UNI

HW1 Abhay Pawar (asp2197)

1

(a) $p(x_i = j \mid \pi, r) = \binom{x_i + r - 1}{x_i} \pi^{x_i} (1-\pi)^r$

Since, x_i's are i.i.d →

$p(x_1, x_2 .. x_N \mid \pi, r) = p(x_1 \mid \pi, r)\, p(x_2 \mid \pi, r) ... p(x_N \mid \pi, r)$

Joint likelihood $= \boxed{\prod_{i=1}^{N} \binom{x_i + r - 1}{x_i} \pi^{\sum_{i=1}^{N} x_i} (1-\pi)^{rN}}$

(b) Taking ln of the joint pdf →

$$\ln(L) = \sum_{i=1}^{N} \ln\left[\binom{x_i + r - 1}{x_i}\right] + \ln(\pi) \sum_{i=1}^{N} x_i + rN \ln(1-\pi)$$

To find MLE w.r.t. π →

$$\frac{\partial \ln(L)}{\partial \pi} = 0 + \frac{\sum_{i=1}^{N} x_i}{\pi} + \frac{rN}{(1-\pi)}(-1) = 0$$

$\therefore \sum x_i - \pi \sum x_i = rN\pi$

$\therefore$ $\boxed{\hat{\pi}_{ML} = \frac{\sum x_i}{(rN + \sum x_i)}}$

(c) $\pi_{MAP} = \arg\max_{\pi} \ln p(\pi \mid x_i\text{'s}, r)$

$= \arg\max_{\pi} \left[\ln(x_i\text{'s} \mid \pi, r) + \ln p(\pi)\right]$

Ignoring third term because it is not dependent on π.

$$p(\pi) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1} (1-\pi)^{b-1}$$

$$\therefore \pi_{MAP} = \arg\max_{\pi} \left[\ln(\pi) \sum_{i=1}^{N} x_i + rN \ln(1-\pi) + (a-1)\ln(\pi) + (b-1)\ln(1-\pi)\right]$$

Excluded all terms which were not a function of π

Taking derivative-

$$\frac{\sum_{i=1}^{N} x_i}{\pi} - \frac{rN}{(1-\pi)} + \frac{a-1}{\pi} - \frac{(b-1)}{(1-\pi)} = 0$$

$$\frac{\sum x_i + a - 1}{\pi} - \frac{(rN + b - 1)}{(1-\pi)} = 0$$

$$\therefore \pi_{MAP} = \frac{\sum x_i + a - 1}{\sum x_i + a + b + rN - 2}$$

(d) $p(\pi \mid x_i's, r) = \dfrac{p(x_i's \mid \pi, r)\, p(\pi)}{p(x_i \mid r)}$ — I

Writing only the numerator $= p(x_i's \mid \pi, r)\, p(\pi)$

$$= \prod_{i=1}^{N} \binom{x_i + r - 1}{x_i} \pi^{\sum x_i} (1-\pi)^{rN} \times \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \pi^{a-1} (1-\pi)^{b-1}$$

$$= K \times \pi^{\sum x_i + a - 1} (1-\pi)^{rN + b - 1}$$

(K is a term not involving π)

Also, the denominator of I does not depend on π

$$\therefore p(\pi \mid x_i's, r) = K_1\, \pi^{\sum x_i + a - 1} (1-\pi)^{rN + b - 1}$$

(K_1 is not a funcⁿ of π).

This is again a beta funcⁿ. with

$a' = \sum x_i + a$, $b' = rN + b$

$$\therefore K_1 = \frac{\Gamma(a' + b')}{\Gamma(a')\Gamma(b')}$$

$$\therefore p(\pi \mid x_i's, r) = \frac{\Gamma(\sum x_i + rN + a + b)}{\Gamma(\sum x_i + a)\Gamma(rN + b)} \pi^{\sum x_i + a - 1} (1-\pi)^{rN + b - 1}$$

(e) For Beta dist, $\mu = a/(a+b)$ & $Var = ab / [(a+b)^2 (a+b+1)]$

$$\therefore E(\pi) = \frac{\sum x_i + a}{(\sum x_i + a + rN + b)}$$

$$Var(\pi) = \frac{(\sum x_i + a)(rN + b)}{(\sum x_i + a + rN + b)^2 (\sum x_i + a + b + rN + 1)}$$

Comparison –

$$E(\pi) = \frac{\sum x_i + a}{\sum x_i + a + rN + b} = \frac{\sum x_i (1 + a/\sum x_i)}{(\sum x_i + rN)\left(1 + \frac{a+b}{\sum x_i + rN}\right)}$$

$$= \hat{\pi}_{ML} \times \left[\frac{1 + a/\sum x_i}{1 + (a+b)/(\sum x_i + rN)}\right]$$

$\therefore E(\pi) > \hat{\pi}_{ML}$ when the multiplicating factor is >1.

Similarly,

$$E(\pi) = \pi_{MAP} \times \left[\frac{1 + (a-1)/\sum x_i}{1 + (a+b-2)/(\sum x_i + rN)}\right]$$

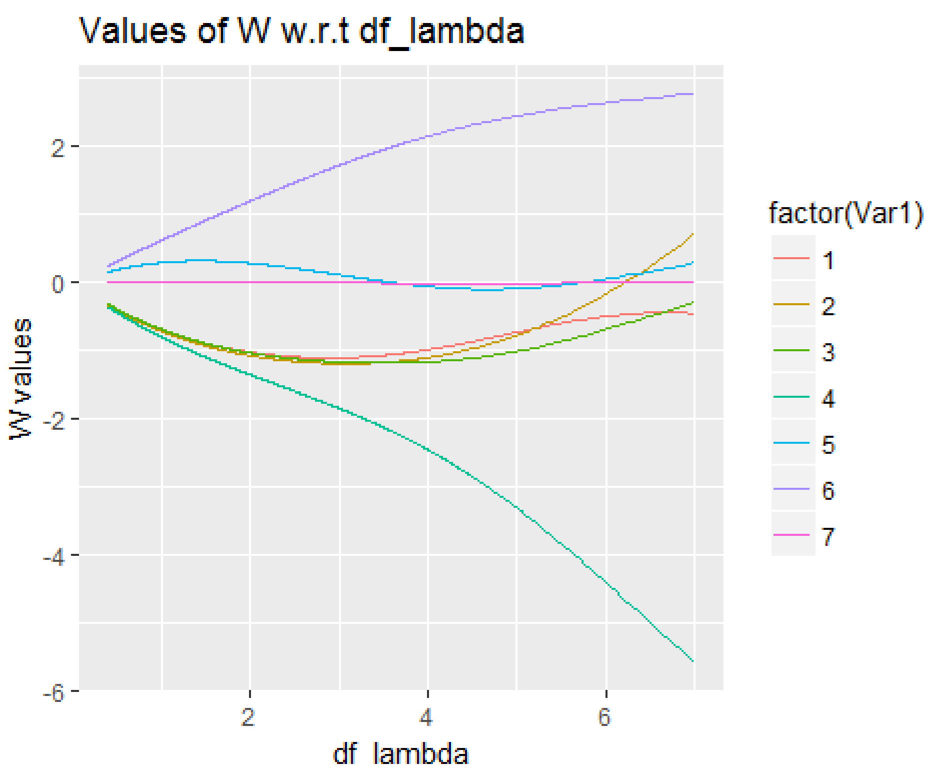
For $N \to \infty$,

$$E(\pi) = \hat{\pi}_{ML} = \pi_{MAP}$$

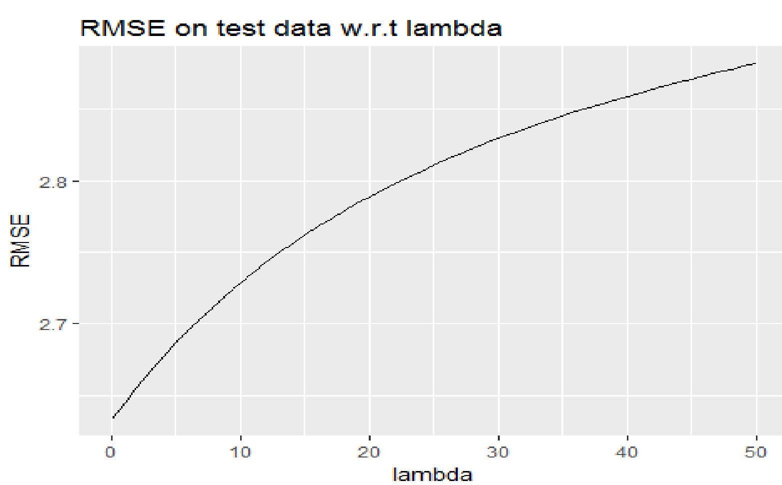
Problem 2(coding):

Part 1:

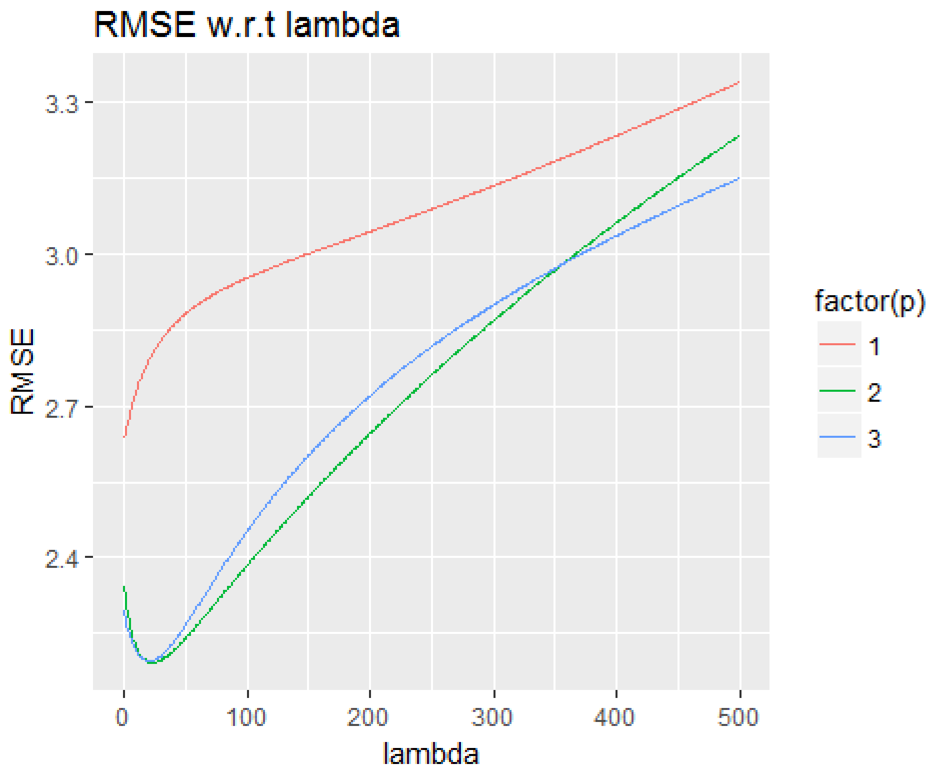
(a) & (b) Values for 4th and 6th dimension are the highest in magnitude. Since, all the dimensions are normalized, w of a dimension will be high when it is highly correlated with the dependent variable (miles/gallon) and can predict the value of dependent variable well. Dimension 4 (car weight) has a negative w, implying negative correlation and Dimension 6 has a positive w, implying positive correlation.



(c) Lambda=0 gives the least value of RMSE and hence, least squares should be used. Ridge regression has worse RMSE than Least Squares solution.



(d) I will **choose p=2** because it gives minimum RMSE for some value of lambda as compared to p=1 and p=3. The **ideal value of lambda** is no more zero and it is **21.** It implies that ridge regression with lambda=21 works better than just the OLS solution for p=2.



```
library(readr)
library(reshape2)
X_test <- read_csv("G:/Acads/ML for data science/hw1-data/X_test.csv",
                   col_names = FALSE)
X_train <- read_csv("G:/Acads/ML for data science/hw1-data/X_train.csv",
                   col_names = FALSE)
y_test <- read_csv("G:/Acads/ML for data science/hw1-data/y_test.csv",
                   col_names = FALSE)
y_train <- read_csv("G:/Acads/ML for data science/hw1-data/y_train.csv",
                   col_names = FALSE)

X_test_mat=as.matrix(X_test)
X_train_mat=as.matrix(X_train)
y_test_mat=as.matrix(y_test)
y_train_mat=as.matrix(y_train)

lambda=c(0:5000)
df_lambda=array(dim=length(lambda))
RMSE=array(dim=length(lambda))
W_mat=matrix(nrow=ncol(X_train),ncol=length(lambda))

#Calculating df_lambda, W_mat, RMSE on Y_test and y_pred on test data
for (i in 1:length(lambda)){
  W_mat[,i]=solve(lambda[i]*diag(ncol(X_train_mat))+((t(X_train_mat))%*%X_train_mat))%*%t(X_trai
n_mat)%*%y_train_mat
  df_lambda[i]=sum(diag(X_train_mat%*%solve(lambda[i]*diag(ncol(X_train_mat))+t(X_train_mat)%*%X
_train_mat)%*%t(X_train_mat)))
  y_pred=X_test_mat%*%W_mat[,i]
  RMSE[i]=sqrt(sum((y_pred-y_test_mat)^2)/length(y_pred))
}

W_mat_melt <- melt(W_mat, id=c())
W_mat_melt$df_lambda=0
for (i in 1:length(W_mat_melt$df_lambda)){
  W_mat_melt$df_lambda[i]=df_lambda[W_mat_melt$Var2[i]]
}
df_lambda[W_mat_melt$Var2[10]]
W_mat_melt$Var2[10]

ggplot(data=W_mat_melt)+geom_line(mapping=aes(y=value,x=df_lambda,colour=factor(Var1)))+
  labs(title="Values of W w.r.t df_lambda",x="df_lambda",y="W values")

rmse_frame=data.frame(RMSE,lambda)
ggplot(data=rmse_frame[1:51,])+geom_line(aes(y=RMSE,x=lambda))+labs(title="RMSE on test data w.
r.t lambda")

#p=2
X_train_mat_cr=matrix(nrow=nrow(X_train_mat),ncol=13)
X_train_mat_cr[,1:7]=X_train_mat
X_test_mat_cr=matrix(nrow=nrow(X_test_mat),ncol=13)
X_test_mat_cr[,1:7]=X_test_mat
W_mat_cr=matrix(nrow=ncol(X_train_mat_cr),ncol=length(lambda))
for (i in 1:6){
```

```
  X_train_mat_cr[,i+7]=X_train_mat[,i]^2
  X_test_mat_cr[,i+7]=X_test_mat[,i]^2
}
for (i in 1:length(lambda)){
  W_mat_cr[,i]=solve(lambda[i]*diag(ncol(X_train_mat_cr))+
((t(X_train_mat_cr))%*%X_train_mat_cr))%*%t(X_train_mat_cr)%*%y_train_mat
  #df_lambda[i]=sum(diag(X_train_mat_cr%*%solve(lambda[i]*diag(ncol(X_train_mat_cr))+t(X_train_m
at_cr)%*%X_train_mat_cr)%*%t(X_train_mat_cr)))
  y_pred=X_test_mat_cr%*%W_mat_cr[,i]
  RMSE[i]=sqrt(sum((y_pred-y_test_mat)^2)/length(y_pred))
}

rmse_frame_cr=data.frame(RMSE,lambda)
ggplot(data=rmse_frame_cr[1:501,])+geom_line(aes(y=RMSE,x=lambda))+labs(title="RMSE on test data
(p=2) w.r.t lambda")

#p=3
X_train_mat_cr3=matrix(nrow=nrow(X_train_mat),ncol=19)
X_train_mat_cr3[,1:13]=X_train_mat_cr
X_test_mat_cr3=matrix(nrow=nrow(X_test_mat),ncol=19)
X_test_mat_cr3[,1:13]=X_test_mat_cr
W_mat_cr3=matrix(nrow=ncol(X_test_mat_cr3),ncol=length(lambda))
for (i in 1:6){
  X_train_mat_cr3[,i+13]=X_train_mat[,i]^3
  X_test_mat_cr3[,i+13]=X_test_mat[,i]^3
}
for (i in 1:length(lambda)){
  W_mat_cr3[,i]=solve(lambda[i]*diag(ncol(X_train_mat_cr3))+((t(X_train_mat_cr3))%*%X_train_mat_
cr3))%*%t(X_train_mat_cr3)%*%y_train_mat
  #df_lambda[i]=sum(diag(X_train_mat_cr%*%solve(lambda[i]*diag(ncol(X_train_mat_cr3))+t(X_train_
mat_cr3)%*%X_train_mat_cr3)%*%t(X_train_mat_cr3)))
  y_pred=X_test_mat_cr3%*%W_mat_cr3[,i]
  RMSE[i]=sqrt(sum((y_pred-y_test_mat)^2)/length(y_pred))
}

rmse_frame_cr3=data.frame(RMSE,lambda)
ggplot(data=rmse_frame_cr3[1:501,])+geom_line(aes(y=RMSE,x=lambda))+labs(title="RMSE on test dat
a(p=3) w.r.t lambda")
#Ideal value of lambda=21

rmse_frame$p=1
rmse_frame_cr$p=2
rmse_frame_cr3$p=3
rmse_all=rbind(rmse_frame[1:501,],rmse_frame_cr[1:501,],rmse_frame_cr3[1:501,])
ggplot(data=rmse_all)+geom_line(aes(y=RMSE,x=lambda,colour=factor(p)))+labs(title="RMSE w.r.t la
mbda")
```