Question 1:d

```
> data<-read.csv("cement.csv")
> cor<-cor(data[,2:5])
> round(cor,3)

       x1     x2     x3     x4
x1  1.000  0.229 -0.824 -0.245
x2  0.229  1.000 -0.139 -0.973
x3 -0.824 -0.139  1.000  0.030
x4 -0.245 -0.973  0.030  1.000
```

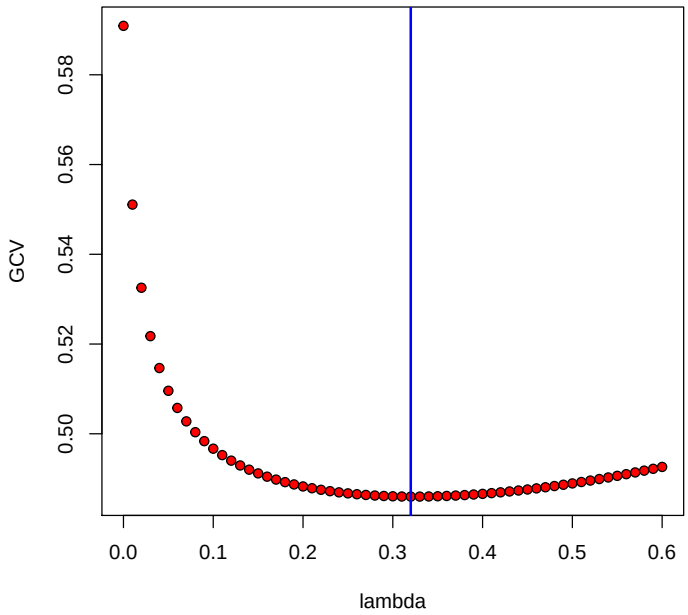
We see some very high correlation like 0.973 and .824. Nex we look at eigenvalues.

```
> eig<-eigen(cor)
> round(eig$val,4)

[1] 2.2357 1.5761 0.1866 0.0016
```

We also see some very low eigenvalues such as 0.0016.This and the high correlation value indicates that we have multicolliniearity.

```
> library(MASS)
> x<-scale(data[,2:5])
> y<-data$y-mean(data$y)
> lambdas <- seq(0,0.6,by = 0.010)
> eg.lmr<- lm.ridge(y~-1+x,lambda=lambdas)
> plot(lambdas,eg.lmr$GCV,pch = 21, bg="red",xlab = "lambda",ylab="GCV")
> lambda.min = lambdas[as.numeric(which(eg.lmr$GCV
+ ==min(eg.lmr$GCV)))]
> abline(v=lambda.min,lwd=2,col="blue")
> eg.lmr0 = lm.ridge(y~-1+x,lambda=lambda.min)
```



From the plot we see that the ideal lambda value for our regression is 0.32.
Prediction with new observation

```
> x0 <- cbind(10,50,20,40)
> x0.star = scale(x0,center=attr(x,"scaled:center"), scale=attr(x,"scaled:scale"))
> y0<-sum(eg.lmr0$coef*x0.star)+mean(data$y)
> print(y0)

[1] 94.2119
```

Thus our new prediction is 94.2111 from the ridge regression model and a lambda 0.32. Now to compare with our regular model.

```
> fit<-lm(y~x1+x2+x3+x4,data=data)
> predict(fit,newdata=data.frame(x1=10,x2=50,x3=20,x4=40))

       1
99.70052
```

With the old regression model our prediction become 99.70052. Considerably higher.

Question 2

A brief critique of Wine Quality : Correlations with Colour Density and Anthocyanin Equilibria in a Group of Young Red Wines by T. Chris Somers and Miceah T. Evans.

The first and major issue that this paper has is that there was is no mention or use of any technique more advanced than simple linear regression. No mention was made of the possible multicolinearity between the variables and no influence or outlier tests were performed.

With having essentially zero backround knowledge or understanding of the underlying science of wine tasting and their chemical properties I will restrict this critique to strictly statistical basis. This paper will first look at the influence and outlier measures of each of the regressions performed in this paper, and if needed remove some observations and see if the results vary from the ones presented in the paper. Then I will briefly look at an MLR model of the data and see if wine quality can be reliably predicted using the variables given. I will check for multicolinearity and perfrom a ridge regression if needed.

Influence Analysis

As said before, this will only look at the regression pairing presented in the paper.

Figure 1: Relation between quality and wine colour density

```
> data<-read.csv("wine_1.csv")
```

Wine Colour is designated x4 in our data and wine quality as y. We have two different wine types present in this data set Shiraz and Cabernet Sauvignon. One of the major questions of this paper is wether the statistical results are different for each of those types. Instead of running two regression to see if they are different, we will instead use an interaction variable between wine type or variable "x1" and colour density "x4." If this interaction is significant we will know the two wine types respond differently to the wine colour density. The result in the paper was that both of the lines were identical.

```
> library(xtable)
> dens<-data[,c("y","x_1","x_4")]
> ## define interaction
> dens$int<-dens$x_1*dens$x_4
> ##fit the model
> fit1<-lm(y~factor(x_1)+x_4+int,data=dens)
> ## print the results
> print(xtable(summary(fit1)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	11.7417	0.9644	12.17	0.0000
factor(x_1)1	-0.4447	1.5966	-0.28	0.7827
x_4	0.5392	0.1231	4.38	0.0002
int	-0.0180	0.2098	-0.09	0.9322

From this we can see that the regression is indeed significant with a p-value less than 0.005.We can also see that the both the interaction variable int and the dummy variable x1 are not statistically significant. Thus, we can confirm the paper's results.

```
> inflm<-influence.measures(fit1)
> table<-summary(inflm)

Potentially influential observations of
         lm(formula = y ~ factor(x_1) + x_4 + int, data = dens) :

   dfb.1_ dfb.f(_1 dfb.x_4 dfb.int dffit cov.r   cook.d hat
32  0.00  -0.40     0.00    0.28   -0.76  0.46_*  0.12   0.07
```

We can see that the that no observation were highly influential as our cook's D and hat values did not indicate a high degree of influence.

Figure 3: Wine Colour Density and Ionisation of Anthocyanins

Relation between wine colour density "x4" and degree of ionisation of anthocyanins "x9". Method is the same as before define interaction variable and run the model.

```
> iondens<-data[,c("x_4","x_1","x_9","x_10")]
> iondens$int<-data$x_1*data$x_9
> fit2<-lm(x_4~factor(x_1)+x_9+int,data=iondens)
> print(xtable(summary(fit2)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.7038	0.8956	1.90	0.0674
factor(x_1)1	1.7606	1.3122	1.34	0.1905
x_9	0.4200	0.0614	6.84	0.0000
int	-0.1697	0.0855	-1.99	0.0569

Here, we have a potentially contradictory result with the paper. Our overall regression is significant, however there is no differnce intercepts i.e. the value factor"x-1"1 is not statistically significant differnce in intercepts between Cabernet and Shiraz. The paper does not mention this but the way the lines are drawn it makes it seem like there is one. The paper is correct on the note that the slopes are different, the interaction variable is negative and has is statistically significant which shows that the slope for the Shiraz wine is lower than

Cabernet. However, the statement that the difference in slopes is certain with an alpha of 0.05 is misleading as the p-value of the t-test on the interaction variable is slightly greater than 0.05. Since the data set is quite small and it is not much greater than 0.05, we can say it is significant.

```
> inflm<-influence.measures(fit2)
> table<-summary(inflm)

Potentially influential observations of
         lm(formula = x_4 ~ factor(x_1) + x_9 + int, data = iondens) :

  dfb.1_ dfb.f(_1 dfb.x_9 dfb.int dffit   cov.r   cook.d hat
8  0.41  -0.28    -0.82    0.59   -1.19_*  0.43_*  0.28   0.14
```

We can see that that there is some slight influence in point 8. As the cook's D statistic is greater than the 4/n rule of thumb. We can try removing the observation and re-running the model to see if anything changes.

```
> iondens2<-iondens[1:31,]
> iondens2$int<-iondens2$x_1*iondens2$x_9
> fit2_2<-lm(x_4~factor(x_1)+x_9+int,data=iondens2)
> print(xtable(summary(fit2_2)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.7038	0.9036	1.89	0.0702
factor(x_1)1	1.4779	1.3830	1.07	0.2947
x_9	0.4200	0.0619	6.78	0.0000
int	-0.1550	0.0887	-1.75	0.0921

From this we can see that some of the values have changed, this could be because of the reduction in the degrees of freedom or the removal of an influential observation. The Rsquared value actually increased from 0.6974 to 0.7013 leads me to believe that observation 32 did have an adverse effect on our model. With observation 32 removed the int variable becomes less significant than it was before, as such we can say that there is no significan difference in the slopes between Cabernet and Shiraz variables.

```
> iondens$int<-data$x_1*data$x_10
> fit3<-lm(x_4~factor(x_1)+x_10+int,data=iondens)
> print(xtable(summary(fit3)))
```

This is another contradictory result. The coefficient of constituent term x1 is not statisically different than zero. The paper mentions specifically that the intercept term is different than zero, according to our tests it is not.

```
> inflm<-influence.measures(fit3)
> table<-summary(inflm)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.9736	0.5287	1.84	0.0762
factor(x_1)1	0.1260	0.8660	0.15	0.8853
x_10	130.5441	10.0308	13.01	0.0000
int	-15.4745	15.8930	-0.97	0.3386

```
Potentially influential observations of
         lm(formula = x_4 ~ factor(x_1) + x_10 + int, data = iondens) :

   dfb.1_ dfb.f(_1 dfb.x_10 dfb.int dffit   cov.r   cook.d hat
4  -0.84   0.51     1.33_*  -0.84    1.66_*  0.37_*  0.51   0.20
28  0.00   0.39     0.00    -0.35    0.51    1.51_*  0.06   0.30
```

Observation 4 seems to have some influence, we can try removing it see what happens.

```
> iondens3<-iondens[-4,]
> iondens3$int<-iondens3$x_1*iondens3$x_10
> fit3_2<-lm(x_4~factor(x_1)+x_10+int,data=iondens3)
> print(xtable(summary(fit3_2)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	1.3542	0.4682	2.89	0.0075
factor(x_1)1	-0.2546	0.7524	-0.34	0.7377
x_10	119.1157	9.2771	12.84	0.0000
int	-4.0462	14.0746	-0.29	0.7759

Int and our dummy variable changed alot, but they are still statistically insqignificant.

Figure 4: Polymetric Pgiments and Anthocynins

I am not going to worry about pigment-total anthocyanins plot as there is clearly no linear relationship there, or of any other kind for that matter.

```
> pigment<-data[,c("x_6","x_1","x_10")]
> pigment$int<-pigment$x_1*pigment$x_10
> fit4<-lm(x_6~factor(x_1)+x_10+int,data=pigment)
> print(xtable(summary(fit4)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.1967	0.2637	0.75	0.4619
factor(x_1)1	0.1774	0.4319	0.41	0.6844
x_10	34.9211	5.0029	6.98	0.0000
int	-10.6810	7.9268	-1.35	0.1886

In this case, as in some of the others there the intercept value does end up being statistically insignificant. I believe this is a result of the use of interaction variables in addition to the use of dummy variables.

```
> inflm<-influence.measures(fit4)
> table<-summary(inflm)

Potentially influential observations of
         lm(formula = x_6 ~ factor(x_1) + x_10 + int, data = pigment) :

  dfb.1_ dfb.f(_1 dfb.x_10 dfb.int dffit   cov.r   cook.d hat
4 -0.93   0.57     1.48_*  -0.93    1.84_*  0.28_*  0.58   0.20
```

The fourth observation has quite a high cook's d value we should look at its effect on the model.

```
> pigment2<-pigment[-4,]
> pigment2$int<-pigment2$x_1*pigment2$x_10
> fit4_2<-lm(x_6~factor(x_1)+x_10+int,data=pigment2)
> print(xtable(summary(fit4_2)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.4010	0.2259	1.77	0.0872
factor(x_1)1	-0.0268	0.3630	-0.07	0.9416
x_10	28.7874	4.4759	6.43	0.0000
int	-4.5473	6.7906	-0.67	0.5088

This point was clearly quite influential as we can see that the slope value on the variable x-10 changed by 4. This did not change th significance of the other parameters.

Figure 5: Quality Ratings and Degree of Ionisation of anthocyanins

```
> quality<-data[,c("y","x_1","x_9")]
> quality$int<-quality$x_1*quality$x_9
> fit5<-lm(y~factor(x_1)+x_9+int,data=quality)
> print(xtable(summary(fit5)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	12.1641	0.8849	13.75	0.0000
factor(x_1)1	0.6599	1.2966	0.51	0.6148
x_9	0.2636	0.0607	4.35	0.0002
int	-0.1149	0.0844	-1.36	0.1846

The paper shows the plot of this regression as two distinct paprallel lines. This however, is not supported by our analysis. We do not get a significant change of slope. The regression is significant like the paper states.

```
> inflm<-influence.measures(fit5)
> table<-summary(inflm)

Potentially influential observations of
         lm(formula = y ~ factor(x_1) + x_9 + int, data = quality) :

   dfb.1_ dfb.f(_1 dfb.x_9 dfb.int dffit cov.r   cook.d hat
15  0.00   0.10     0.00    0.07    0.70  0.43_*  0.10   0.06
17  0.00   0.06     0.00   -0.07   -0.12  1.52_*  0.00   0.24
28  0.00   0.16     0.00   -0.13    0.22  1.45_*  0.01   0.22
```

Cook's D values are all quite low and therefore we do not need to do anything here.

MLR

In this section we seek to build a predictive model for wine quality.

First we check for multicolinearity

```
> cor<-cor(data[,2:11])
> round(cor,3)

        x_1    x_2    x_3    x_4    x_5    x_6    x_7    x_8    x_9   x_10
x_1   1.000 -0.089  0.115 -0.009  0.013 -0.164  0.143  0.050  0.164  0.143
x_2  -0.089  1.000 -0.582  0.213  0.152  0.220  0.086  0.096 -0.049  0.086
x_3   0.115 -0.582  1.000 -0.391 -0.371 -0.325 -0.369  0.405 -0.496 -0.369
x_4  -0.009  0.213 -0.391  1.000  0.996  0.945  0.937  0.016  0.797  0.937
x_5   0.013  0.152 -0.371  0.996  1.000  0.925  0.959  0.003  0.826  0.959
x_6  -0.164  0.220 -0.325  0.945  0.925  1.000  0.780 -0.043  0.691  0.780
x_7   0.143  0.086 -0.369  0.937  0.959  0.780  1.000  0.037  0.847  1.000
x_8   0.050  0.096  0.405  0.016  0.003 -0.043  0.037  1.000 -0.456  0.037
x_9   0.164 -0.049 -0.496  0.797  0.826  0.691  0.847 -0.456  1.000  0.847
x_10  0.143  0.086 -0.369  0.937  0.959  0.780  1.000  0.037  0.847  1.000
```

We see some very high correlations such 0.959 and 0.937. Suggesting high multicolinearity.

```
> eig<-eigen(cor)
> round(eig$val,4)

 [1] 5.6614 1.5564 1.3199 1.0058 0.3256 0.1097 0.0199 0.0012 0.0000 0.0000
```

Similarly some of the eigenvalues are quite low as well.

Model Testing

We do this mostly because we can, it would be interesting to see the predictive value of the Ridge regression when we have some obviously theoretically linked values such degree of ionization of anthocynins in percent and ionized

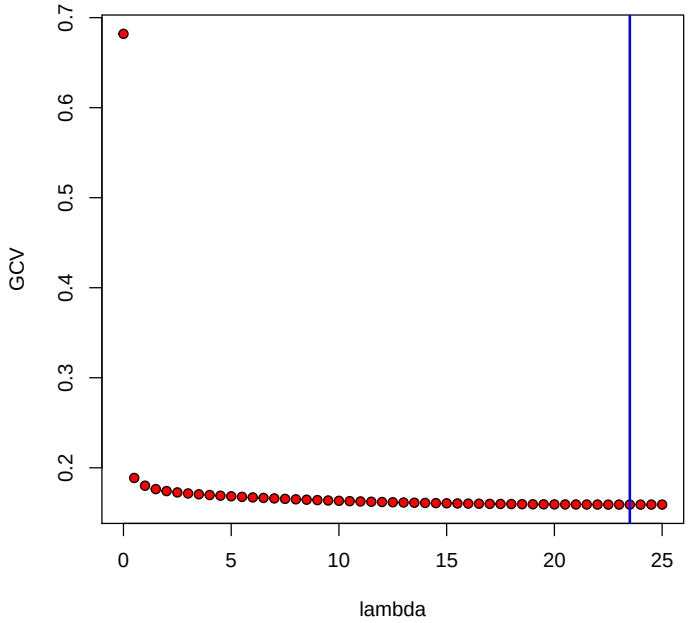
anthocynins total. I will perform Ridge Regression and then perform variable selection and compare the resulting models. My methodology will be as follows. I will split the data into a learning and testing set. The learning test wil bewhat I calibrate the models on and testing will be what I evaluate them on.

```
> splitdf <- function(dataframe, seed=105) {
+     if (!is.null(seed)) set.seed(seed)
+     index <- 1:nrow(dataframe)
+     trainindex <- sample(index, trunc(length(index)/2))
+     trainset <- dataframe[trainindex, ]
+     testset <- dataframe[-trainindex, ]
+     list(trainset=trainset,testset=testset)
+ }
> sets<-splitdf(data)
> train<-sets$trainset
> test<-sets$testset
```

We split our data set into two randomly selected halfs. Now we will perform our regressions.

Ridge Regression

```
> library(MASS)
> x<-scale(train[,2:11])
> y<-train$y-mean(train$y)
> lambdas <- seq(0,25,by = 0.5)
> eg.lmr<- lm.ridge(y~-1+x,lambda=lambdas)
> plot(lambdas,eg.lmr$GCV,pch = 21, bg="red",xlab = "lambda",ylab="GCV")
> lambda.min = lambdas[as.numeric(which(eg.lmr$GCV
+ ==min(eg.lmr$GCV)))]
> abline(v=lambda.min,lwd=2,col="blue")
> eg.lmr0 = lm.ridge(y~-1+x,lambda=lambda.min)
```



We know our lambda now we will test our model on the test data set.

```
> predicted_ridge<-1:16
> for (i in 1:length(test)) {
+     x0<-test[i,2:11]
+     x0.star = scale(x0,center=attr(x,"scaled:center"), scale=attr(x,"scaled:scale"))
+     y0<-sum(eg.lmr0$coef*x0.star)+mean(train$y)
+     predicted_ridge[i]<-y0
+ }
```

Variable Selection Method

Will use basic stepwise selection, using both directions.

```
> fit_null<-lm(y~factor(x_1)+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_10,data=train)
> step3<-stepAIC(fit_null, direction="both")

> print(xtable(summary(step3)))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-6.4403	21.3424	-0.30	0.7690
factor(x_1)1	-1.5561	0.8855	-1.76	0.1094
x_2	6.6253	5.5301	1.20	0.2585
x_3	-0.0263	0.0162	-1.63	0.1347
x_4	-6.0413	2.3706	-2.55	0.0289
x_5	10.2320	3.7039	2.76	0.0200

We see that we end up using x-2 "ph", x-3 "Total S02",x-4-"color density" and x-5 wine colour. Now to see how well our model fares with the test data set.

```
> predicted_step<-predict(step3,test)
> results<-data.frame(test$y,predicted_ridge,predicted_step)
> print(xtable(results))
```

	test.y	predicted_ridge	predicted_step
2	18.30	17.15	19.95
3	17.10	16.68	19.39
5	16.80	15.87	17.98
6	16.50	17.04	18.68
7	15.80	14.84	16.40
8	15.20	15.57	17.81
9	15.20	14.77	16.14
11	14.00	14.12	15.77
17	16.30	16.87	17.96
19	16.00	16.80	16.92
23	15.30	16.02	16.58
25	14.80	12.00	17.40
26	14.30	13.00	14.01
27	14.30	14.00	15.52
28	14.20	15.00	6.82
30	13.80	16.00	16.40

As we can can see the ridge seemed to have slighly better predicitive value than the model chosen by stepwise regression. If we want to look at a very basic numeric summary of predictive value, mean squared error.

```
> results$error_ridge<-results[,1]-results[,2]
> results$error_step<-results[,1]-results[,3]
> mse_ridge<-sum(results$error_ridge^2)/15
> mse_step<-sum(results$error_step^2)/15
> mse_ridge

[1] 1.36557

> mse_step

[1] 6.667344
```

Clearly, the Ridge regression outperformed stepwise model selection for train and test data sets. One should note that this was for a very small data set these kinds of results are very unreliable and I would not speculate in the wine market using any of these models. For example, if we set the set as something different I would wager we would get drastically different results.