Exercise 1: Use the data lathe1 from problem 5.12 and do the following:

a. As described in 5.12.2, fir the full second order model using log(Life) as the response and speed and feed as the predictors. A full second-order model is one which includes both predictors, second-order polynomial terms in both predicts, and the interaction term between the predicts. Report your fitted model.

Solution:
The full second-order model es defined by,

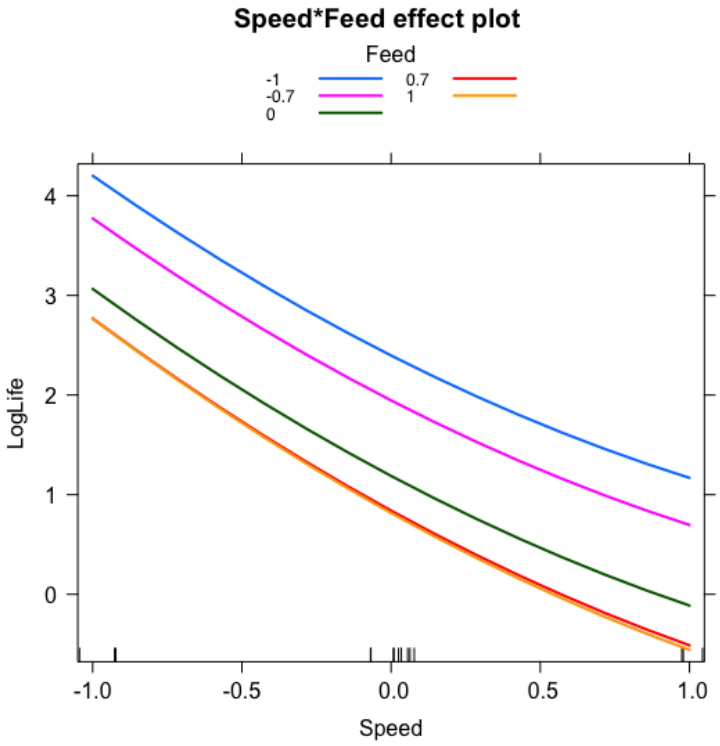
$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X^1 X^2 + \epsilon$$

Fitting the model in r we get the following,
Code:

```
> df <- lathe1

> df$LogLife <- log(df$Life)

> FullSecondOrderModel <- lm(LogLife ~ Speed
                           + Feed
                           + I(Speed^2)
                           + I(Feed^2)
                           + Speed:Feed, data = df)

> summary(FullSecondOrderModel)
Call:
lm(formula = LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2) +
    Speed:Feed, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43349 -0.14576 -0.02494  0.16748  0.47992

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.18809    0.10508  11.307 2.00e-08 ***
Speed        -1.58902    0.08580 -18.520 3.04e-11 ***
Feed         -0.79023    0.08580  -9.210 2.56e-07 ***
I(Speed^2)    0.28808    0.10063   2.863 0.012529 *
I(Feed^2)     0.41851    0.10063   4.159 0.000964 ***
Speed:Feed   -0.07286    0.10508  -0.693 0.499426

---------------------------------------------------------------

Residual standard error: 0.2972 on 14 degrees of freedom
Multiple R-squared:  0.9702,    Adjusted R-squared:  0.9596
F-statistic: 91.24 on 5 and 14 DF,  p-value: 3.551e-10
```

b. Obtain the efedct plot for the interaciton between Speed and Feed in the full second-order model you should see a series of quadratic curves which owing to the insignificance of the interaction term, essentially match one another except separated by vertical shifts.

Solution:
Using r to obtain the effect plot we get,



Code:

```
> plot(Effect(c("Speed", "Feed"), FullSecondOrderModel), multiline = TRUE)
```

c. Test the interaction term in your model. You can use the coefficient's t-test or perform a partial F test using anova(). Does your result agree with what you're effect plot showed.

Solution:
Performing partial F test in r, using the anova() function we get the following,
Code:

```
> SecondOrderModelNoInteraction <- lm(LogLife ~ Speed +
                                                Feed +
                                                I(Speed^2) +
                                                I(Feed^2), data = df)
> anova(FullSecondOrderModel, SecondOrderModelNoInteraction)

Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2) + Speed:Feed
Model 2: LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2)
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     14 1.2367
2     15 1.2791 -1 -0.042466 0.4807   0.4994
```

We can see that with a with a p-value of .4994, adding the interaction to the model does not provide a statistically significant improvement. This does match what our effect plot is telling us. The effect plot above described an inverse relationship between speed on log(life) and feed on log(life), these effect are likely cancelling each other out and we are left with an insignificant parameter.

d. Remove the interaction term and refit the model. Test the quadratic terms in speed an feed to determine if the first order model is adequate.

Solution:
Below is the partial F tests for determining which second order terms should be included in the model. In general all the squared terms contributed on the $\alpha = .05$ significance level to the model. The first test evaluates the second order speed term against the first order model. The second test evaluates the second order feed term against the first order model. The next two test show us that of the two second order terms the feed term is most significant. Finally the last test compares the second order model against the first order model returning a significant improvement. I would say that the second order model is likely better than the first order.
Code:

```
> anova(FirstOrderModel, SecondOrderModelNoFeed)
Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed
Model 2: LogLife ~ Speed + Feed + I(Speed^2)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     17 3.7431
2     16 2.8071  1   0.93605 5.3354 0.03457 *
---------------------------------------------------------
```

```
> anova(FirstOrderModel, SecondOrderModelNoSpeed)
Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed
Model 2: LogLife ~ Speed + Feed + I(Feed^2)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     17 3.7431
2     16 2.0031  1      1.74 13.899 0.00183 **
---------------------------------------------------------
```

```
> anova(SecondOrderModelNoSpeed, SecondOrderModel)
Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed + I(Feed^2)
Model 2: LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2)
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     16 2.0031
2     15 1.2791  1   0.72395 8.4895 0.01069 *
---------------------------------------------------------
```

```
> anova(SecondOrderModelNoFeed, SecondOrderModel)
Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed + I(Speed^2)
Model 2: LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     16 2.8071
2     15 1.2791  1    1.5279 17.918 0.0007232 ***
---------------------------------------------------------
```

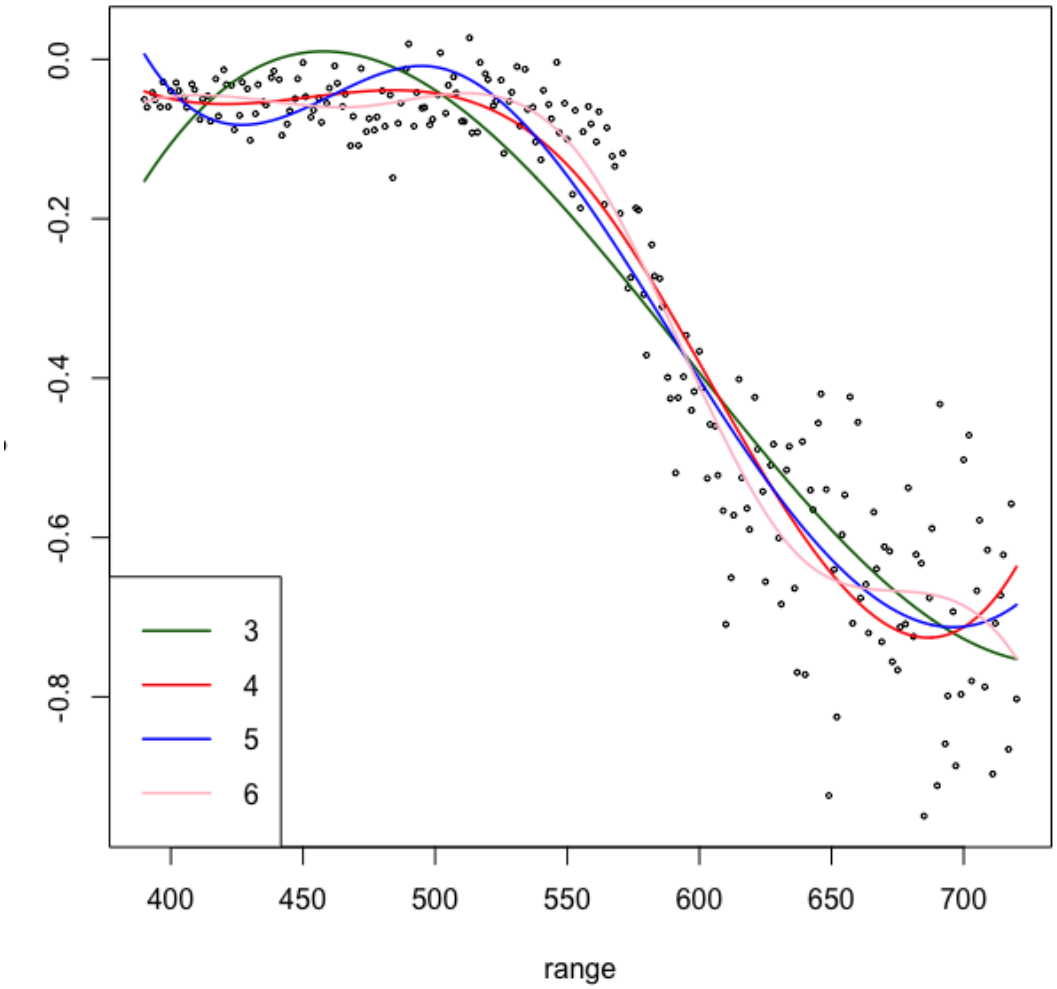
```
> anova(FirstOrderModel,SecondOrderModel)
Analysis of Variance Table

Model 1: LogLife ~ Speed + Feed
Model 2: LogLife ~ Speed + Feed + I(Speed^2) + I(Feed^2)
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     17 3.7431
2     15 1.2791  2     2.464 14.447 0.0003181 ***
---------------------------------------------------------
```

Exercise 2: Use the data frame lidar in the SemiPar library, which you will likely have to install. . . This data frame contains the response logratio and predictor range obtained from 221 observations of a light detection and ragning experiment. Do the following:

a. Fit splines to the data with three, four , five and six degress of freedom. Plot the data with the fitted plines overlaind on top, in a single plot.

Solution:
Fitting the spline, and plotting the data we get,

Figure 1: Spline Plot range vs. logratio



Code:

```
> Spline3 = lm(lidar$logratio ~ bs(lidar$range, df=3))
> Spline4 = lm(lidar$logratio ~ bs(lidar$range, df=4))
> Spline5 = lm(lidar$logratio ~ bs(lidar$range, df=5))
> Spline6 = lm(lidar$logratio ~ bs(lidar$range, df=6))

> plot(logratio ~ range, data = lidar, xlab = 'range', ylab = 'logratio', cex = .4)
> lines(sort(lidar$range),predict(Spline3)[order(lidar$range)], lwd = 1.5,col = "darkgreen")
> lines(sort(lidar$range),predict(Spline4)[order(lidar$range)], lwd = 1.5,col = "red")
> lines(sort(lidar$range),predict(Spline5)[order(lidar$range)], lwd = 1.5,col = "blue")
> lines(sort(lidar$range),predict(Spline6)[order(lidar$range)], lwd = 1.5,col = "pink")
> legend("bottomleft", legend =c("3",'4', '5', '6'), lwd = 1.5, col =c("darkgreen","red", "blue", 'pink'))
```

b. Use a spline with three degrees of freedom to predict the logratio for an observation with range equal to 900, Then do the same for a spline with four degrees of freedom. Which prediction do you trust more and why.

Solution:
Using the predict function in r we can get the prediction for an observation equal to 900. Note that 900 is outside of the range of data used to build each model. In general this level of extrapolation will likely result in a mediocre prediction, especially with polynomial regressions (and splines). That is why I would expect a more bias model with lower variability to perform better with this prediction i.e. the three degrees of freedom,
Code:

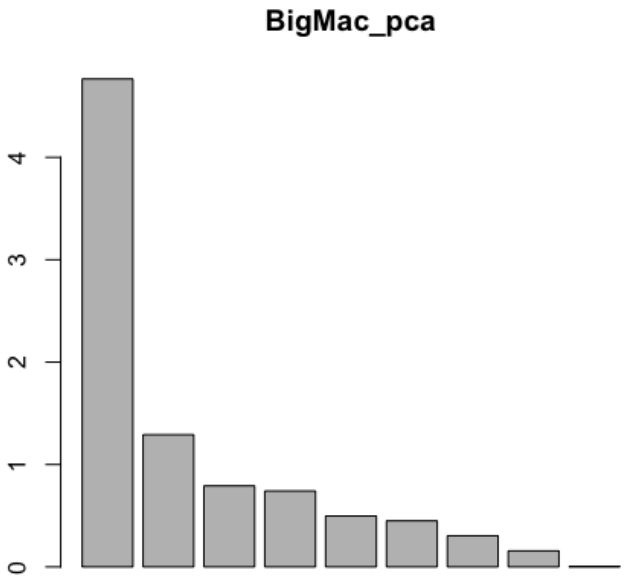
```
> predict(Spline3,
+         newdata = data.frame(range = 900),
+         interval = "prediction",
+         level=.95)
       fit        lwr      upr
1 0.438684 -0.2756517 1.15302
Warning message:
In bs(range, degree = 3L, knots = numeric(0), Boundary.knots = c(390L,
:
  some 'x' values beyond boundary knots may cause ill-conditioned bases

-------------------------------------------------------------------------
> predict(Spline4,
+         newdata = data.frame(range = 900),
+         interval = "prediction",
+         level=.95)
       fit      lwr      upr
1 5.236268 3.860862 6.611674
Warning message:
In bs(range, degree = 3L, knots = c(`50%` = 555), Boundary.knots = c(390L,
:
  some 'x' values beyond boundary knots may cause ill-conditioned bases
```

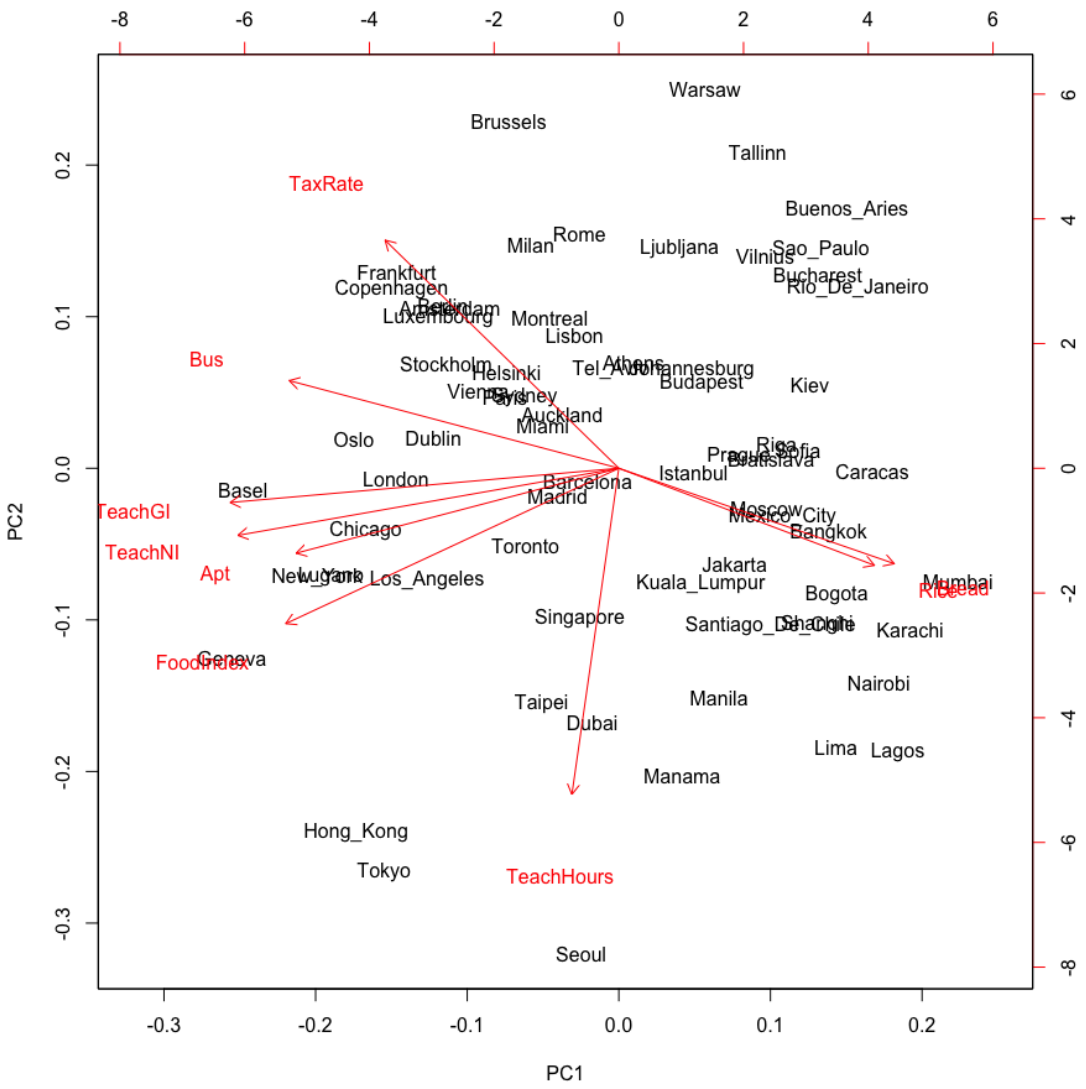
Exercise 3: Use the data frame BigMac2003 in alr4, whcih contains the respoonse Bigmac, of the price of a bigmac in various world cities in 2003, expressed in minutes of labor. It also contains quantitative predictors Bread, Rice, Bus, and Apt, which are prices of other goods, as well as economic indicators FoodIndex, TeachGI, TeachNI, TaxRate, and TeachHours.

a. Use principal component analysis on the nine predictors. Report your scree plot and biplot.

Solution:
Consider the following r code,
Code:

```
> BigMac_pca <- prcomp(~Bread + Rice + FoodIndex +
                         Bus + Apt + TeachGI +
                         TeachNI + TaxRate + TeachHours,
                         scale.=TRUE, data= df)
> plot(BigMac_pca)
> biplot(BigMac_pca)
```




Warsaw
Brussels
Tallinn
TaxRate
Buenos_Aries
Milan
Rome
Ljubljana
Vilnius
Sao_Paulo
Frankfurt
Bucharest
Copenhagen
Rio_De_Janeiro
Montreal
Lisbon
Bus
Stockholm
Helsinki
Budapest
Kiev
Auckland
Miami
Oslo
Dublin
Istanbul
Caracas
Basel
London
Barcelona
Madrid
TeachGI
Moscow
Chicago
Bangkok
TeachNI
Toronto
Jakarta
Apt
Los_Angeles
Kuala_Lumpur
Bogota
Singapore
Karachi
Geneva
Nairobi
Taipei
Manila
Dubai
Lima
Lagos
Manama
Hong_Kong
Tokyo
TeachHours
Seoul
PC1
PC2

b. How many principal components are necessary to use in order to account for at least 90 percent of the variance in the prediction.

Solution:
Using the summary command in r, we can see the cumulative proportion of variance explained for each PCA,
Code:

```
> summary(BigMac_pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5
Standard deviation     2.1831 1.1364 0.88955 0.86104 0.70474
Proportion of Variance 0.5295 0.1435 0.08792 0.08238 0.05518
Cumulative Proportion  0.5295 0.6730 0.76094 0.84332 0.89850
      PC6     PC7     PC8     PC9
  0.67058 0.55096 0.39552 0.06183
  0.04996 0.03373 0.01738 0.00042
  0.94846 0.98219 0.99958 1.00000
```

In order to account for 90 percent of the variance, we should include 6 principle components(we might be able to get away with 5 minding rounding error).

c. Fit the MLR model with the response BigMac and the first four principal components as regressors. Then form the model that contains all nine original predictors. Compare the coefficients of determination for the two models. Are you satisfied that the model with fewer regressors fits sufficiently well compared to the full model?

Solution:
Fitting the models in r we get,
Code:

```
> summary(lm(BigMac~ BigMac_pca$x[, 1:4], data = df))
Call:
lm(formula = BigMac ~ BigMac_pca$x[, 1:4], data = df)
Residuals:
    Min      1Q  Median      3Q     Max
-47.625 -11.173  -2.807   6.038  91.446

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              37.2754     2.4232  15.382  < 2e-16 ***
BigMac_pca$x[, 1:4]PC1   10.5786     1.1181   9.461 8.94e-14 ***
BigMac_pca$x[, 1:4]PC2   -4.0159     2.1481  -1.870  0.06612 .
BigMac_pca$x[, 1:4]PC3    0.9618     2.7441   0.350  0.72712
BigMac_pca$x[, 1:4]PC4    8.2970     2.8349   2.927  0.00474 **
---------------------------------------------------------------
Residual standard error: 20.13 on 64 degrees of freedom
Multiple R-squared:  0.6137,    Adjusted R-squared:  0.5896
F-statistic: 25.42 on 4 and 64 DF,  p-value: 1.244e-12
```

```
> summary(lm(BigMac~., data = df))
Call:
lm(formula = BigMac ~ ., data = df)
Residuals:
    Min      1Q  Median      3Q     Max
-41.916 -10.053  -1.024   7.359  81.512

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 40.359743  16.884721   2.390   0.0200 *
Bread        0.387238   0.183319   2.112   0.0389 *
Rice         0.965387   0.182620   5.286  1.9e-06 ***
FoodIndex   -0.512792   0.194416  -2.638   0.0107 *
Bus         -0.229961   4.533740  -0.051   0.9597
Apt          0.003929   0.007795   0.504   0.6161
TeachGI      1.848863   1.363304   1.356   0.1802
TeachNI     -2.287929   1.830213  -1.250   0.2162
```

```
TaxRate      -0.775878   0.397161  -1.954   0.0555 .
TeachHours    0.295898   0.335860   0.881   0.3819
---------------------------------------------------------------------
Residual standard error: 18.73 on 59 degrees of freedom
Multiple R-squared:  0.6918,    Adjusted R-squared:  0.6448
F-statistic: 14.71 on 9 and 59 DF,  p-value: 3.744e-12
```

From the summaries we can see that the PCA model has an coefficient of determination of 0.6137 and the full model including all predictors has a coefficient of determination of 0.6918. Abiding by the principle of parsimony we want to explain the most that we can with the least amount of effort, PCA helps us achieve that, and in that sense this result is very satisfying.