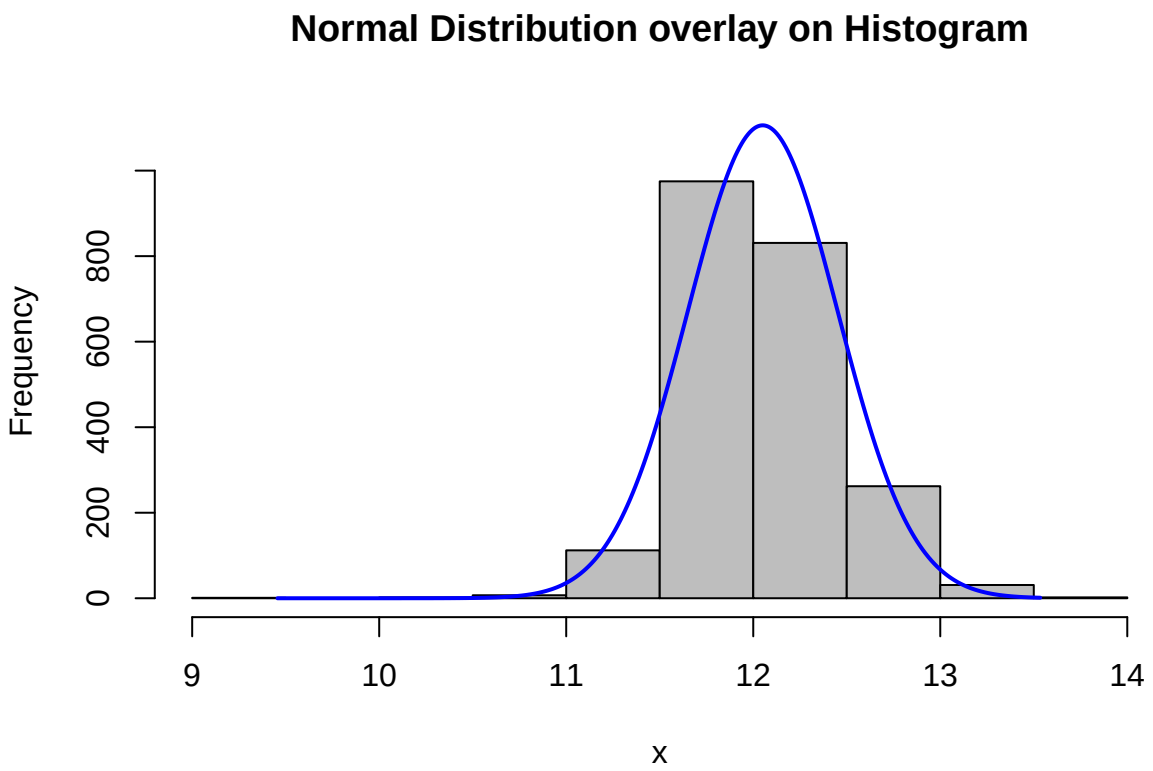
Coursework 2, Principle Components Analysis and House Price Predictions

2023-10-21

Question 1

Produce a histogram showing the empirical distribution of log prices. Overlay the plot with a Normaldensity with mean equal to the sample average of log prices and variance equal to the sample varianceof log prices. How can you use this plot to guide or validate the specification of the prediction modelfor price? (5 marks)

```
library(rcompanion)
library(glmnet)
library(boot)
hp <- read.csv("AmesHousePrices.csv", stringsAsFactors = T)
x <- hp[sapply(hp,is.numeric)]
drop <- c("price","MS.SubClass","Overall.Qual","Overall.Cond","BsmtFin.SF.2",
          "Bsmt.Full.Bath","Bsmt.Half.Bath",
          "Full.Bath", "Half.Bath", "Mo.Sold", "Yr.Sold")
x = x[,!(names(x) %in% drop)]
x[,2:ncol(x)] <- scale(x[,2:ncol(x)])

#1
plotNormalHistogram( log(hp$price), prob = FALSE,
                     main = "Normal Distribution overlay on Histogram",
                     )
```



This plot shows that the log of house prices are roughly normally distributed. Although this is not a totally accurate test, this is useful to check that data is roughly normally distributed, as some models require or work best with normally distributed residuals, such as OLS. Normally distributed values implies this may be the case. It also suggests that taking logs is an appropriate transformation to make, as it implies the errors may also be normally distributed.

Question 2

Perform Principal Components Analysis (PCA) on the numerical features in your data. Why focusonly on the numerical features? How many Principal Components (PC) do you find? Why do you findexactly that number of PCs? What is the variance of the first PC, i.e. PC1? What is the variance of thelast PC? What is the share of the total feature variance is captured by the first PC? Produce a screeplot of your PCA (for the plot, you may restrict attention to the first, say, 10 PCs). Briefly commenton the scree plot. (8 marks)

```
pcaprice = prcomp(x, scale = T)
(pcaprice$sdev[1])**2
```

```
## [1] 6.318309
```

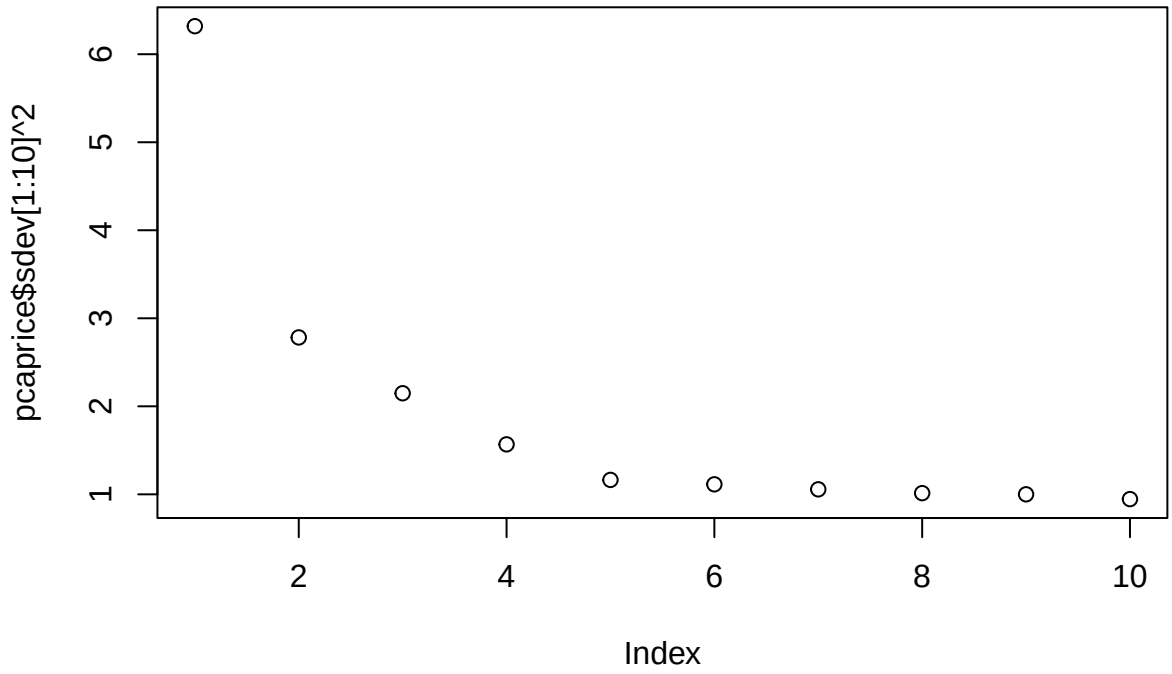
```
(pcaprice$sdev[26])**2
```

```
## [1] 2.573033e-30
```

```
total_sum <- 0
for (i in 1:length(pcaprice$sdev)) {
  total_sum <- (pcaprice$sdev[i])^2 + total_sum
  total_sum
}
pcaprice$sdev[1]**2/total_sum
```

```
## [1] 0.2430119
```

```r
plot(pcaprice$sdev[1:10]^2)
```



We only focus on the numerical features as PCA only works with numerical features. This is because PCA utilizes linear algebra, and requires continuous data to work. It should be noted that there are ways of adapting PCA for certain types of continuous data, though it has not been done here. e find 26 PCs, with the first having the highest variance of 6.32 (2d.p), and the last having the lowest variance of 1.13 e-30 (2d.p). We find 26 exactly, as PCA finds either n or p principal components, whichever is lower. in this case, there are fewer regressors than there are observtions, so we find the number of PCs equivalent to the number of regressors. PC1 makes up 24.3% of the total variance. The scree plot shows the variance of each of the first 10 PCs, and shows that it is decreasing at a slowing rate as PCi increases. This is because of how each PC is computed, where the rotational variables are picked to maximise the ammount of variance they can take into account. After theyre found, the values are removed and the process is repeated on the now lower variance residuals, resulting in lesser and lesser variance accounted for by the PCs.

Question 3

List the rotations for the first five PCs in your PCA. Provide a brief interpretation of each of the firstthree PCs based on the rotations. (12 marks)

Interpreting the roational values can give us informatiom about the relative importance of regressors in each principle component. A high theta value indicates that a country scores highly in PCi for high values of that variable; the value of each theta shows us the relative importance of each theta for PCi. Looking and the PC1, we see the highest absolute values are area, basement square footage, first floor squre footage, garage area and the storage capacity of the garage in terms of cars, with values of -0.31. The smallest absolute values are assigned to 3 season porch area and low quality finished square feet, with values of -0.01 and 0.01 respectively, followed by screen porch with a value of -0.03 and then misc value and pool area, with values of -0.04 and -0.05 respectively. These results suggest that PC1 is strongly influenced by factors influencing the size of the house, and as a result PC1 could represent a component highly influenced by area, and less so the quality of that area.

For PC2, second floor square footage, bedrooms and rooms above grade all have high absolute coefficients, of 0.4, 0,43 and 0.37 respectively. Lowest coefficients are masonry veneer area, open porch SF and misc val, with values of -0.01, 0.02 and 0.01 respectively. This suggests that PC2 could be influenced heavily by a house having a second floor or not, as open porches and masonry veneer areas are typically all found on the

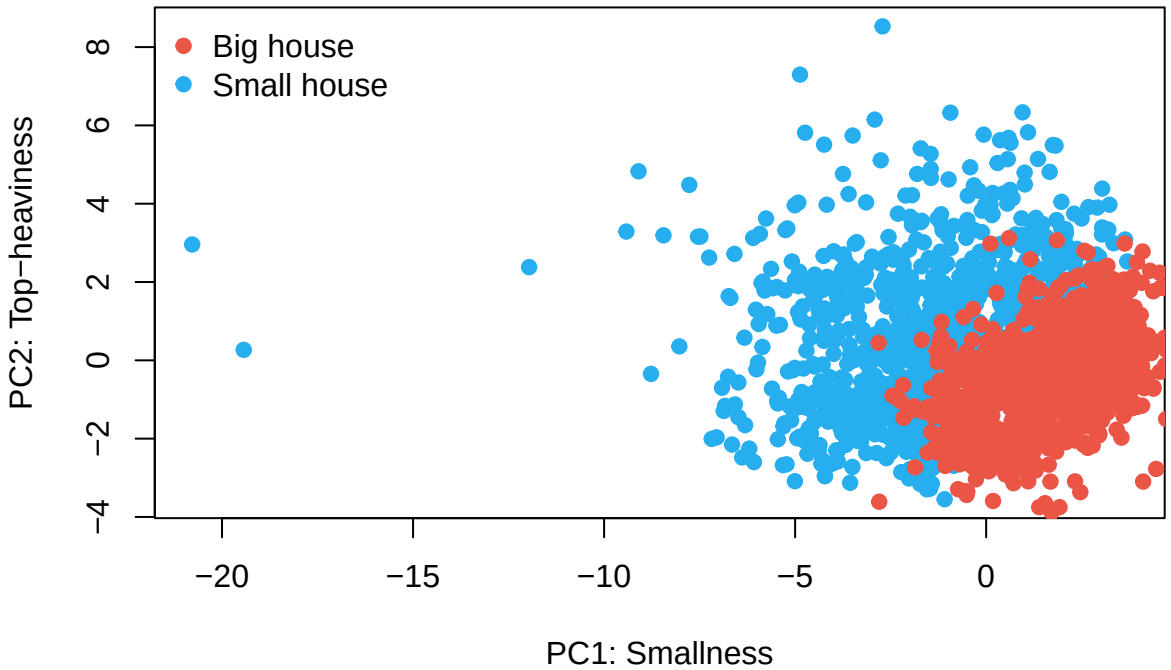
ground floor and have low values, while all high coefficient variables represent second floor factors. This would also account for why area has a relatively high coefficient of 0.29. Its high because second floor houses are likely to have a higher area, and thus is can account for some variance in this factor, but its not as high as the second floor values as its not directly a product of if the house has 2 floors.

For PC3, basement SF 1 has the highest absolute value of 0.39, with the next largest absolute values being garage year built, second floor square footage and year remodelled or rooms added, with values of -0.3, -0.32 and -0.29 respectively. PC3 may be representing some sort of contrast between the basement area and more recent additions to the house, with an emphasis on larger, maybe older basments vs newer parts of the house.

Question 4

Predict the PCs for each observation. Produce a scatter plot of the predicted PC1 against the predictedPC2 that supports (or questions, as the case may be) the interpretation you gave PC1 in question 3.Comment on the plot. (12 marks)

```
# Generate principal component scores from the PCA model
pc_results <- predict(pcaprice, newdata = x)

# Define color scheme for the plot
color_scheme = c("#ea5545", "#27aeef")

# Categorize houses based on area size
is_small_house <- (x$area < median(x$area))

# Create a scatter plot for big houses first, using the negation of the small house logical vector
plot(pc_results[!is_small_house, 1:2], col = color_scheme[2],
     xlab = "PC1: Smallness", ylab = "PC2: Top-heaviness", pch = 19)

# Overlay the small houses on the same plot
points(pc_results[is_small_house, 1:2], col = color_scheme[1], pch = 19)

# Add a legend to the plot for clarification
legend("topleft", bty = "n", pch = 19, col = color_scheme, legend = c("Big house", "Small house"))
```



The

scatter plot presents a distinct division according to house size. It appears that the houses with higher scores on the presumed 'smallness' factor are, in fact, smaller in size, while those with lower scores on this factor are typically larger. However, it's important to recognize that the first principal component (PC1) doesn't solely represent size—it's influenced by all the other 26 variables in the dataset. Despite this complexity, the visual evidence from the plot does seem to support the interpretation of PC1 as an indicator of 'smallness' in this instance, although it's worth noting that this conclusion is based on a somewhat limited set of data points.

Question 5

Fit a linear regression of log hp$price onto all the regressors in hp. Report in- and out-of-sampleM SE. Comparing in- and out-of-sample M SE, what do you observe? Does it surprise you? Brieflyexplain your answer. (8 marks)

```
x.mat <- as.data.frame(hp)
set.seed(1)

#split to training and test
test <- sample(1:nrow(x.mat), ceiling(nrow(x.mat) / 4))
training <- log(x.mat$price[-test])
testing <- log(x.mat$price[test])

#In sample
linmodel <- glm(log(price) ~ ., data = x.mat)
cv <- cv.glm(x.mat, linmodel, K= 10)

#Out of Sample
yhat_test <- predict(linmodel, newdata = x.mat)
residuals_test <- yhat_test - log(x.mat$price)
is_mse <- mean(residuals_test^2)
cv_mse <- cv$delta[1]
cv_mse
```

```
## [1] 0.02290376
```

```
is_mse
```

```
## [1] 0.01654336
```

Comparing OOS and IS MSE, we get values of 0.023 (2 d.p) and 0.017 (2 d.p) respectively. This doesn't suprise me, as we expect models to fit in sample data that they were trained on better than out of sample data.

Question 6

Fit a linear regression of log hp$price onto all the regressors in hp interacted with Neighborhood.Make a table where you peport in- and out-of-sample M SE for this regression and the ones from theregression fitted in 5. Compare in- and out-of-sample M SE of the two fitted models. What do youobserve? Does it surprise you? Briefly explain your answer. (8 marks)

```
suppressWarnings({
#train new glm model
linmodelN <- glm(log(price) ~ . * Neighborhood - Neighborhood, data = x.mat)

#CV for OOS MSE
cvN <- cv.glm(x.mat, linmodelN, K= 10)
yhat_testN <- predict(linmodelN, newdata = x.mat)
```

```
#IS MSE
y_testN <- x.mat$price
residuals_testN <- yhat_testN - log(x.mat$price)
is_mseN <- mean(residuals_testN^2)
oos_mseN <- cvN$delta[1]
})
is_mseN
```

```
## [1] 0.01516709
```

```
oos_mseN
```

```
## [1] 0.02287231
```

```
table_df <- data.frame(matrix(NA, nrow = 2, ncol = 2))

# Set column names
colnames(table_df) <- c('Full', 'Interacted' )

# Set row names
rownames(table_df) <- c('IS MSE', 'OOS MSE')
table_df['IS MSE', 'Full'] <- is_mse
table_df['OOS MSE', 'Full'] <- cv_mse
table_df['IS MSE', 'Interacted'] <- is_mseN
table_df['OOS MSE', 'Interacted'] <- oos_mseN
# Display the table
table_df
```

```
##                Full Interacted
## IS MSE  0.01654336 0.01516709
## OOS MSE 0.02290376 0.02287231
```

This table shows that the when interacted with neighborhood terms, the MSE for the out of sample fit stays virtually the same, while the in - sample MSE gets a little lower, suggesting to an extent that it potentially helps explain some of the data, or at the very least doesnt throw off the fit of the regression. It is unsuprising that the inclusion of additional variables increases the in sample fit, as it encourages overfitting. However the similar OOS scores also suggest that the inclusion of the interaction terms does not contribute to the model being less accurate out of sample.

Question 7

Fit a linear regression of log hp$price onto all the regressors in hp interacted with Neighborhood usingthe lasso algorithm. Report out-of-sample M SE at the out-of-sample M SE-minimizing penalty weight.Compare this out-of-sample M SE to those you reported in 5 and 6. What do you observe? Does itsurprise you? Briefly explain your answer. (8 marks)

```
drop <- c("price")
feat = hp[,!(names(hp) %in% drop)]
x.df <-  model.matrix( ~ Neighborhood*., data=feat)
y <- log(hp$price)
lasso <- cv.glmnet(x.df, y, alpha = 1,
                   )


optimal_lambda <- lasso$lambda.min
oos_mseL <- lasso$cvm[lasso$lambda == optimal_lambda]
```

```
predicted_values <- predict(lasso, newx = x.df, s = optimal_lambda, type = "response")
residuals_testN <- predicted_values - log(x.mat$price)
is_mseL <- mean(residuals_testN^2)

table_df <- data.frame(matrix(NA, nrow = 2, ncol = 3))

# Set column names
colnames(table_df) <- c('Full', 'Interacted', 'Lasso' )

# Set row names
rownames(table_df) <- c('IS MSE', 'OOS MSE')
table_df['IS MSE', 'Full'] <- is_mse
table_df['OOS MSE', 'Full'] <- cv_mse
table_df['IS MSE', 'Interacted'] <- is_mseN
table_df['OOS MSE', 'Interacted'] <- oos_mseN
table_df['IS MSE', 'Lasso'] <- is_mseL
table_df['OOS MSE', 'Lasso'] <- oos_mseL
table_df
```

```
##               Full Interacted      Lasso
## IS MSE  0.01654336 0.01516709 0.01609427
## OOS MSE 0.02290376 0.02287231 0.02188372
```

```
is_mseL
```

```
## [1] 0.01609427
```

```
oos_mseL
```

```
## [1] 0.02188372
```

again we observe a similar pattern, where both in and out of sample MSE remain very similar across all models. The decrease in MSE from both full and interacted to lasso is unsurprising, as lasso varies penalty weight in order to minimize OOS MSE while simplifying the model.

Question 8

The command hpPC <- data.frame(hp,as.data.frame(Z[,1:ncol(Z)-1])) omits the last PrincipalComponent. Why do omit that last component in our prediction model? (3 marks)

We can safely exclude the last PC, as it is likely to contain only very small values and be of little significance. This is backed up by the incredibly low ammount of variance PC26 explains (1.13 e-30 (2d.p)), shown in Q2. The removal of the last PC helps allows us to actually reduce dimensions from the full model and see how the reduced model in its fullest form compares, as well as reducing computational cost.

Question 9

Fit a linear regression of log hp$price onto the PC-features in hpPC by OLS. Do not include any otherfeatures, nor should you include any interaction terms. Report in- and out-of-sample M SE. Make atable where you report in- and out-of-sample M SE for this regression and the ones from the regressionsfitted in 5 and in 6. Compare in- and out-of-sample M SE for the PCR with those of the fitted modelsfrom 5 and 6. What do you observe? Does it surprise you? Briefly explain your answer. (8 marks)

```
hpPC <- data.frame(hp,as.data.frame(pc_results[,1:ncol(pc_results)-1]))
regressors <- tail(names(hpPC), 25)

formula_str <- paste("log(price) ~", paste(regressors, collapse = " + "))
```

```
PCR <- glm(as.formula(formula_str), data = hpPC)
PCRcv <- cv.glm(hpPC, PCR, K = 10)
predicted_values <- predict(PCR, newdata = hpPC)
residuals_testP <- log(x.mat$price) - predicted_values
is_mseP <- mean(residuals_testP^2)

table_df <- data.frame(matrix(NA, nrow = 2, ncol = 3))

# Set column names
colnames(table_df) <- c('Full', 'Interacted', 'PCR' )

# Set row names
rownames(table_df) <- c('IS MSE', 'OOS MSE')
table_df['IS MSE', 'Full'] <- is_mse
table_df['OOS MSE', 'Full'] <- cv_mse
table_df['IS MSE', 'Interacted'] <- is_mseN
table_df['OOS MSE', 'Interacted'] <- oos_mseN
table_df['IS MSE', 'PCR'] <- is_mseP
table_df['OOS MSE', 'PCR'] <- PCRcv$delta[1]
table_df
```

```
##                Full Interacted        PCR
## IS MSE  0.01654336 0.01516709 0.02889369
## OOS MSE 0.02290376 0.02287231 0.03510455
```

```
is_mseP
```

```
## [1] 0.02889369
```

```
PCRcv$delta[1]
```

```
## [1] 0.03510455
```

The results of this regression show an increase in both in and out of sample MSE, suggesting a less appropriate model has been trained. This is suprising, as a full PC model should explain all of the variance explained in a full model, and so I would expect the out of sample fit to be quite similar.

10

Fit a linear regression of log hp$price onto the PC-features in Z as well as the regressors in hp interactedwith Neighborhood using the lasso algorithm. Make a table where you report out-of-sample M SE atthe out-of-sample M SE-minimizing penalty weight for this lasso and the lasso fitted in 7. What doyou observe? Does it surprise you? Briefly explain your answer. (8 marks)

```
xmat2 <- cbind(x.df,pc_results[,1:ncol(pc_results)-1])
y <- log(hpPC$price)
pclasso <- cv.glmnet(xmat2, y, alpha = 1, )
optimal_pclambda <- which(pclasso$lambda == pclasso$lambda.min)
optimal_pcmse <- pclasso$cvm[pclasso$lambda == optimal_pclambda]
predicted_values <- predict(pclasso, newx = xmat2, s = optimal_lambda, type = "response")
residuals_testN <- predicted_values - log(hpPC$price)
is_mseL <- mean(residuals_testN^2)

table_df <- data.frame(matrix(NA, nrow = 2, ncol = 2))

# Set column names
```

```r
colnames(table_df) <- c('Lasso PC', 'PCR'  )

# Set row names
rownames(table_df) <- c('IS MSE', 'OOS MSE')
table_df['IS MSE', 'Lasso PC'] <- is_mseL
table_df['OOS MSE', 'Lasso PC'] <- pclasso$cvm[optimal_pclambda]
table_df['IS MSE', 'PCR'] <- is_mseP
table_df['OOS MSE', 'PCR'] <- PCRcv$delta[1]
table_df
```

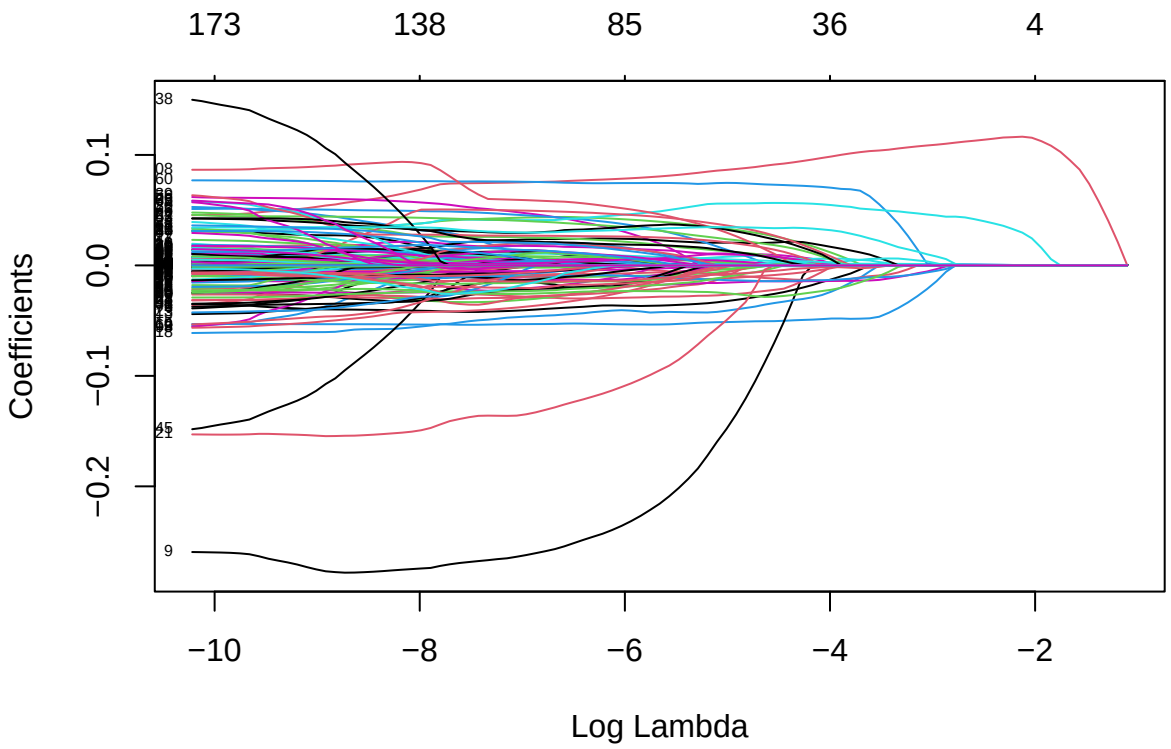
```
##           Lasso PC        PCR
## IS MSE  0.01612825 0.02889369
## OOS MSE 0.02123494 0.03510455
```

When this regression has been run, we find that for the lasso PC we just ran, both the IS and OOS MSE is lower. This suggests that this model better explains the variance in y than the PCRs alone, and that the combination of the two gives us the best model. I would attribute the success to lasso's ability to select the most usefull regressors, and zero out the rest. It has selected the most useful PCs and regressors. I would however be wary of colinearity, as both the PCs and original regressors are explainng the same vairance in Y.

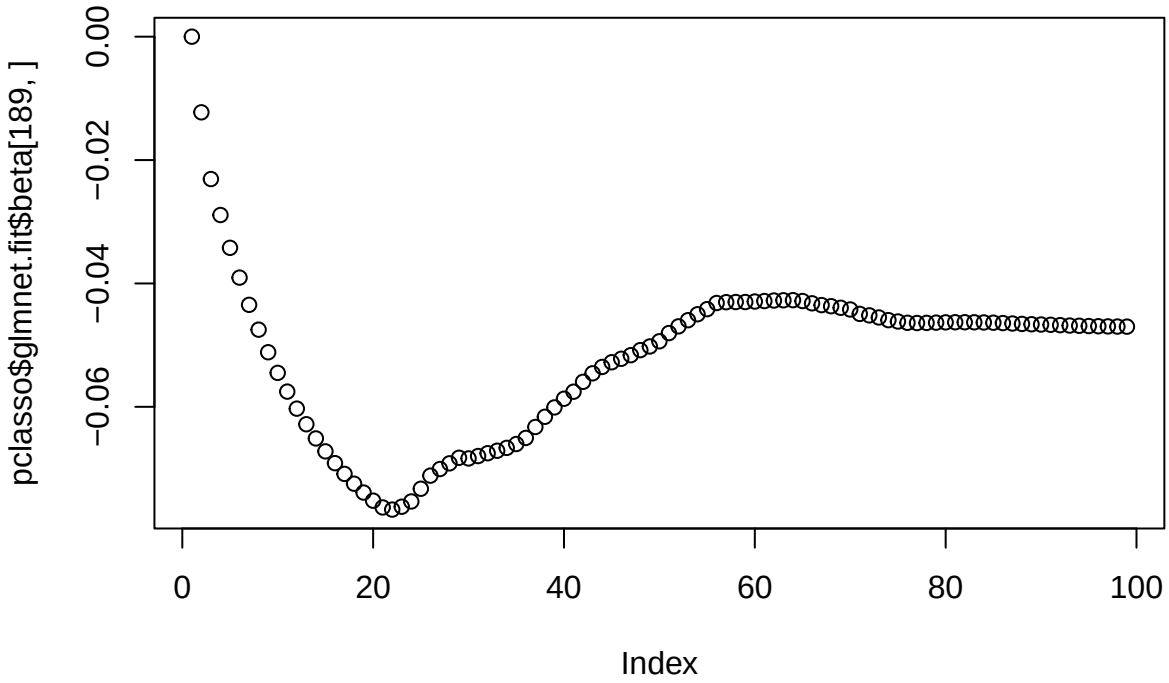
Question 11

Plot the coefficients along the regularization path (i.e. as a function of the log penalty weight log lambda) forthe PCR-lasso from 10. Highlight the coefficient path for the PC1-feature in the PCR-lasso. What doyou observe? Does it surprise you? Briefly explain your answer. (10 marks)

```r
plot(lasso$glmnet.fit, xvar = "lambda",label = T)
```




```r
plot(pclasso$glmnet.fit$beta[189,])
```



This graph shows us the regularisation path of PC1, extracted from all regressors. It shows that the value of PC1 shallowly decreaseses, then more rapidly increases followed by a sharp decrease to 0 as lambda increases. This indicates that it is a significant variable, as we would expect from PC1, as while the 'coefficient budget' shrinks, the ammount of value assigned to it increases. We would expect PC1 to have a high ammount of importance, and budget spent on it as it should capture the most variance out of any PC.

Question 12

Questions 9 and 10 both included the hint "You do not need to include the PCA in the cross validationto get the out-of-sample M SE." This is cheating! A true assessment of the OOS predictions wouldinclude the PCA in the cross validation. Briefly outline how such a procedure could be carried out. (10marks)

To correctly perform PCR, we need to make sure we conduct PCA as a part of each training step, or K fold in cross validation to make sure that the PCA is only fitted on the set of training data in each fold, to make it as accurate as possible. The new algorithm would work the same as a standard K fold algorithm, but would have the additional step of performing PCA on training data from each fold before cross validating, and then fitting the model on the training data using those new principle components. The average error would then be calculated as standard in K fold CV.