Mini Project #6

Sudarshana Jagadeeshi

Contribution of each group member: I completed the project in full

**Section 1**

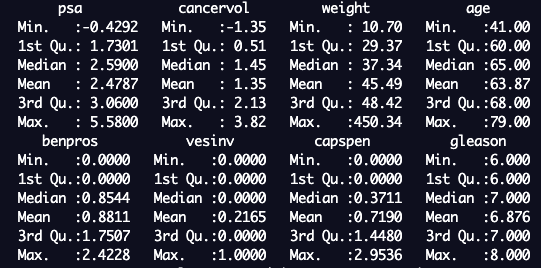
**1.**

We first start by loading the data and removing the subject column. It is just for indexing and should not be used as a predictor. Vesinv is already in 0,1 one-hot form, so I leave it alone. We then plot density curvesof some of the prominent numerical columns:

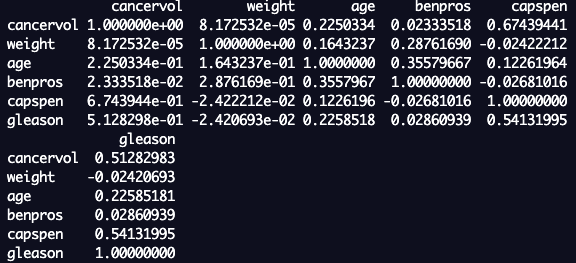
|  |  |
| --- | --- |
|  |  |
|  |  |

Of these 5, we can see that weight is approximately normal, but some other columns are not. Psa has some right skew to it, and could be a candidate to log normalize. The other three columns (capspen, benpros, and cancervol) are definitely non-normal, and will need to be normalized.

After we normalize these, we summarize the dataframe, to get a better picture of our dataset.

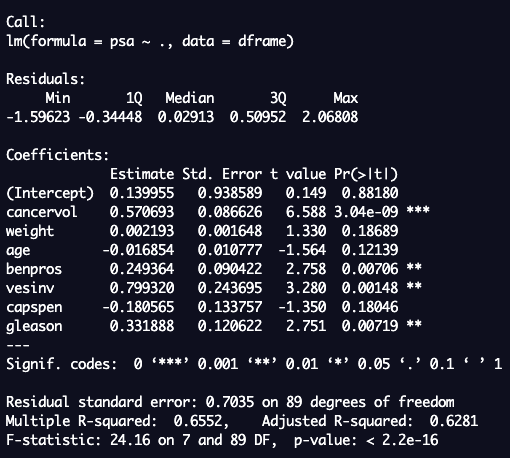


We now need to decide which columns to eliminate, to prevent overfitting our model. We can get the correlation matrix for this purpose. It will tell us which columns predict/are related to each other, so that we may remove some of these redundant columns.



The matrix has broken onto the second line, but we can see that cancervol and gleason are highly similar, as well as capspen and cancervol, benpros and weight.

We now fit our initial model.

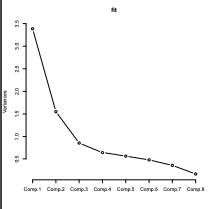


We see four variables with a high significance here. However, there are also variables with larger p-values, like weight and capsen. We can eliminate the worst variable each time, observing the RSE, which is the sqrt of the MSE. The results are shown below:

|  |  |
| --- | --- |
|  |  |
|  |  |

As we can see, the RSE increases slightly until it reaches the model in the middle left. However, after that model, it proceeds to get much larger. It seems that psa ~ cancervol + benpros + vesinv + gleason is the winning formula, as it is a reasonably good fit that doesn’t overfit the model.

We can further confirm this by plotting a scree plot. This is something I learned in my ML class- we should look for where the graph begins to level off, or the “elbow” of the data. It seems the elbow here is at 3 or 4, which means 3 or 4 predictor variables will do an satisfactory job of capturing the variance of the data.



I have all but decided on model 4. I further plot some QQ plots of the residuals to see if they are distributed evenly…

|  |  |
| --- | --- |
|  | Top Left: Model 4  Above: Model 7 (predicting only with cancervol)  Left: Model 1(Using all variables) |

I am pleased with this result and note that model 7 is insufficient to capture the intricacies of the data, while model 4 and 1 and fitting the line well, with 4 doing even better than 1.

Choosing model4, we call the predict function with the parameters described in the description. The result is 10.71089. I looked through the dataset for similar patients with parameters (cancervol, benpros, vesinv,gleason)= (7,2.5,0,7). They were patients 22 and 62, and the psa were 5 and 21 respectively. It seems that 10 splits the difference and could be reasonable.

**Section 2**

**Question 1**

**library(graphics) #for scree plot**

**#1. Load/Normalize data**

**dframe <- read.csv("prostate\_cancer.csv", header= TRUE) #first row is NOT data**

**dframe <- subset (dframe, select = -subject) #we don't want subject**

**plot(density(dframe$psa)) #plot density curves**

**plot(density(dframe$cancervol))**

**plot(density(dframe$weight))**

**plot(density(dframe$benpros))**

**plot(density(dframe$capspen))**

**dframe$psa <- log(dframe$psa) #start transforming data**

**dframe$capspen <- log(dframe$capspen + 1) #+1 because there are 0's in the dataset**

**dframe$benpros <- log(dframe$benpros + 1)**

**dframe$cancervol <- log(dframe$cancervol)**

**#2. Analyze correlaton**

**print(summary(dframe))**

**colscorr <- c(2,3,4,5,7,8) #we cannot put vesinv in the cor table**

**print(cor(dframe[,colscorr]))**

**#3. Start constructing a variety of models**

**model <- lm(psa ~ ., data=dframe) # ~. means all variables besides psa**

**print(summary(model))**

**#remove the age attribute, its p-value is the worst**

**model2 <- lm(psa ~ cancervol + weight + benpros + vesinv + capspen + gleason, data=dframe)**

**print(summary(model2))**

**#remove the weight attribute, its p-value is the worst**

**model3 <- lm(psa ~ cancervol + benpros + vesinv + capspen + gleason, data=dframe)**

**print(summary(model3))**

**#remove the capspen attribute, its p-value is the worst**

**model4 <- lm(psa ~ cancervol + benpros + vesinv + gleason, data=dframe)**

**print(summary(model4))**

**#All the remaining attributes have good p-values, but lets go further and remove gleason, just to test**

**model5 <- lm(psa ~ cancervol + benpros + vesinv, data=dframe)**

**print(summary(model5))**

**#Remove vesinv**

**model6 <- lm(psa ~ cancervol + benpros, data=dframe)**

**print(summary(model6))**

**#Remove benpros**

**model7 <- lm(psa ~ cancervol, data=dframe)**

**print(summary(model7))**

**#Do PCA, not necessary, but affirms assumptions**

**PCA <- princomp(dframe, cor=TRUE)**

**plot(PCA,type="lines")**

**#4. Residual plots**

**plot(fitted(model4), resid(model4))**

**abline(h=0)**

**qqnorm(resid(model4))**

**qqline(resid(model4))**

**plot(fitted(model7), resid(model7))**

**abline(h=0)**

**qqnorm(resid(model7))**

**qqline(resid(model7))**

**plot(fitted(model), resid(model))**

**abline(h=0)**

**qqnorm(resid(model))**

**qqline(resid(model))**

**#5. do a prediction**

**#the mode of vensinv and gleason is 0 and 7**

**#exp is necessary because we log transformed psa**

**typical <- data.frame(cancervol = mean(dframe$cancervol), benpros= mean(dframe$benpros), vesinv= 0, gleason= 7)**

**prediction <- exp(predict(model4, typical))**

**print(prediction)**