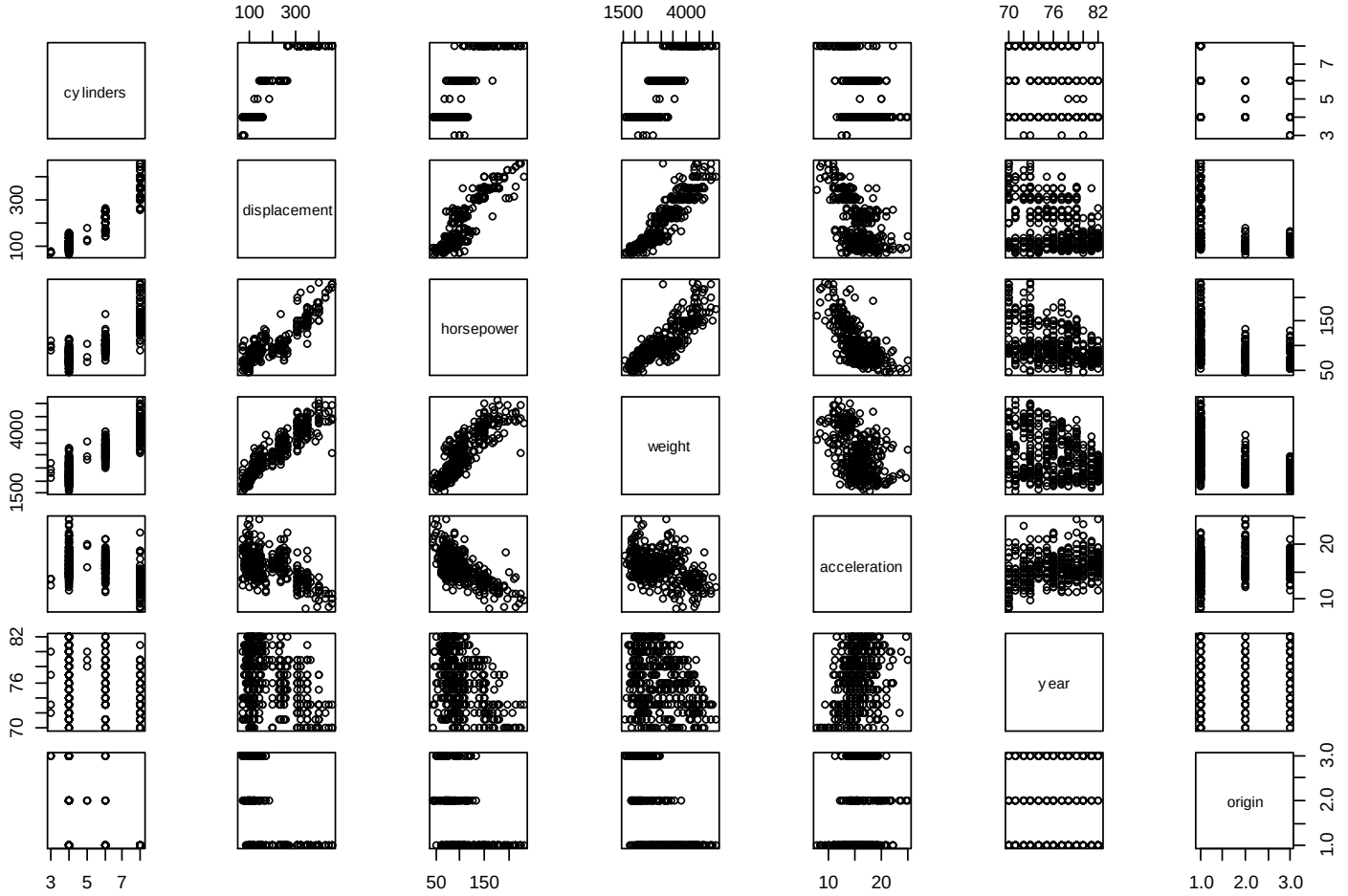
Outline

- New slot in assignments. “Backup for redone assignments”
  - When you redo an assignment that I’ve requested you can put it in here and send me an email that you’ve done so. (michael.luvalle@rutgers.edu)
- High dimensional plots
  - Adding time to guided tours
  - Pairs plots
  - Using eigenvectors
- Pure exploratory data analysis mystery.pck is a dump file in R, on the class resources file figure out what is hidden in the data set send me a picture and tell me how you found it.

Adding a time indicator in the animate function

```
animate.time.path<-function(x,index.f=holes)
{
    n1<-length(x[,1])
    edges<-matrix(c(1:(n1-1),2:n1),ncol=2)
    m1<-
  animate(x,tour_path=guided_tour(index.f),dis
  play_xy(axes = "bottomleft", edges = edges))
}
```

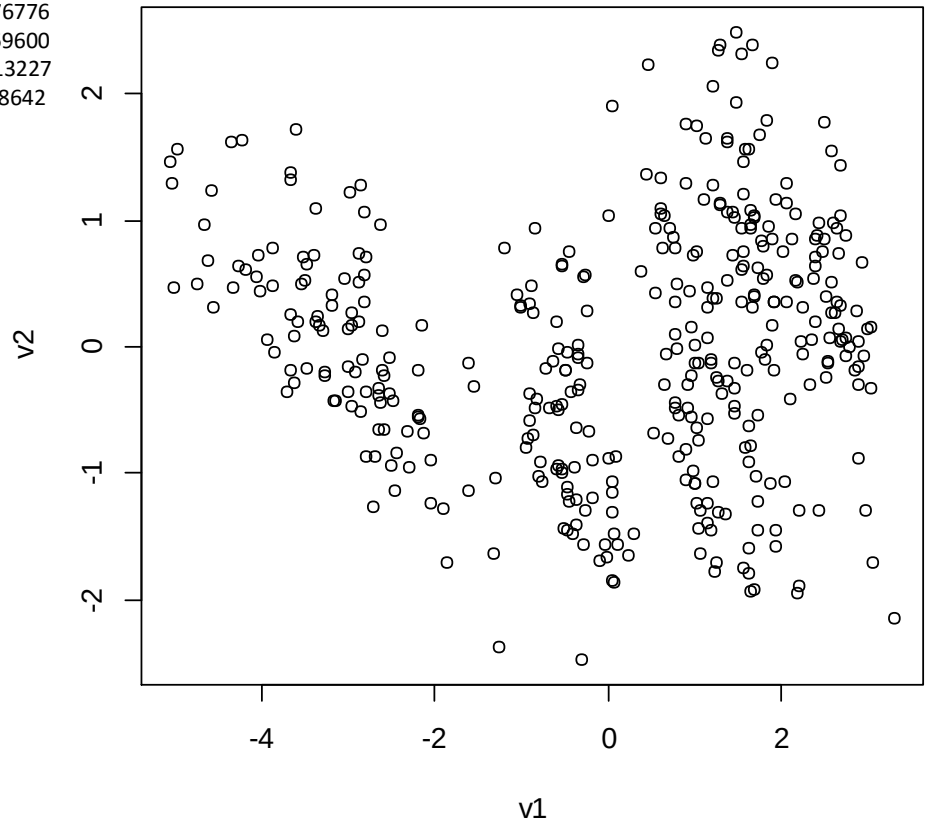
Pairs plots



pairs (x.auto)

Using eigen vectors to project data

```
> estr<-eigen(cor(x.auto))
> estr$vectors
          [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
[1,] -0.4381181 -0.1202135 -0.02552952 -0.24087119  0.70016012 -0.19483078
[2,] -0.4533656 -0.1074428 -0.02219632 -0.15933085  0.17628654 -0.06371028
[3,] -0.4375575  0.1424399 -0.17850576 -0.12458174 -0.57939323 -0.61360775
[4,] -0.4319464 -0.2027754  0.01259408 -0.33522884 -0.34853612  0.70776776
[5,]  0.2977076 -0.4821858  0.55906361 -0.52733279 -0.12091625 -0.27069600
[6,]  0.2147328 -0.6424978 -0.73184478 -0.02888161 -0.02423084 -0.06313227
[7,]  0.2995629  0.5170022 -0.34450506 -0.71399002  0.07947985  0.05718642
           [,7]
[1,] -0.454694715
[2,]  0.849647407
[3,] -0.169288055
[4,] -0.193274082
[5,]  0.018385950
[6,]  0.009090993
[7,]  0.070128408
> v1<-x.auto%*%estr$vector[,1]
> v2<-x.auto%*%estr$vector[,2]
> plot(v1,v2)
> v1<-scale(x.auto)%*%estr$vector[,1]
> v2<-scale(x.auto)%*%estr$vector[,2]
> plot(v1,v2)
```



Alternative path for assignment 3

- Copy code
- In each ## comment on the code below it
- Run the code and compare the confidence intervals
- Return the confidence intervals, your comments, and the commented code
- Using your own code instead of mine is plus 20 extra credit

Lab part 1: What is the flow:

- Which is the “master program” controlling things?
- What do the sub programs do?
- Within those sub programs what do the lines do and why?

```
> leaps.then.press.plot2
function(xmat0,yvec,xpred,ncheck=5)
{
#
#input quadratic matrix with less than 30 columns eg. the result of x.auto2a<-matrix.2ndorder.make(xmat[,-7],F)
#also, no need for plotting, just pull out best, xpred is one of the row vectors from x.auto2a, but all terms with weight are divided by 2
#
      ##
      leaps.str<-leaps(xmat0,yvec)
      ##
      z1<-leaps.str$Cp-leaps.str$size
      ##
      o1<-order(z1)
      matwhich<-(leaps.str$which[o1,])[1:ncheck,]
      MPSEvec<-NULL
      ##
      for(i in 1:ncheck){
            ls.str0<-regpluspress(xmat0[,matwhich[i,]],yvec)
            ##
            parvec<-matwhich[i,]
            npar<-sum(parvec)
            ## (WHY npar+1)
            MPSE<-ls.str0$press/(length(yvec)-(npar+1))
            MPSEvec<-c(MPSEvec,MPSE)
      }
      ##
      I1<-(MPSEvec==min(MPSEvec))
      ##
      i<-c(1:ncheck)[I1]
      ##
      xmat.out<-xmat0[,matwhich[i,]]
      ##
      xpred.out<-xpred[matwhich[i,]]
      ##
      list(xmatout=xmat.out,yvec=yvec,xpredout=xpred.out)


}
```

```
> bootreg<-
function(xmat,yvec,xpred,nboot=10000,alpha=0.1)
{
      ##
      lstr0<-leaps.then.press.plot2(xmat,yvec,xpred)
      xmat0<-lstr0$xmatout
      yvec0<-lstr0$yvec
      xpred0<-lstr0$xpredout
      ##
      rprd.list<-regpred(xpred0,xmat0,yvec0)
      ypred0<-rprd.list$pred
      sdpred0<-rprd.list$sd
      df0<-rprd.list$df
      ##
      bootvec<-NULL
      nobs<-length(yvec0)
      for(i in 1:nboot){
if(floor(i/100)==(i/100)){
print(i)
}
            ##
            vboot<-sample(c(1:nobs),replace=T)
#### The terms below where xmat0[vboot,], should have been the original xmatrix (by the
          logic of the bootstrap, think about what is happening)
            xmatb<-xmat[vboot,]
            yvecb<-yvec[vboot]
            ##
            lstrb<-leaps.then.press.plot2(xmatb,yvecb,xpred)
            ##
            xmatb0<-lstrb$xmatout
            yvecb0<-lstrb$yvec
            xpredb0<-lstrb$xpredout
            ##
            rprd.list<-regpred(xpredb0,xmatb0,yvecb0)
            ypredb<-rprd.list$pred
            sdpredb<-rprd.list$sd
            dfb<-rprd.list$df
            ##
            bootvec<-c(bootvec,(ypredb-ypred0)/sdpredb)
      }
 ##
      lq<-quantile(bootvec,alpha/2)
      uq<-quantile(bootvec,1-alpha/2)
##
      LB<-ypred0-(sdpred0)*uq
      UB<-ypred0-(sdpred0)*lq
##
      NLB<-ypred0-(sdpred0)*qt(1-alpha/2,df0)
      NUB<-ypred0+(sdpred0)*qt(1-alpha/2,df0)
list(bootstrap.confidence.interval=c(LB,UB),normal.confidence.interval=c(NLB,NUB
          ))
}
```

```
> regpred<-
function(xpred,xmat,y){
#calculate regression
ls.str<-lsfit(xmat,y)
#calculate prediction
ypred<-sum(ls.str$coef*c(1,xpred))
#use ls.diag to extract covariance matrix
ycov<-ls.diag(ls.str)$cov.unscaled
#use ls.diag to extract std deviation
std.dev<-ls.diag(ls.str)$std.dev
#variance of data around line
v1<-std.dev^2
#variance of prediction
vpred<-v1*c(1,xpred)%*%ycov%*%c(1,xpred)
df=length(y)-length(diag(ycov))
list(pred=ypred,sd=sqrt(vpred),df=df)
}
```

Lab Part 2

- Work on assignments 3
- If done, download mystery.pck from resources, load it into R using the source command. Find out the interesting views of mystery.dat using the multidimensional viewing tools you have, copy them onto a page, and write it up. (this is assignment 4, due in one week).
- You can try using linear model tools as well! Try what you want, see if different variables can predict different other variables see what is interesting in mystery.dat.
- Consider all the plot tools you have learned