http://machinelearningmastery.com/penalized-regression-in-r/\]](http://machinelearningmastery.com/penalized-regression-in-r/), I'll use glmnet to build a model penalizing both the L1-norm and L2-norm.

```
glm.model <- glmnet(fingerprints.train, permeability.train,
  family = "gaussian", alpha = 0.5, lambda = 0.001)

permeability.glm.predict <- predict(glm.model, fingerprints.test)
plot(permeability.glm.predict, permeability.test, main = "Observed versus Predicted Permeability from GLMNET",
  xlab = "Predicted Permeability", ylab = "Observed Permeability")
abline(0, 1, col = "red")
text(0, 30, paste("R^2 = ", round(cor(permeability.test,
  permeability.glm.predict)^2, 2)))
text(0, 25, paste("RMSE = ", round(sqrt(sum((permeability.test -
  permeability.glm.predict)^2)), 2)))
```

Observed versus Predicted Permeability from GLM Model



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

If we were to recommend a model, it would be the GLM Net model (ie, elasticnet), since it has a high R^2 . However, since mistakes in the model may overlook good candidates, I would make sure to tune this as best I could, and confirm the data pre-processing steps taken make sense.