```
#You will often need packages and may need to install packages
#the first time you use a given computer.

install.packages("glmnet"); install.packages("leaps")
source("http://lagrange.math.siu.edu/Olive/slpack.txt")

library(glmnet); library(MASS); library(mgcv)
library(leaps); library(survival)

source("http://lagrange.math.siu.edu/Olive/sldata.txt")
x<-ceriodx;y<-ceriody
tdata <- as.data.frame(cbind(x,y))
out <- glm(y˜., family=poisson, data=tdata)
x<-as.matrix(x); d<-dim(x)[2]+1 #d=no. of predictors
ESP <- predict(out)
prpiplot2(ESP=ESP,y=y,d=4,B=1000) #use for lots of PR models
prpiplot(x,y,B=1000) #full PR model
$cov
[1] 0.9714286
#ceriodplot

par(mfrow=c(1,2))
prpiplot2(ESP=ESP,y=y,d=4,B=1000)
$cov
[1] 0.9714286

prpiplot2(ESP=ESP,y=y,d=4,B=1000000)
$cov
[1] 0.9857143
par(mfrow=c(1,1))

banknote<-matrix(scan(),nrow=200,ncol=7,byrow=T)
#copied from the Cook and Weisberg ARC software
#or get from Weisberg's alr3 R package
#diagonal causes perfect separation and problems for BE
bank<-as.data.frame(banknote[,-7])
names(bank)<-c("status","length","left","right","bottom","top")
outf<-glm(status˜.,family=binomial,data=bank)
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -475.5814   404.9133  -1.175  0.24018
length         0.3753     1.4175   0.265  0.79120
left          -1.5305     4.0802  -0.375  0.70758
right          3.6275     3.2848   1.104  0.26946
bottom         5.2386     1.8723   2.798  0.00514 **
top            6.9957     2.1811   3.207  0.00134 **
back <- step(outf)
```

```
bank<-as.data.frame(banknote)
names(bank)<-c("status","length","left","right","bottom","top","diagonal")
outf<-glm(status~.,family=binomial,data=bank)
back <- step(outf)

y<-bank[,1]
out2<-glm(y~right+bottom,family=binomial,data=bank)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -417.8617   104.7356  -3.990 6.62e-05 ***
right          3.0662     0.8036   3.816 0.000136 ***
bottom         2.0806     0.3251   6.400 1.56e-10 ***

par(mfrow=c(1,2))
mv<-0*1:200 + 1
ESP<-predict(out2)
lrpiplot(ESP=ESP,y=y,mv=mv,d=3)
$cov
[1] 0.995
outg<-gam(status~s(right)+s(bottom),family=binomial,data=bank)
EAP <- predict(outg)
lrpiplot(ESP=EAP,y=y,mv=mv,d=3)
$cov
[1] 1.000

#overdispersion
y <- species[,1]
x<-log(species[,-1])
x<-x[,c(1,2,4,6)]
prplot(x,y)
tdata <- as.data.frame(cbind(x,y))
out <- glm(y~., family=poisson, data=tdata)
ESP <- predict(out)
d<-dim(x)[2]+1
prpiplot2(ESP=ESP,y=y,d=d) #lots of PR models
prpiplot(x,y) #full PR model
$cov
[1] 0.8965517 #undercoverage

brpisim(n=100,p=4,k=1,m=40,nruns=5000,psi=0.0) #takes about 15 minutes
$mndRL     #mean d for lasso, relaxed lasso
[1] 3.0336
$mndd      # mean d for Olive and Hawkins forward selection
[1] 2.3354
$mnddbe   #mean d for backward elimination
[1] 2.312
$int
[1] 0
```

```
$b
[1] 1 0 0
$fullpicov
[1] 0.9586
$fullpimenlen
[1] 9.4774
$gampicov
[1] 0.9562
$gampimenlen
[1] 9.4732
$laspicov
[1] 0.9752
$laspimenlen
[1] 10.4292
$RLpicov
[1] 0.9738
$RLpimenlen
[1] 10.348
$ohfspicov
[1] 0.972
$ohfspimnlen
[1] 10.1166
$vspicov
[1] 0.9596
$vspimnlen
[1] 9.4744

prpisim2(n=100,p=4,k=1,nruns=5000,psi=0,B=1000,alpha=0.05)
#takes a while with 5000 runs, change n, p, k, and psi as in the paper.
$int
[1] 1
$b
[1] 1 0 0
$fullpicov
[1] 0.9676
$fullpimenlen
[1] 6.6686
$gampicov
[1] 0.964
$gampimenlen
[1] 6.6626
$laspicov
[1] 0.9792
$laspimenlen
[1] 7.3202
$RLpicov
[1] 0.9794
```

```
$RLpimenlen
[1] 7.2414
$ohfspicov
[1] 0.9784
$ohfspimnlen
[1] 7.111
$vspicov
[1] 0.9726
$vspimnlen
[1] 7.093

#often need n large or MLE won't converge
wpisim(n=1000,p=10,k=4,nruns=5000,psi=1/sqrt(10),gam=4)
$int
(Intercept)
0.003190349
$beta
 [1] 1 1 1 1 0 0 0 0 0 0
$fullpicov
[1] 0.9526
$fullpimenlen
[1] 1.05116


wpisim(n=1000,p=10,k=4,nruns=5000,psi=0.9,gam=4)
$int
(Intercept)
  0.0169485
$b
 [1] 1 1 1 1 0 0 0 0 0 0
$fullpicov
[1] 0.9476
$fullpimenlen
[1] 1.0554
```