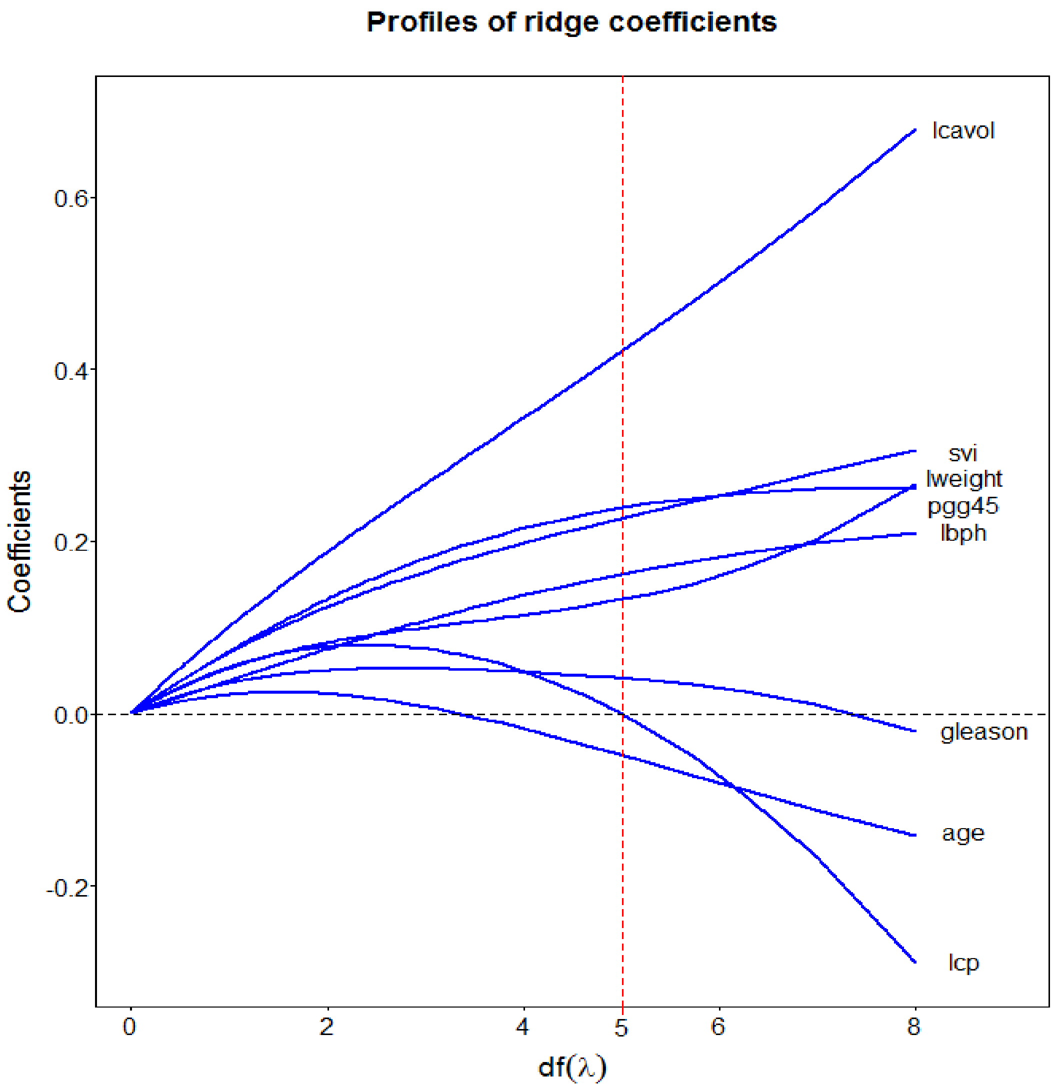
104225015 Lin Yuan-Hsin

High-dimensional data analysis HW3

Reproduce **Figure 3.8 3.10** and **Table 3.3**. (do not need PCR PLS Std Error)

Figure 3.8



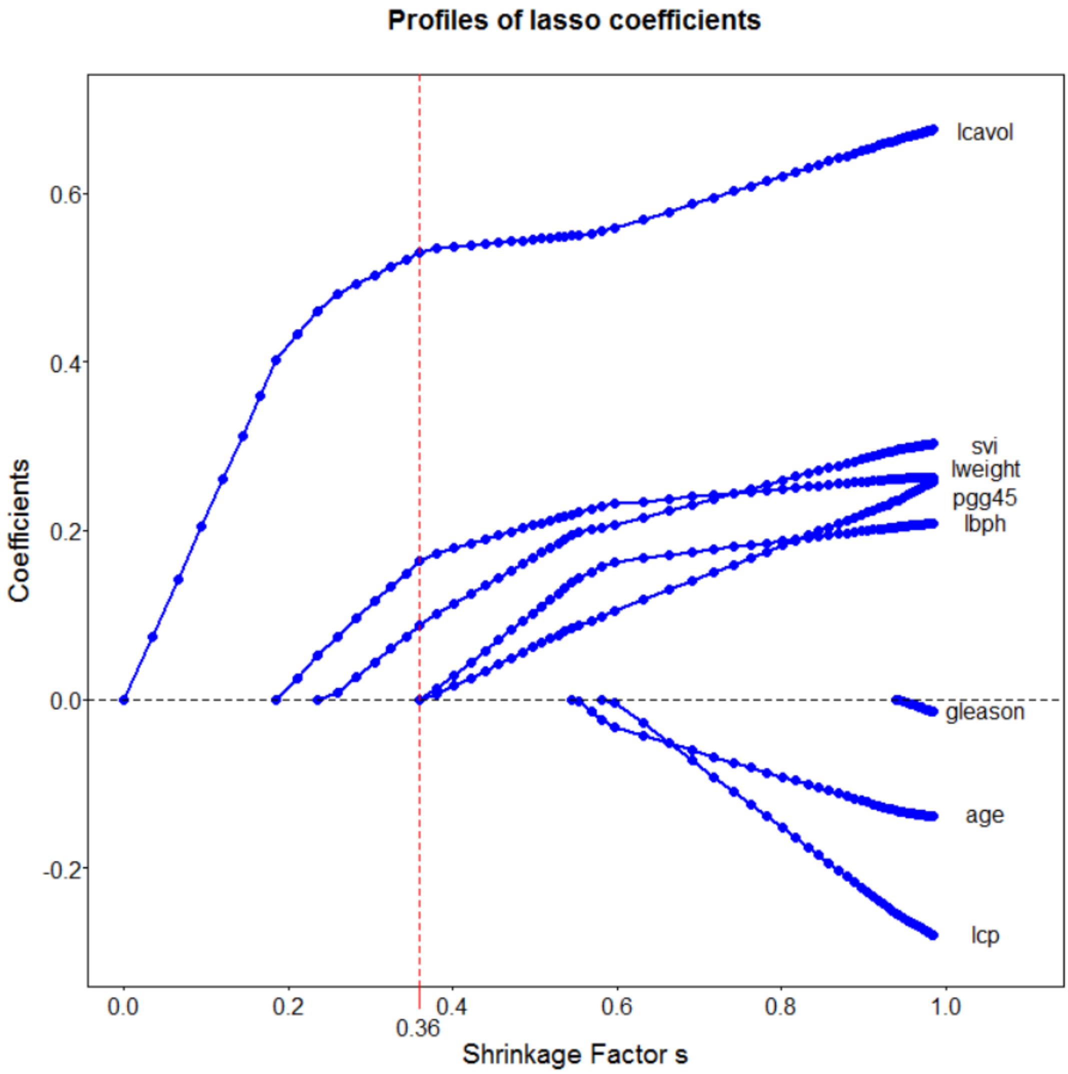
There is Profiles of ridge coefficients for the prostate cancer example, as the tuning parameter is varied. Coefficients are plotted versus the effective degrees of freedom. We use effective degree of freedom $df(\lambda) = trace[X(X^TX + \lambda I)^{-1}X^T]$, where X is centered inputs.Vertical line is drawn at $df(\lambda) = 5.0002 \cong 5$, the value chosen by cross validation. When the $df(\lambda) \cong 5$, we found that

$\hat{\beta}_{lcavol} = 0.4210001259$ $\hat{\beta}_{lweight} = 0.2387919865$

$\hat{\beta}_{age} = -0.0480240287$ $\hat{\beta}_{lbph} = 0.1623195446$ $\hat{\beta}_{svi} = 0.2271296444$

$\hat{\beta}_{lcp} = -0.0000998623$ $\hat{\beta}_{gleason} = 0.0410749002$ $\hat{\beta}_{pgg45} = 0.1324521648$

Figure 3.10



There is Profiles of lasso coefficients, as the tuning parameter t is varied. Coefficients are plotted versus $s = \sum_{i=1}^{p} |\hat{\beta}_i^{lasso}| \,/\, \sum_{i=1}^{p} |\hat{\beta}_i|$. A vertical line is drawn at $s = 0.36$, the value chosen by cross-validation. When the $s = 0.36$, we use interpolation method, Arithmetic mean or directly from the package to find out the parameters.

The following are parameters which calculated from the "glmnet" package:

$\hat{\beta}_{lcavol} = 0.533054114$ $\hat{\beta}_{lweight} = 0.169735954$

$\hat{\beta}_{lbph} = 0.002973318$ $\hat{\beta}_{svi} = 0.094376956$

Interpolation method and Arithmetic mean are in the output (Lasso1 and Lasso2)

Table 3.3

Term	**LS**	**Best Subset**	**Ridge**	**Lasso**
Intercept	2.46493	2.4773573	2.4523450851	2.46836
Lcavol	0.67953	0.7397137	0.4210001259	0.533054114
Lweight	0.26305	0.3163282	0.2387919865	0.169735954
Age	-0.14146		-0.0480240287	
Lbph	0.21015		0.1623195446	0.002973318
Svi	0.30520		0.2271296444	0.094376956
Lcp	-0.28849		-0.0000998623	
Gleason	-0.02131		0.0410749002	
Pgg45	0.26696		0.1324521648	
Test Error	0.521274	0.4924823	0.4915854	0.4781344

We compared $\text{Test Error} = E\left(Y_0 - \hat{Y}\right)^2 = E\left(Y_0 - X_0\hat{\beta}\right)^2$ from each model.

And in the method of the best subset selection, we use the Bayesian information criterion (BIC) to adjudicate the best model. Therefore, the subset including lcavol and lweight has the minimum $BIC = -51.30$. So we choose that for best subset selection.

	(Intercept)	lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	rsq	adjr2	rss	cp	BIC
1	1	1	0	0	0	0	0	0	0	0.54	0.53	44.53	24.77	-43.26
1	1	0	0	0	0	1	0	0	0	0.31	0.30	66.42	67.92	-16.46
1	1	0	0	0	0	0	1	0	0	0.24	0.23	73.24	81.36	-9.92
1	1	0	1	0	0	0	0	0	0	0.24	0.22	73.61	82.09	-9.58
1	1	0	0	0	0	0	0	0	1	0.20	0.19	76.95	88.68	-6.60
1	1	0	0	0	0	0	0	1	0	0.12	0.10	84.99	104.52	0.05
1	1	0	0	0	1	0	0	0	0	0.07	0.05	89.62	113.65	3.61
1	1	0	0	1	0	0	0	0	0	0.05	0.04	91.29	116.94	4.84
2	1	1	1	0	0	0	0	0	0	0.61	0.60	37.09	12.11	-51.30
2	1	1	0	0	1	0	0	0	0	0.58	0.57	39.99	17.83	-46.25
2	1	1	0	0	0	1	0	0	0	0.56	0.55	42.31	22.40	-42.47
2	1	1	0	0	0	0	0	0	1	0.55	0.53	43.42	24.59	-40.74
2	1	1	0	0	0	0	0	1	0	0.54	0.52	44.42	26.56	-39.21
2	1	1	0	0	0	0	1	0	0	0.54	0.52	44.47	26.65	-39.14
2	1	1	0	1	0	0	0	0	0	0.54	0.52	44.50	26.70	-39.10
2	1	0	1	0	0	1	0	0	0	0.46	0.45	51.71	40.93	-29.03
2	1	0	0	0	1	1	0	0	0	0.43	0.41	55.04	47.49	-24.85
3	1	1	1	0	0	1	0	0	0	0.64	0.62	34.91	9.80	-51.16
3	1	1	1	0	0	0	0	0	1	0.63	0.61	35.43	10.84	-50.15
3	1	1	0	0	1	1	0	0	0	0.63	0.61	35.97	11.90	-49.15
3	1	1	1	0	1	0	0	0	0	0.63	0.61	36.02	11.99	-49.06
3	1	1	1	0	0	0	0	1	0	0.62	0.60	36.66	13.25	-47.88
3	1	1	1	1	0	0	0	0	0	0.62	0.60	36.82	13.57	-47.59
3	1	1	1	0	0	0	1	0	0	0.61	0.60	37.09	14.10	-47.09
3	1	1	0	0	1	0	0	0	1	0.60	0.58	38.55	16.97	-44.52
3	1	1	0	1	1	0	0	0	0	0.59	0.57	39.80	19.45	-42.37
4	1	1	1	0	1	1	0	0	0	0.66	0.64	32.81	7.68	-51.09
4	1	1	1	0	0	1	0	0	1	0.65	0.62	34.07	10.16	-48.57
4	1	1	1	0	1	0	0	0	1	0.64	0.62	34.26	10.52	-48.21
4	1	1	1	0	0	1	1	0	0	0.64	0.62	34.27	10.55	-48.18
4	1	1	1	0	0	1	0	1	0	0.64	0.62	34.61	11.21	-47.53
4	1	1	1	1	0	1	0	0	0	0.64	0.62	34.71	11.42	-47.33
4	1	1	1	0	0	0	1	0	1	0.64	0.62	34.75	11.49	-47.26
4	1	1	1	1	0	0	0	0	1	0.64	0.62	34.82	11.62	-47.13
4	1	1	1	0	0	0	0	1	1	0.63	0.61	35.33	12.63	-46.15
5	1	1	1	0	1	1	0	0	1	0.67	0.64	32.07	8.21	-48.43
5	1	1	1	0	0	1	1	0	1	0.67	0.64	32.22	8.50	-48.12
5	1	1	1	1	1	1	0	0	0	0.66	0.64	32.28	8.63	-47.98
5	1	1	1	0	1	1	1	0	0	0.66	0.64	32.32	8.71	-47.90
5	1	1	1	0	1	1	0	1	0	0.66	0.63	32.64	9.33	-47.25
5	1	1	1	1	1	0	0	0	1	0.66	0.63	33.19	10.42	-46.13
5	1	1	1	1	0	1	0	0	1	0.65	0.62	33.64	11.31	-45.22
5	1	1	1	0	0	1	1	1	0	0.65	0.62	33.65	11.32	-45.21
5	1	1	1	0	1	0	1	0	1	0.65	0.62	33.84	11.69	-44.84
6	1	1	1	0	1	1	1	0	1	0.68	0.65	30.54	7.19	-47.50
6	1	1	1	1	1	1	0	0	1	0.68	0.64	31.20	8.49	-46.08
6	1	1	1	1	0	1	1	0	1	0.67	0.64	31.57	9.23	-45.27
6	1	1	1	1	1	1	0	1	0	0.67	0.64	31.81	9.69	-44.78
6	1	1	1	1	1	1	1	0	0	0.67	0.64	31.81	9.70	-44.76
6	1	1	1	0	1	1	1	1	0	0.67	0.64	31.91	9.90	-44.55
6	1	1	1	0	1	1	0	1	1	0.67	0.63	32.00	10.08	-44.36
6	1	1	1	0	0	1	1	1	1	0.67	0.63	32.16	10.38	-44.04
6	1	1	1	1	1	0	1	0	1	0.66	0.63	32.60	11.26	-43.12
7	1	1	1	1	1	1	1	0	1	0.69	0.66	29.44	7.02	-45.76
7	1	1	1	0	1	1	1	1	1	0.68	0.65	30.41	8.95	-43.57
7	1	1	1	1	1	1	1	1	0	0.68	0.64	30.96	10.02	-42.38
7	1	1	1	1	1	1	0	1	1	0.68	0.64	31.19	10.49	-41.87
7	1	1	1	1	0	1	1	1	1	0.67	0.63	31.57	11.23	-41.07
7	1	1	1	1	1	0	1	1	1	0.66	0.62	32.52	13.10	-39.09
7	1	1	0	1	1	1	1	1	1	0.65	0.61	33.27	14.57	-37.57
7	1	0	1	1	1	1	1	1	1	0.54	0.49	44.04	35.80	-18.77
8	1	1	1	1	1	1	1	1	1	0.69	0.65	29.43	9.00	-41.58

104225015 Lin Yuan-Hsin

Output

```
> beta_lse
                 lpsa
           2.46493292
lcavol     0.67952814
lweight    0.26305307
age       -0.14146483
lbph       0.21014656
svi        0.30520060
lcp       -0.28849277
gleason   -0.02130504
pgg45      0.26695576
> mean((y0-Y_hat)^2)
[1] 0.521274
```

```
> beta_Ridge
                   lpsa
          2.4523450851
lcavol    0.4210001259
lweight   0.2387919865
age      -0.0480240287
lbph      0.1623195446
svi       0.2271296444
lcp      -0.0000998623
gleason   0.0410749002
pgg45     0.1324521648
df        5.0002277992
> mean((y0-Y_hat_Ridge)^2)
[1] 0.4915854
```

```
> beta_LASSO1
                   [,1]
intercept 2.4682758821
lcavol    0.5299019817
svi       0.0881830394
lweight   0.1647164765
pgg45     0.0001893617
lbph      0.0004219772
gleason   0.0000000000
age       0.0000000000
lcp       0.0000000000
> mean((y0-Y_hat_LASSO1)^2)
[1] 0.483242
```

```
> beta_LASSO2
                  [,1]
intercept 2.468194388
lcavol    0.531960853
svi       0.094569801
lweight   0.168769508
pgg45     0.002774607
lbph      0.006182986
gleason   0.000000000
age       0.000000000
lcp       0.000000000
> mean((y0-Y_hat_LASSO2)^2)
[1] 0.4789707
```

```
> beta_best
             lpsa
        2.4773573
lcavol  0.7397137
lweight 0.3163282
> mean((y0-Y_hat_best)^2)
[1] 0.4924823
```

```
> beta_LASSO
9 x 1 sparse Matrix of class "dgCMatrix"
                 s0
        2.468359783
lcavol  0.533054114
lweight 0.169735954
age     .
lbph    0.002973318
svi     0.094376956
lcp     .
gleason .
pgg45   .
> mean((y0-Y_hat_LASSO.)^2)
[1] 0.4781344
```

104225015 Lin Yuan-Hsin

R-Code

```
rm(list=ls(all=TRUE))
library(leaps)
library(psych)
library(ElemStatLearn)
library(Matrix)
library(foreach)
library(glmnet)
data(prostate)


### constract data
attach(prostate)
lcavol=(lcavol-mean(lcavol))/sd(lcavol)
lweight=(lweight-mean(lweight))/sd(lweight)
age=(age-mean(age))/sd(age)
lbph=(lbph-mean(lbph))/sd(lbph)
svi=(svi-mean(svi))/sd(svi)
lcp=(lcp-mean(lcp))/sd(lcp)
gleason=(gleason-mean(gleason))/sd(gleason)
pgg45=(pgg45-mean(pgg45))/sd(pgg45)
data=cbind(lpsa,lcavol,lweight,age,lbph,svi,lcp,gleason,pgg45,train)


###linear model in the prostate cancer data
X=subset(data,select=c(lcavol:pgg45),train==TRUE)
Y=subset(data,select=c(lpsa),train==TRUE)
model=lm(Y~X)
beta_LSE=model$coefficients[-1]
```

```
##### Figure 3.8 (Ridge) #####
### Ridge
# Centered inputs
X=subset(data,select=c(lcavol:pgg45),train==TRUE)
X=X-matrix(1,nrow(X),1)%*%colMeans(X)
# find beta and degree of freedom
f_beta.df_Ridge=function(lambda){
   beta=solve(t(X)%*%X+lambda*diag(nrow(t(X)%*%X)))%*%t(X)%*%Y
   df=tr(X%*%solve(t(X)%*%X+lambda*diag(nrow(t(X)%*%X)))%*%t(X))
   return(rbind(beta,df))
}

beta.df_Ridge=c()
for(i in seq(0,10^5,5)){
   beta.df_Ridge=cbind(beta.df_Ridge,f_beta.df_Ridge(i))
}

#sort data
beta.df_Ridge=beta.df_Ridge[,order(beta.df_Ridge["df",])]

# draw coefficients
plot(c(0,9),c(-0.3,0.7),type="n",xlab="",ylab="",axes=FALSE,main="Profiles of
ridge coefficients")
axis(1,at=c(seq(0,8,2)),mgp=c(3,0.2,0),tcl=-0.15)
axis(2,las=1,font.axis=1,at=c(seq(-0.2,0.6,0.2)),mgp=c(3,0.2,0),tcl=-0.15)
box()
```

```
for(i in 1:8){

points(beta.df_Ridge[9,],beta.df_Ridge[i,],type="o",pch=16,col="blue",xlab="",yla
b="",lwd=0.5)
}
abline(h=0,col="black",lty=2)
abline(v=5,col="red",lty=2)
mtext(side=1,line=2,expression(df(lambda)),cex=1.2)
mtext(side=2,line=2,las=3,"Coefficients",cex=1.2)
axis(1,at=5,mgp=c(3,0.25,0),tcl=-0.3,col="red")
text(x=8.5,y=beta.df_Ridge["lcavol",ncol(beta.df_Ridge)],label="lcavol")
text(x=8.5,y=beta.df_Ridge["svi",ncol(beta.df_Ridge)],label="svi")
text(x=8.5,y=beta.df_Ridge["lweight",ncol(beta.df_Ridge)]+0.01,label="lweight")
text(x=8.5,y=beta.df_Ridge["pgg45",ncol(beta.df_Ridge)]-0.025,label="pgg45")
text(x=8.5,y=beta.df_Ridge["lbph",ncol(beta.df_Ridge)],label="lbph")
text(x=8.7,y=beta.df_Ridge["gleason",ncol(beta.df_Ridge)],label="gleason")
text(x=8.5,y=beta.df_Ridge["age",ncol(beta.df_Ridge)],label="age")
text(x=8.5,y=beta.df_Ridge["lcp",ncol(beta.df_Ridge)],label="lcp")

##### Figure 3.10 (Lasso) #####
### Lasso
# find beta
fit=glmnet(X,Y,family="gaussian")
s=colSums(abs(fit$beta))/sum(abs(beta_LSE))
beta0_LASSO=cbind(s=s,beta=coef.glmnet(fit)["(Intercept)",])
beta1_LASSO=cbind(s=s,beta=fit$beta["lcavol",])
beta2_LASSO=cbind(s,beta=fit$beta["svi",])
```

```
beta3_LASSO=cbind(s,beta=fit$beta["lweight",])
beta4_LASSO=cbind(s,beta=fit$beta["pgg45",])
beta5_LASSO=cbind(s,beta=fit$beta["lbph",])
beta6_LASSO=cbind(s,beta=fit$beta["gleason",])
beta7_LASSO=cbind(s,beta=fit$beta["age",])
beta8_LASSO=cbind(s,beta=fit$beta["lcp",])


# delect 0 for good plot
for(i in 2:8){
    eval(parse(text = paste('beta',i,'_LASSO=beta',i,'_LASSO[-1:-
(length(which(beta',i,'_LASSO[,2]==0))-1),]',sep='')))
}


# draw coefficients
plot(c(0,1.1),c(-0.3,0.7),type="n",xlab="",ylab="",axes=FALSE,main="Profiles of
lasso coefficients")
axis(1,at=c(seq(0,1,0.2)),mgp=c(3,0.2,0),tcl=-0.15)
axis(2,las=1,font.axis=1,at=c(seq(-0.2,0.6,0.2)),mgp=c(3,0.2,0),tcl=-0.15)
box()
for(i in 1:8){
    eval(parse(text =
paste('points(beta',i,'_LASSO[,1],beta',i,'_LASSO[,2],type="o",pch=16,col="blue",
xlab="",ylab="",lwd=2)',sep='')))
}
abline(h=0,col="black",lty=2)
abline(v=0.36,col="red",lty=2)
mtext(side=1,line=2,"Shrinkage Factor s",cex=1.2)
```

```
mtext(side=2,line=2,las=3,"Coefficients",cex=1.2)
axis(1,at=0.36,mgp=c(3,1,0),tcl=-0.8,col="red")
text(x=1.05,y=beta1_LASSO[nrow(beta1_LASSO),"beta"],label="lcavol")
text(x=1.05,y=beta2_LASSO[nrow(beta2_LASSO),"beta"],label="svi")
text(x=1.05,y=beta3_LASSO[nrow(beta3_LASSO),"beta"]+0.01,label="lweight")
text(x=1.05,y=beta4_LASSO[nrow(beta4_LASSO),"beta"]-0.02,label="pgg45")
text(x=1.05,y=beta5_LASSO[nrow(beta5_LASSO),"beta"],label="lbph")
text(x=1.05,y=beta6_LASSO[nrow(beta6_LASSO),"beta"],label="gleason")
text(x=1.05,y=beta7_LASSO[nrow(beta7_LASSO),"beta"],label="age")
text(x=1.05,y=beta8_LASSO[nrow(beta8_LASSO),"beta"],label="lcp")

##### Table 3.3 #####
### mean prediction error

# LS #
x0=subset(data,train==FALSE,select=c(lcavol:pgg45))
y0=subset(data,train==FALSE,select=c(lpsa))
X=cbind(1,X)
x0=cbind(1,x0)
beta_lse=solve(t(X)%*%X)%*%t(X)%*%Y
Y_hat=x0%*%beta_lse
mean((y0-Y_hat)^2)

# best subset selection #
out.all=regsubsets(X,Y, nbest=9, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2, rss=s.all$rss,
```

```
cp=s.all$cp,BIC=s.all$bic),2)

X_new=subset(data,select=c(lcavol,lweight),train==TRUE)

X_new=cbind(1,X_new)

beta_best=solve(t(X_new)%*%X_new)%*%t(X_new)%*%Y

x0_new=subset(data,train==FALSE,select=c(lcavol,lweight))

x0_new=cbind(1,x0_new)

Y_hat_best=x0_new%*%beta_best

mean((y0-Y_hat_best)^2)


# Ridge #

beta_Ridge=rbind(mean(Y),f_beta.df_Ridge(23.9955))

beta_Ridge.=beta_Ridge[-nrow(beta_Ridge),]

Y_hat_Ridge=x0%*%beta_Ridge.

mean((y0-Y_hat_Ridge)^2)


# Lasso #

# interpolation method

beta_LASSO1=c()

for(i in 0:8){

   eval(parse(text = paste('

                                        n=which.min(abs(beta',i,'_LASSO[,"s"]-0.36))

                                        s0=beta',i,'_LASSO[n,"s"]

                                        s1=beta',i,'_LASSO[n+1,"s"]

                                        b0=beta',i,'_LASSO[n,"beta"]

                                        b1=beta',i,'_LASSO[n+1,"beta"]

                                        beta_LASSO1=rbind(beta_LASSO1,(b0*s1-
b1*s0+0.36*(b1-b0))/(s1-s0))',sep='')))
```

```
}
# Arithmetic mean
beta_LASSO2=c()
n=which.min(abs(beta0_LASSO[,"s"]-0.36))
for(i in 0:8){
  eval(parse(text =
paste('beta_LASSO2=rbind(beta_LASSO2,(beta',i,'_LASSO[n,"beta"]+beta',i,'_LA
SSO[n+1,"beta"])/2)',sep='')))
}
# directly from packages
lasso.final = glmnet(X, Y, family="gaussian", alpha = 1, lambda = 0.2083)
beta_LASSO=rbind(lasso.final$a0, lasso.final$beta)
length(beta_LASSO)
# predict
rownames(beta_LASSO1)=c("intercept","lcavol","svi","lweight","pgg45","lbph","
gleason","age","lcp")
rownames(beta_LASSO2)=c("intercept","lcavol","svi","lweight","pgg45","lbph","
gleason","age","lcp")
x0_new.l=subset(data,train==FALSE,select=c(lcavol,svi,lweight,lbph))
x0_new.l=cbind(1,x0_new.l)
x0_new.ll=subset(data,train==FALSE,select=c(lcavol,lweight,lbph,svi))
x0_new.ll=cbind(1,x0_new.ll)


beta_LASSO.1=beta_LASSO1[-(7:9),]
beta_LASSO.1=beta_LASSO.1[-5]
beta_LASSO.2=beta_LASSO2[-(7:9),]
beta_LASSO.2=beta_LASSO.2[-5]
```

```
beta_LASSO.=beta_LASSO[-(7:9),]
beta_LASSO.=beta_LASSO.[-4]


Y_hat_LASSO1=x0_new.l%*%beta_LASSO.1
Y_hat_LASSO2=x0_new.l%*%beta_LASSO.2
Y_hat_LASSO.=x0_new.ll%*%beta_LASSO.


mean((y0-Y_hat_LASSO1)^2)
mean((y0-Y_hat_LASSO2)^2)
mean((y0-Y_hat_LASSO.)^2)


###output
beta_lse
mean((y0-Y_hat)^2)
beta_best
mean((y0-Y_hat_best)^2)
beta_Ridge
mean((y0-Y_hat_Ridge)^2)
beta_LASSO1
mean((y0-Y_hat_LASSO1)^2)
beta_LASSO2
mean((y0-Y_hat_LASSO2)^2)
beta_LASSO
mean((y0-Y_hat_LASSO.)^2)
```