DATA 303 Assignment 3

Izzy Southon, 300597453

Due: 11:59 PM Friday 17 May 2024

Assignment Questions

1. **(8 marks)**

Data were collected on 158 cruise ships in operation around the world in 2013. Complaints had been raised by customers about overcrowding on cruises and there was interest in investigating whether there was a trend of overcrowding on certain types of ships. As part of the investigation, a regression analysis was carried out that could be used to predict passenger density (number of passengers per unit area) based on ship characteristics. The data are available in the dataset `cruise_ship.csv` and include the following variables:

```
Variable | Description
--- | ------------------
‘age.2013‘ | Age (as of 2013)
‘tonnage‘ | Weight of ship (1000s of tonnes)
‘passengers.100‘ | Maximum number of passengers (100s)
‘length‘ | Length of ship (100s of feet)
‘cabins‘ | No. of passenger cabins (100s)
‘crew.100‘ | No. of crew member (100s)
‘pass.density‘ | Passenger density (no. of passengers per square foot)
```

There were high correlations among some pairs of predictors. As a result, principal components regression was used to avoid potential multi-collinearity issues.

```
cs<-read.csv("cruise_ship.csv")
str(cs)
```

```
## 'data.frame':    158 obs. of  10 variables:
##  $ name          : chr  "Journey" "Quest" "Celebration" "Conquest" ...
##  $ line          : chr  "Azamara" "Azamara" "Carnival" "Carnival" ...
##  $ line_grp      : chr  "Other" "Other" "Carnival" "Carnival" ...
##  $ age.2013      : int  6 6 26 11 17 22 15 23 19 6 ...
##  $ tonnage       : num  30.3 30.3 47.3 110 101.4 ...
##  $ passengers.100: num  6.94 6.94 14.86 29.74 26.42 ...
##  $ length        : num  5.94 5.94 7.22 9.53 8.92 8.55 8.55 8.55 8.55 9.51 ...
##  $ cabins        : num  3.55 3.55 7.43 14.88 13.21 ...
##  $ pass.density  : num  42.6 42.6 31.8 37 38.4 ...
##  $ crew.100      : num  3.55 3.55 6.7 19.1 10 9.2 9.2 9.2 9.2 11.5 ...
```

a. **(3 marks)** Obtain the principal components (PCs) for the 6 predictors in the dataset using the `pcr` command and provide summary output. Based on the output:

```
library(pls)
library(pander)
library(MASS)
library(leaps)
```

```
set.seed(1)
pcr.mod<-pcr(pass.density ~ age.2013 + tonnage + passengers.100 + length +
               cabins + crew.100, data = cs, scale = TRUE, validation="CV")
summary(pcr.mod)
```

```
## Data:    X dimension: 158 6
##  Y dimension: 158 1
## Fit method: svdpc
## Number of components considered: 6
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           8.667    8.663    7.676    7.148    7.098    5.246    5.360
## adjCV        8.667    8.656    7.663    7.134    7.084    5.230    5.335
##
## TRAINING: % variance explained
##                1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X               84.347    95.21    97.57    99.02    99.73   100.00
## pass.density     1.535    25.56    36.19    37.51    66.20    67.09
```

```
lm.mod<-lm(pass.density ~ age.2013 + tonnage + passengers.100 + length +
               cabins + crew.100, data=cs)
summary(lm.mod)$r.squared
```

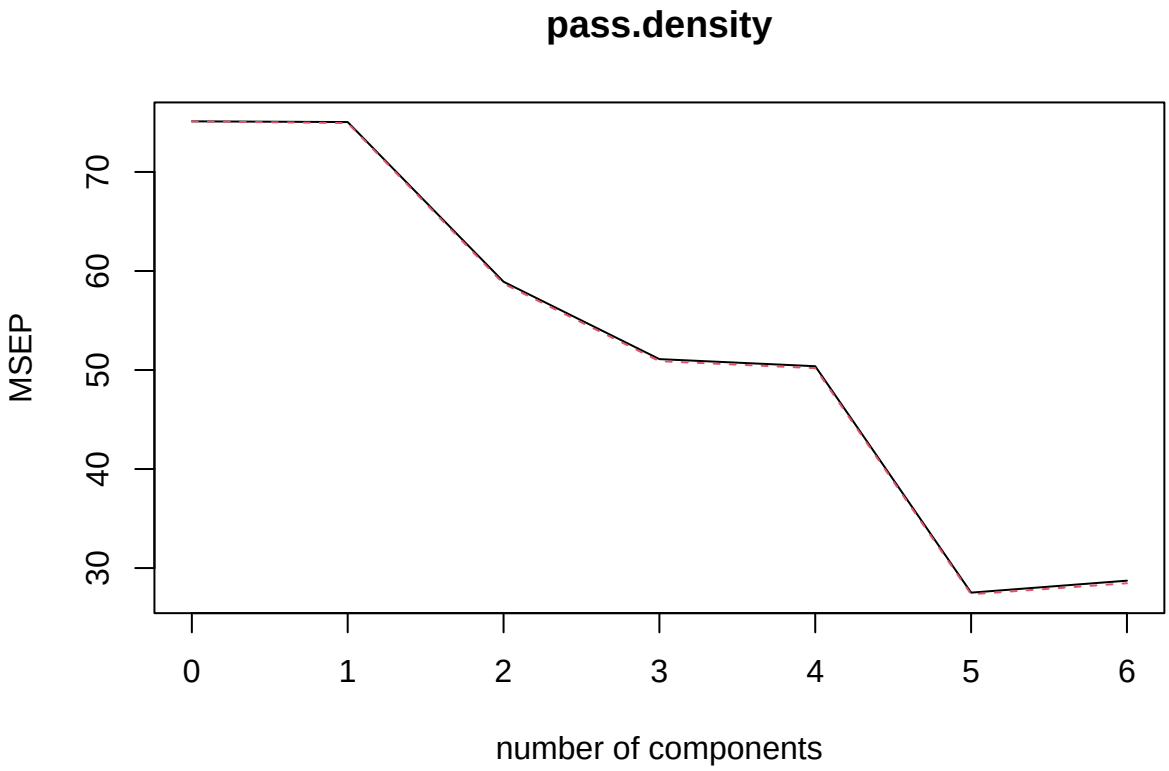
```
## [1] 0.6708501
```

i. How many principal components are required to explain at least 90% of the variance in the predictors?
ii. What would be the R^2 value for a linear model for `pass.density` that includes all predictors. Explain your answer based on the summary output of the `pcr` command.

More than two principal components are required to explain at least 90% of the variance in the predictors (the predictors being the numerical predictors age.2013, tonnage, passengers.100, length, cabins, and crew.100). The R^2 value for a linear model for `pass.density` that includes all predictors is 0.6709. All 6 PCs explain 67.09% of the variation in passenger density. This is the same as the value of R2 in the least squares regression model that includes all predictors.

b. **(3 marks)** Obtain a plot of the cross-validation mean squared error (MSE) for each number of principal components and state the number of PCs that gives the smallest cross-validation MSE.

```
validationplot(pcr.mod, val.type = "MSEP")
```




```
min.pcr = which.min(MSEP(pcr.mod)$val[1,1, ] ) - 1
min.pcr
```

```
## 5 comps
##       5
```

The plot shows that the smallest cross-validation MSE occurs for 5 PCs.

c. **(2 marks)** Based on your results from parts (a) and (b), how many PCs would you choose to represent the predictors in a regression model in place of all the predictors. Explain your answer briefly.

There is no widely accepted method for selecting the number of PCs. In part a, 2 PCs explained at least 90% of variation in the predictors, meaning that the 2 PCs captured at least 90% of the information in the 6 predictors. 1 PC explained at least 80% of variation in the predictors, meaning 1 PC captured at least 80% of the information in the 6 predictors. In part b the CV error shows a roughly downward trend. This suggests that using more than 5 PCs does not significantly increase the predictive performance of the model. 90% is an acceptably high amount of variation to capture and therefore the dimension of the predictor space can be reduced from 5 to 2 and still capture most the the variability in the space. We opt to use 2 PCs to represent the predictors in a regression model.

2. **(12 marks)**

In a 1978 study on absenteesim from school, data on 146 children from Walgett, New South Wales, Australia were collected. The number of days absent from school in a particular school year was recorded for each child, together with some demographic information. The data are available in the dataset `quine.csv` and include the following variables:

```
Variable | Description
--- | ------------------
‘Eth |‘ Ethnic background | Aboriginal or Not, ("A" or "N").
‘Sex |‘ Factor with levels ("F" or "M")
‘Age |‘ Primary ("F0"), or forms "F1," "F2" or "F3".
‘Lrn |‘ "AL" = Average learner, "LD"=Learning disabilities
‘Days |‘ Days absent from school in the year.
```

```
quine<-read.csv("quine.csv")
str(quine)
```

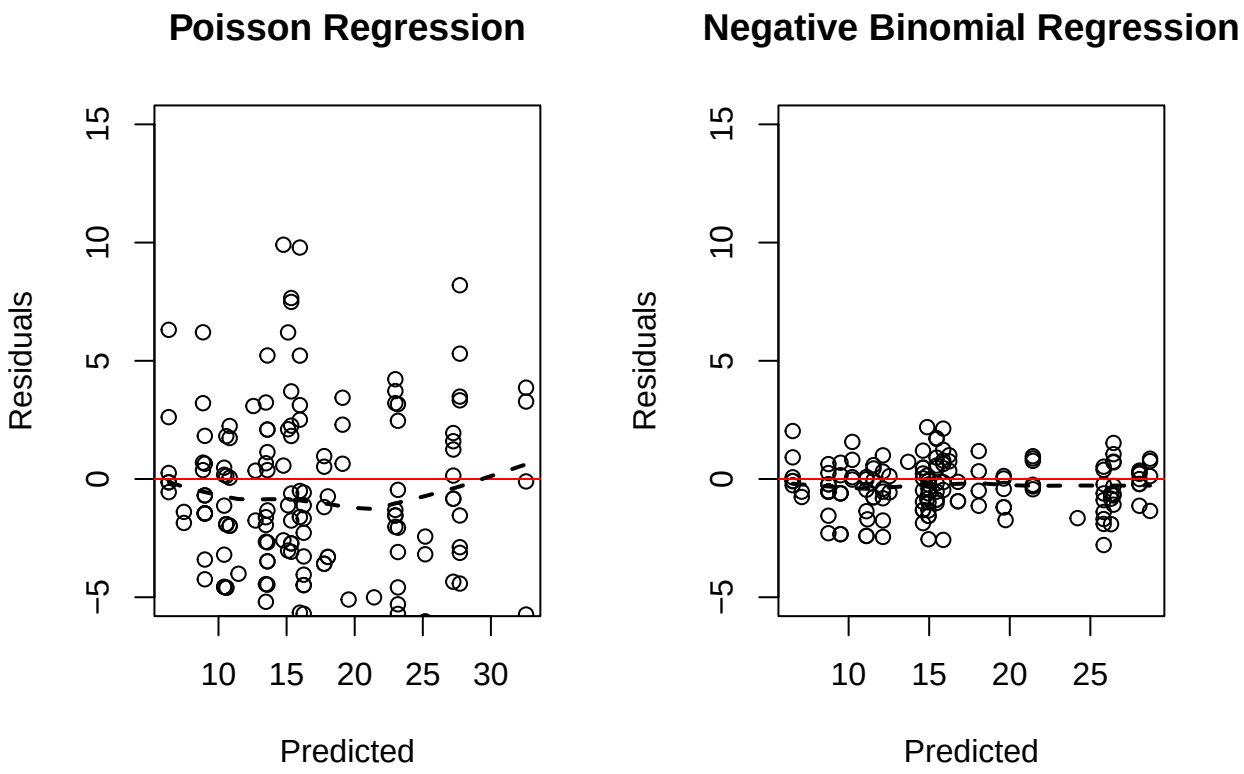
```
## 'data.frame':    146 obs. of  5 variables:
##  $ Eth : chr  "A" "A" "A" "A" ...
##  $ Sex : chr  "M" "M" "M" "M" ...
##  $ Age : chr  "F0" "F0" "F0" "F0" ...
##  $ Lrn : chr  "LD" "LD" "LD" "AL" ...
##  $ Days: int  2 11 14 5 5 13 20 22 6 6 ...
```

a. **(2 marks)** Explain briefly why it is unnecessary to use an offset variable in a model for the number of days absent.

It is unnecessary to use an offset variable (a variable with a coefficient constrained to be 1) because all children had equal exposure. The number of days absent from school were counted over a period of the school year for all children in the study.

b. **(3 marks)** Fit a Poisson and a negative binomial regression model with `Days` as the response variable and the rest of the variables as predictors. Obtain plots of residuals against predicted values. Comment on what the plots show.

```
# Fitting models
# Poisson
quine.pois<-glm(Days ~ as.factor(Eth) + as.factor(Sex) + as.factor(Age) + as.factor(Lrn),
data = quine, family = poisson)

# Negative binomial regression
quine.nb<-glm.nb(Days ~ as.factor(Eth) + as.factor(Sex) + as.factor(Age) + as.factor(Lrn),
data = quine)

# Plots of residuals against predicted values
par(mfrow=c(1,2))
# Poisson
plot(predict(quine.pois, type = "response"), residuals(quine.pois), main="Poisson Regression",
ylab="Residuals", xlab="Predicted", ylim=c(-5,15))
abline(h=0,lty=1,col="red")
lines(lowess(predict(quine.pois,type="response"),residuals(quine.pois)),lwd=2, lty=2)

# Negative binomial
plot(predict(quine.nb,type="response"),residuals(quine.nb), main="Negative Binomial Regression",
ylab="Residuals", xlab="Predicted", ylim=c(-5,15))
abline(h=0,lty=1,col="red")
lines(lowess(predict(quine.nb,type="response"),residuals(quine.nb)), lwd=2, lty=2)
```



In the poisson model, the dotted line seems to show no trend in the residual vs predicted values relationship, however, the dotted line increases past the zero line which might suggest that as the predicted values increase, the model may under-predict as the number of days absent gets larger. For the negative binomial model, the dotted line shows no trend in the residual vs predicted values relationship.

c. **(2 marks)** Calculate AIC and BIC statistics for both models in part (b) and print these in a table. State the preferred model based on these results.

```
ICs<-data.frame(c("Poisson", "NB"),
c(AIC(quine.pois), AIC(quine.nb)),
c(BIC(quine.pois), BIC(quine.nb)))
colnames(ICs)<-c("Model", "AIC", "BIC")
library(pander)
pander(ICs)
```

Model	AIC	BIC
Poisson	2299	2320
NB	1109	1133

Based on these results, AIC and BIC both indicate preference for the negative binomial model as a lower AIC and BIC are obtained for the NB model than for the poisson model.

d. **(2 marks)** Give the value of $\hat{\theta}$ from the fitted negative binomial model.

```
summary(quine.nb)
```

```
##
## Call:
## glm.nb(formula = Days ~ as.factor(Eth) + as.factor(Sex) + as.factor(Age) +
##     as.factor(Lrn), data = quine, init.theta = 1.274892646, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7918  -0.8892  -0.2778   0.3797   2.1949
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       2.89458    0.22842  12.672  < 2e-16 ***
## as.factor(Eth)N  -0.56937    0.15333  -3.713 0.000205 ***
## as.factor(Sex)M   0.08232    0.15992   0.515 0.606710
## as.factor(Age)F1 -0.44843    0.23975  -1.870 0.061425 .
## as.factor(Age)F2  0.08808    0.23619   0.373 0.709211
## as.factor(Age)F3  0.35690    0.24832   1.437 0.150651
## as.factor(Lrn)LD  0.29211    0.18647   1.566 0.117236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.2749) family taken to be 1)
##
##     Null deviance: 195.29  on 145  degrees of freedom
## Residual deviance: 167.95  on 139  degrees of freedom
## AIC: 1109.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.275
##           Std. Err.:  0.161
##
##  2 x log-likelihood:  -1093.151
```

From the summary output above, we find $\hat{\theta} = 1.275$

e. **(3 marks)** Use the formula for $Var(Y)$ for the negative binomial distribution and the value of $\hat{\theta}$ in part (d) to explain why your conclusion in part (c) is not surprising.

```
mu <- mean(quine$Days)
theta <- 1.275

variance <- mu + (mu^2) / theta
variance
```

```
## [1] 228.926
```

The conclusion in part (c) is not surprising because $\hat{\theta} = 1.275$ and $Var(Y) = 228.926$, meaning that count variables do not meet the Mean=Variance assumption. Since the variance of 228.926 is much greater than

the mean of the response variable "Days", there is over-dispersion. Therefore, given the over-dispersion in the data (as indicated by the variance and theta), it's not surprising that the negative binomial model is preferred over the Poisson model because the negative binomial model incooperates over-dispersion by estimating the amount of extra variation as

$Var(Y) = \mu + \frac{\mu^2}{\theta} = E(Y) + \frac{(E(Y))^2}{\theta}$, where θ is an over-dispersion parameter.

3. **(20 marks)**

The majority of modern high rise structures are dependent on concrete for structural integrity and durability. High-performance concrete (HPC) is made using a mix of ingredients, and it is of interest to predict the performance of HPC from the composition of ingredients. We will make use of data contained in the dataset `concrete.csv`. This dataset contains experimental data on the use of seven different ingredients in HPC:

```
* cement (‘CEMENT‘)
* slag (‘SLAG‘)
* fly ash (‘FLY_ASH‘)
* water (‘WATER‘)
* superplasticizer (‘SP‘)
* coarse aggregate (‘COARSE_AG‘)
* fine aggregate (‘FINE_AG‘)

All of these ingredients are measured in kg/m$^3$.  Three different outcomes were measured:

* slump (‘SLUMP‘)
* flow (‘FLOW‘)
* 28-day compressive strength (‘COMP_STRENGTH‘)
```

```
con<-read.csv("concrete.csv")
str(con)
```

```
## 'data.frame':    103 obs. of  11 variables:
##  $ ID           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ CEMENT       : num  273 163 162 162 154 147 152 145 152 304 ...
##  $ SLAG         : num  82 149 148 148 112 89 139 0 0 0 ...
##  $ FLY_ASH      : num  105 191 191 190 144 115 178 227 237 140 ...
##  $ WATER        : num  210 180 179 179 220 202 168 240 204 214 ...
##  $ SP           : num  9 12 16 19 10 9 18 6 6 6 ...
##  $ COARSE_AG    : num  904 843 840 838 923 860 944 750 785 895 ...
##  $ FINE_AG      : num  680 746 743 741 658 829 695 853 892 722 ...
##  $ SLUMP        : num  23 0 1 3 20 23 0 14.5 15.5 19 ...
##  $ FLOW         : num  62 20 20 21.5 64 55 20 58.5 51 51 ...
##  $ COMP_STRENGTH: num  35 41.1 41.8 42.1 26.8 ...
```

a. **(1 mark)** It is of interest to estimate the coefficient of variation for the slump-to-flow ratio $\left(\frac{\texttt{SLUMP}}{\texttt{FLOW}}\right)$. Note that the coefficient of variation is given by

$$\text{CV} = \frac{\sigma}{\mu},$$

which is estimated by

$$\widehat{\text{CV}} = \frac{\overline{X}}{S^2}.$$

Estimate the coefficient of variation for the slump-to-flow ratio from the original data.

```r
# Calculate SLUMP/FLOW ratio
con$SLUMP_FLOW <- con$SLUMP / con$FLOW

# Compute mean and standard deviation of SLUMP/FLOW ratio
mean_slump_flow <- mean(con$SLUMP_FLOW)
sd_slump_flow <- sd(con$SLUMP_FLOW)

# Calculate coefficient of variation
CV_slump_flow <- (sd_slump_flow / mean_slump_flow)
CV_slump_flow_perc <- (sd_slump_flow / mean_slump_flow) * 100

# Print the coefficient of variation
print(CV_slump_flow)
```

```
## [1] 0.4498718
```

```r
print(CV_slump_flow_perc)
```

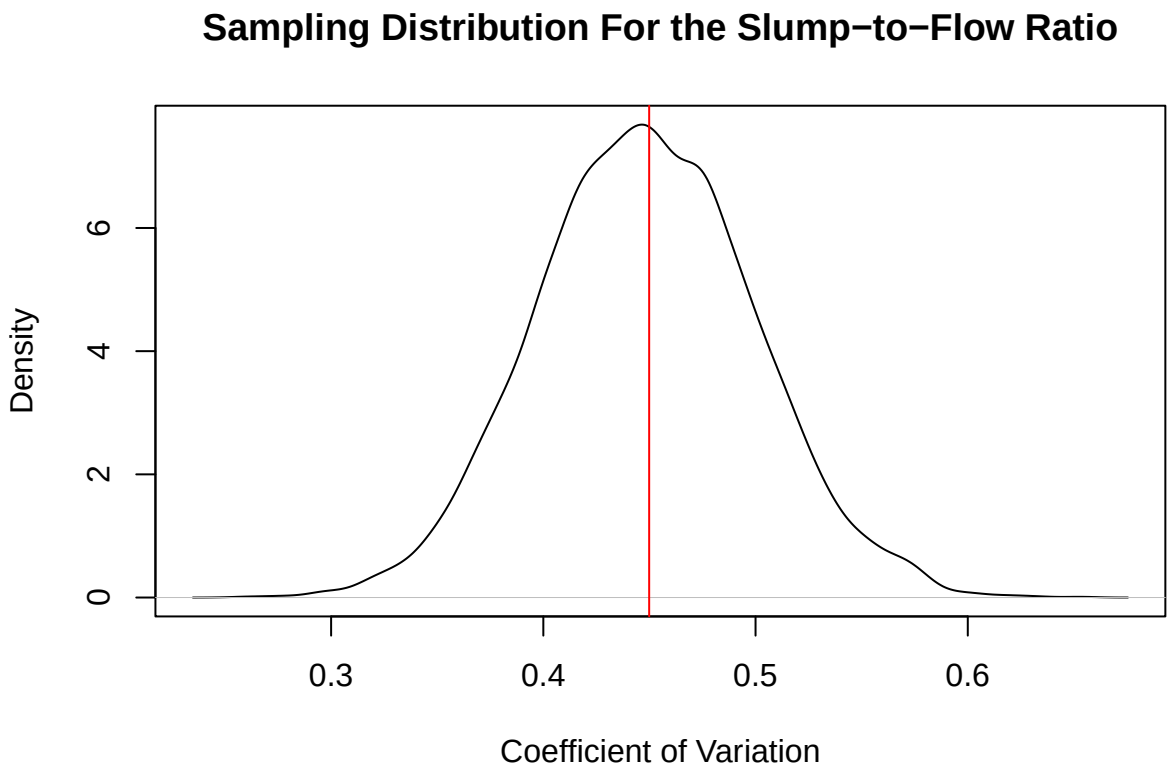
```
## [1] 44.98718
```

From the original data, the estimate for the coefficient of variation for the slump-to-flow ratio is 0.4499 (4dp) or 44.99%.

b. **(3 marks)** Now use 10,000 bootstrap samples to simulate the sampling distribution for the coefficient of variation for the slump-to-flow ratio. Present a density plot of the sampling distribution and a vertical bar at the estimated coefficient of variation from the original data. Describe the shape of the sampling distribution.

```r
set.seed(0)

nboot<-10000

bootstrap_CV <- numeric(nboot)

# Perform bootstrap resampling
for (i in 1:nboot) {

  # Sample with replacement from SLUMP/FLOW ratio
  bootstrap_sample <- sample(con$SLUMP_FLOW, replace = TRUE)

  # Calculate mean and standard deviation of the bootstrap sample
  mean_bootstrap <- mean(bootstrap_sample)
  sd_bootstrap <- sd(bootstrap_sample)

  # Calculate coefficient of variation for the bootstrap sample
  bootstrap_CV[i] <- sd_bootstrap / mean_bootstrap
}

# Plot the density of bootstrap CV values
density_plot <- density(bootstrap_CV)
```

```r
# Create a new plot
plot.new()

# Plot the density
plot(density_plot, main = "Sampling Distribution For the Slump-to-Flow Ratio",
     xlab = "Coefficient of Variation", ylab = "Density")

# Add a vertical line for the estimated CV from the original data
abline(v = CV_slump_flow, col = "red")
```



The density plot above shows that the shape of the sampling distribution seems to follow a normal distribution. This indicates that the variability of the slump to flow ration (SLUMP/FLOW) across the bootstrap samples is symmetric around the mean.

We now turn our focus to predicting `COMP_STRENGTH`, measured in millions of pascals (MPa), using a linear regression of `COMP_STRENGTH` on the seven ingredients: `CEMENT`, `SLAG`, `FLY_ASH`, `WATER`, `SP`, `COARSE_AG`, and `FINE_AG`.

c. **(4 marks)** Carry out an exhaustive model search using best subset selection of the seven predictors `CEMENT`, `SLAG`, `FLY_ASH`, `WATER`, `SP`, `COARSE_AG`, and `FINE_AG`. Use the `regsubsets` function in the `leaps` package, explaining which predictors would be included in the best model selected using BIC. How would the set of predictors selected change if instead using adjusted R^2 or the C_p statistic?

The exhaustive model search using 7 predictors has $2^7 = 128$ possible models, but exclude the null model, leaving 127 models.

```
exhaustive_model<-regsubsets(COMP_STRENGTH ~ CEMENT + SLAG + FLY_ASH + WATER + SP + COARSE_AG + FINE_AG
exhaustive_model_summary<-summary(exhaustive_model)
pander(exhaustive_model_summary$outmat)
```

	CEMENT	SLAG	FLY_ASH	WATER	SP	COARSE_AG	FINE_AG
1 (1)	*						
2 (1)	*		*				
3 (1)	*		*	*			
4 (1)	*		*	*		*	
5 (1)	*		*	*		*	*
6 (1)	*	*	*	*		*	*
7 (1)	*	*	*	*	*	*	*

```
c(best_adjrsq=which.max(exhaustive_model_summary$adjr2),
  best_cp=which.min(exhaustive_model_summary$cp),
  best_bic=which.min(exhaustive_model_summary$bic))
```

```
## best_adjrsq     best_cp    best_bic
##           6           6           5
```

The best model according to BIC is model 5. Model 5 includes the predictors `CEMENT`,`FLY_ASH`,`WATER`,`COARSE_AG`, and `FINE_AG`. Both adjusted R^2 and the C_p statistic found model 6 to be the best model, adding `SLAG` to the set of predictors.

d. **(4 marks)** Now carry out best subset selection using 20 repetitions of 10-fold cross-validation and the criterion of test MSE. Focusing on the top 10 models in terms of test MSE, which predictors would be included in the best model selected using this approach? Why?

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:pls':
##
##     R2
```

```
library(doParallel) # allows you to do parallel computing
```

```
## Loading required package: foreach

## Loading required package: iterators

## Loading required package: parallel
```

```r
library(foreach)
```

```r
# Set random number generator seed for replicability of results.
set.seed(0)

# Specify the indices of predictors to be considered.
variable.indices <- 2 : 8 # columms of the variables to be used as predictors

# Produce a matrix
all.comb <- expand.grid(as.data.frame(matrix(rep(0 : 1, length(variable.indices)), nrow = 2)))[-1, ] #

# Fire up 75% of computer cores for parallel processing.
nclust <- makeCluster(detectCores() * 0.75)
registerDoParallel(nclust)

##############################################
## MSE: Repeated 10-fold cross-validation. ##
##############################################

# Specify the number of repetitions of k-fold cross-validation.
folds <- 10
reps <- 20

fitControl <- trainControl(method = "repeatedcv", number = folds, repeats = reps, seeds = 1 : (folds * r

model.fits <- foreach(i = 1 : nrow(all.comb), .packages = "caret") %dopar%
{
  model.equation <- as.formula(paste("COMP_STRENGTH ~", paste(names(con)[variable.indices][all.comb[i,]
  train(model.equation, data = con, method = "lm", trControl = fitControl)
}

MSE.extract <- function(x)
{
  return(as.numeric(x$results[2]) ^ 2)
}

# Apply the function to all of the candidate models that were fit using
# repeated 10-fold cross-validation.
MSE.rep.cv <- sapply(model.fits, MSE.extract)

# View the 10 lowest estimated values for test MSE.
sort(MSE.rep.cv)[1 : 10]
```

```
##  [1] 7.096110 7.114803 7.179631 7.180168 7.232859 7.330889 7.364629 7.365193
##  [9] 7.376770 7.461854
```

```r
# View the top 10 models in terms of the objective of minimising MSE.
order(MSE.rep.cv)[1 : 10]
```

```
##  [1] 125 111  63 109 127  95 107  61  47 123
```

```
# Construct a matrix in which to store information on which variables are included in the 10 best model
best.models <- matrix(NA, nrow = 10, ncol = length(variable.indices), dimnames = list(NULL, names(con)[v

# Cycle through the top 10 models and save TRUEs and FALSEs for columns of variables included and not i
for(i in 1 : 10)
{
  best.models[i, ] <- all.comb[order(MSE.rep.cv)[i], ] == 1
}

pander(best.models)
```

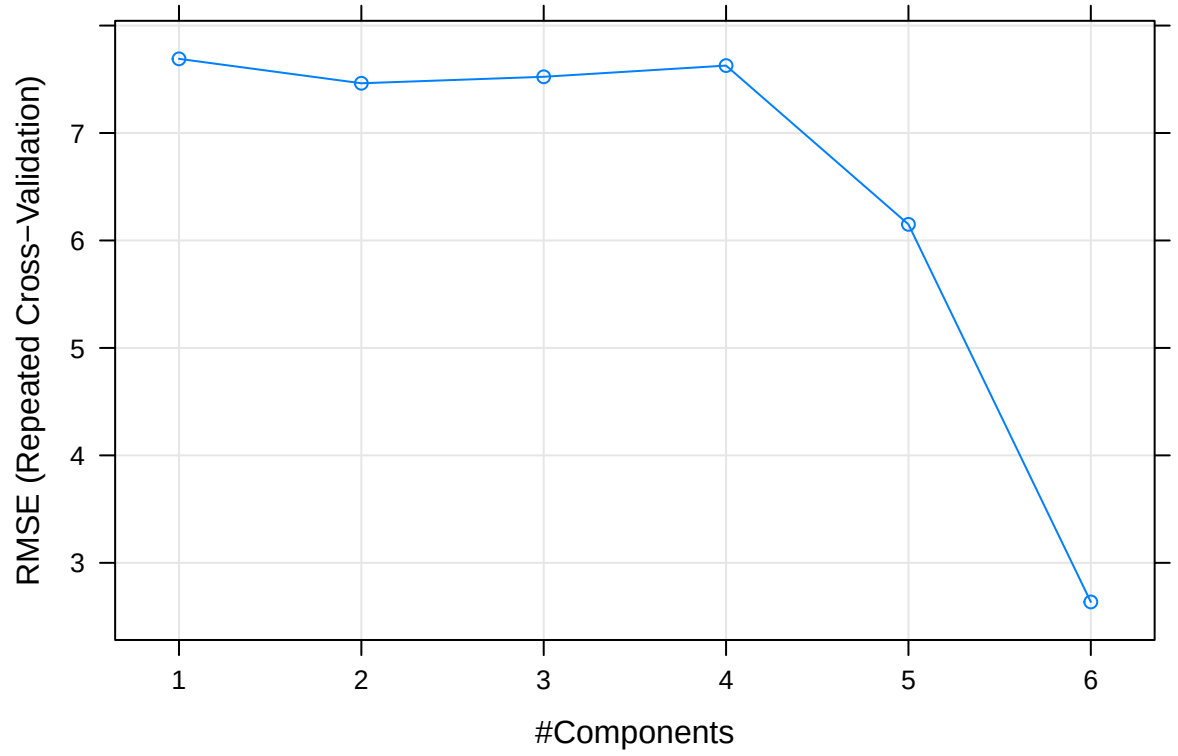
CEMENT	SLAG	FLY_ASH	WATER	SP	COARSE_AG	FINE_AG
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE

Using this approach, the predictors that would be included in the best model are `CEMENT`, `FLY_ASH`, `WATER`, `SP`, `COARSE_AG`, and `FINE_AG` (`SLAG` is excluded from the model).

e. **(3 marks)** Finally, use 20 repetitions of 10-fold cross-validation to find the optimal number of components if using principal component regression to predict the compression strength.

```
set.seed(0)

reps<-20
folds<-10

unregister_dopar <- function()
{
  env <-  foreach:::.foreachGlobals
  rm(list=ls(name=env),pos=env)
}
  unregister_dopar()

variable.indices <- 2 : 8 # columns of the variables to be used as predictors

# Carry out repeated k-fold cross-validation of PCR.
fitControl <- trainControl(method = "repeatedcv", number = folds, repeats = reps,
savePredictions = TRUE)
model.equation <- as.formula(paste("COMP_STRENGTH ~", paste(names(con)[variable.indices],
collapse = " + ")))
pcr <- train(model.equation, data = con, method = "pcr", scale = TRUE, tuneLength = 10,
trControl = fitControl)
```

```r
# Produce a plot of estimated RMSE vs. number of PCs.
plot(pcr)
```




```r
pander(cbind("PCs" = 1 : 6, "MSE" = as.numeric(pcr$results[, 2] ^ 2)))
```

PCs	MSE
1	59.14
2	55.69
3	56.6
4	58.18
5	37.82
6	6.948

```r
summary(pcr$finalModel)
```

```
## Data:    X dimension: 103 7
##  Y dimension: 103 1
## Fit method: svdpc
## Number of components considered: 6
## TRAINING: % variance explained
##           1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X         31.7796    53.40    69.24    83.61    93.34    99.96
## .outcome   0.7928    12.17    12.49    13.42    43.75    89.66
```

```
# Shut down cores.
stopCluster(nclust)
```

The optimal number of components if using principal component regression to predict the compression strength is 4 PCs which explain over 80% of the variance.

f. **(5 marks)** Compare the three different methods used for model selection in parts (c), (d), and (e):

- best subset selection using an exhaustive model search using `regsubsets` and the criteria of adjusted R^2, the C_p statistics, and/or BIC;
- best subset selection using an exhaustive model search using 20 repetitions of 10-fold cross-validation and the criterion of test MSE; and
- principal component regression using 20 repetitions of 10-fold cross-validation and the criterion of test MSE.

Briefly explain the relative advantages and disadvantages of the three methods and when we would prefer that particular method. Explain which method's results you would likely prefer here and why?

The advantages of using best subset selection using `regsubsets` is that it is computationally quicker than the other feature selection methods. It also explores all possible subsets of predictors. It can also compare models based on different criteria such as adjusted R^2, the C_p, and BIC. Disadvantages include that it is limited to numerical predictors only, and treats dummy variables as separate predictors. It can also only be used for categorical predictors than have 2 or less levels. `regsubsets` uses criteria to select its best models, and these criteria can give different results in practice which may also be a disadvantage. We prefer this method when we are short for time, and/or want to find the best model based on specific criteria where all possible subset combinations want to be explored.

Advantages of using best subset selection using an exhaustive model search using 20 reps of 10-fold CV is that all possible subsets of predictors are considered to form candidate models. These candidate models are compared with the model minimising test MSE. Another advantage is that this method utilises CV, and therefore helps to prevent overfitting by estimating model performance on unseen data. It gives a more accurate representation of predictive performance than best subset selection alone does. A disadvantage of this method is that it is more computationally expensive that the previous method, as the number of repetitions and folds in CV increase computational time. It is also important to note that the number of folds matters, as small K leads to higher bias, and that K is chosen based on the dataset size. This method is preferred when dealing with a larger number of predictors, or when model interpretability is not the main focus and predictive accuracy is.

Advantages of PCR is that it might provide better predictions of the response than linear regression does. More advantages are that PCR reduces dimensionality by transforming predictors into principal components, which can help solve issues of multicollinearity and reduce overfitting.This method is preferred in situations when reducing the number of predictors has minimal real world impacts in terms of time, money, or effort.

The goal of this task was to predict `COMP_STRENGTH` measured in MPa, using a linear regression of the response on the 7 ingredients (predictors). Based on this, I would prefer to use the results from best subset selection using an exhaustive model search using 20 repetitions of 10-fold cross-validation and the criterion of test MSE. This is because this method tells us what predictors are best in predicting `COMP_STRENGTH` while balancing between model interpretability, predictive accuracy, and computational efficiency. Using cross-validation provides a reliable estimate for predictive accuracy, yet avoids overfitting.

Assignment total: 40 marks