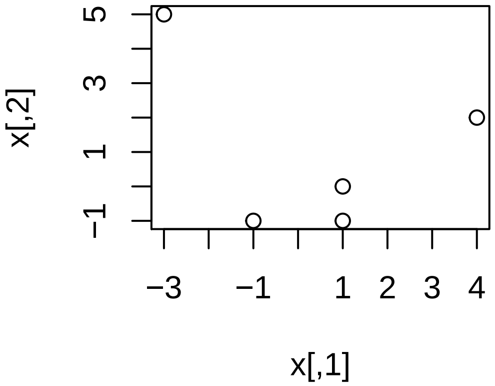
Assignment 2

Name Student no.

Q1

a) - e)

```
x <- matrix(c(4 ,1,-1,-3,1, 2,0,-1,5,-1), nrow=5)
plot(x)
```



```
var(x)
```

```
##       [,1]  [,2]
## [1,]  6.80 -2.25
## [2,] -2.25  6.50
```

```
cor(x)
```

```
##           [,1]      [,2]
## [1,]  1.000000 -0.338432
## [2,] -0.338432  1.000000
```

```
xs<- scale(x)

xs
```

```
##            [,1]       [,2]
## [1,]  1.3805370  0.3922323
## [2,]  0.2300895 -0.3922323
## [3,] -0.5368755 -0.7844645
## [4,] -1.3038405  1.5689291
## [5,]  0.2300895 -0.7844645
## attr(,"scaled:center")
## [1] 0.4 1.0
## attr(,"scaled:scale")
## [1] 2.607681 2.549510
```

```
t(xs)%*%(xs)/(5-1)
```

```
##           [,1]      [,2]
## [1,]  1.000000 -0.338432
## [2,] -0.338432  1.000000
```

```
var(xs)
```

```
##           [,1]      [,2]
## [1,]  1.000000 -0.338432
## [2,] -0.338432  1.000000
```

```
eigen(cor(x))
```

```
## eigen() decomposition
## $values
## [1] 1.338432 0.661568
##
## $vectors
##            [,1]       [,2]
## [1,] -0.7071068 -0.7071068
## [2,]  0.7071068 -0.7071068
```

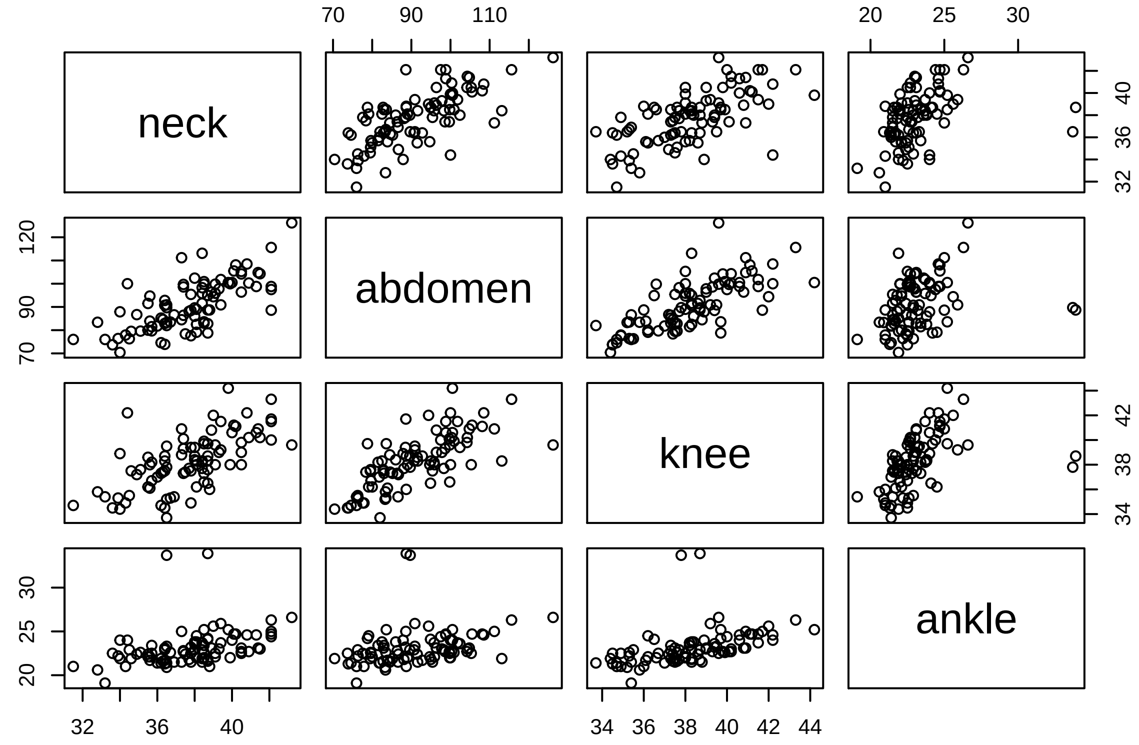
```
prcomp(x, scale=TRUE)
```

```
## Standard deviations (1, .., p=2):
## [1] 1.1569062 0.8133683
##
## Rotation (n x k) = (2 x 2):
##             PC1       PC2
## [1,] -0.7071068 0.7071068
## [2,]  0.7071068 0.7071068
```

Q2

a)

```
bfat <- read.table("data/bodyfat.txt", header=T)
bfat <- bfat[,c("neck","abdomen", "knee", "ankle")]
pairs(bfat)
```



```
bfat[bfat[,4]>30, ]
```

```
##    neck abdomen knee ankle
## 31 38.7    88.7 38.7  33.9
## 84 36.5    89.7 37.8  33.7
```

There are two outliers with extreme ankle values, but non extreme values on other variables. They are observations 31 and 84.

b)

```
screeplot <- function(p) {
  e <- p$sdev ^ 2
  e <- e / sum(e)
  plot(
```

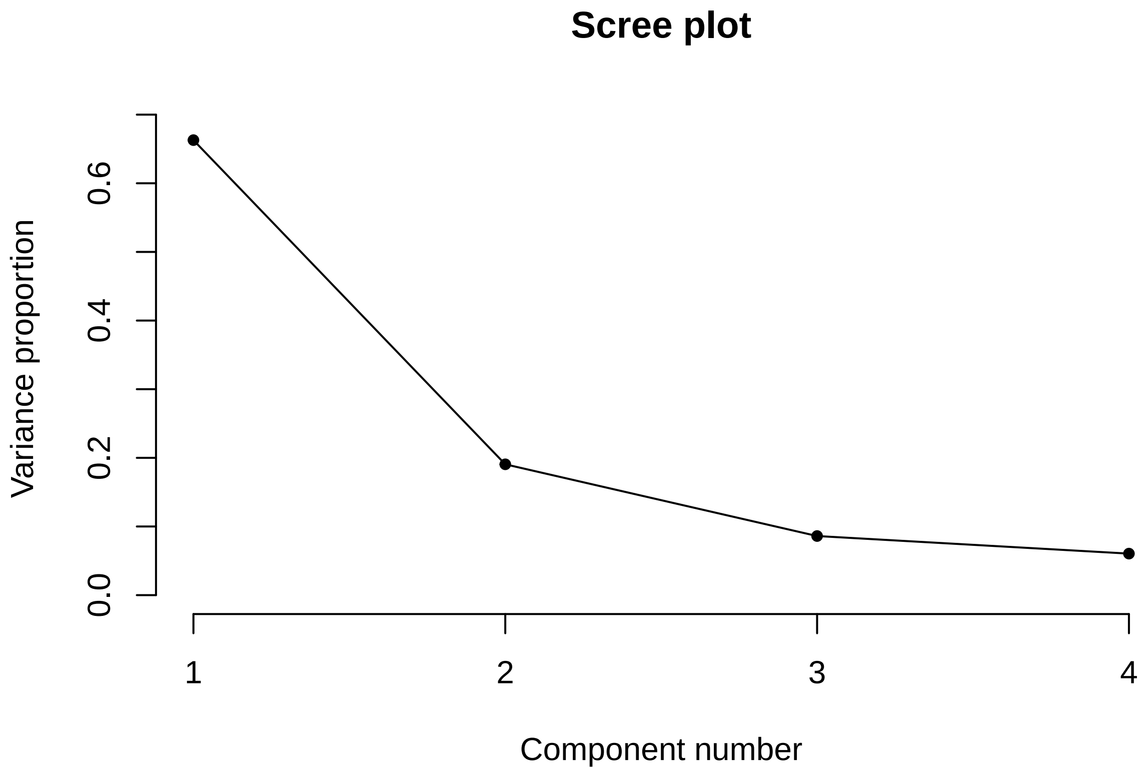
```r
    1:length(e),
    e,
    xlab = "Component number",
    pch = 20,
    ylab = "Variance proportion",
    main = "Scree plot",
    axes = F,
    ylim = c(0, max(e)*1.04)
  )
  lines(1:length(e), e)
  axis(1, at = 1:length(e))
  axis(2)
}

# solution
p <- prcomp(bfat, scale=TRUE)
p$rotation[,1:2]
```

```
##               PC1         PC2
## neck    0.5283837  0.21938447
## abdomen 0.5351101  0.35203936
## knee    0.5460990  0.05660109
## ankle   0.3691120 -0.90814925
```

```r
summary(p)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.6283 0.8731 0.58678 0.49183
## Proportion of Variance 0.6629 0.1906 0.08608 0.06047
## Cumulative Proportion  0.6629 0.8535 0.93953 1.00000
```

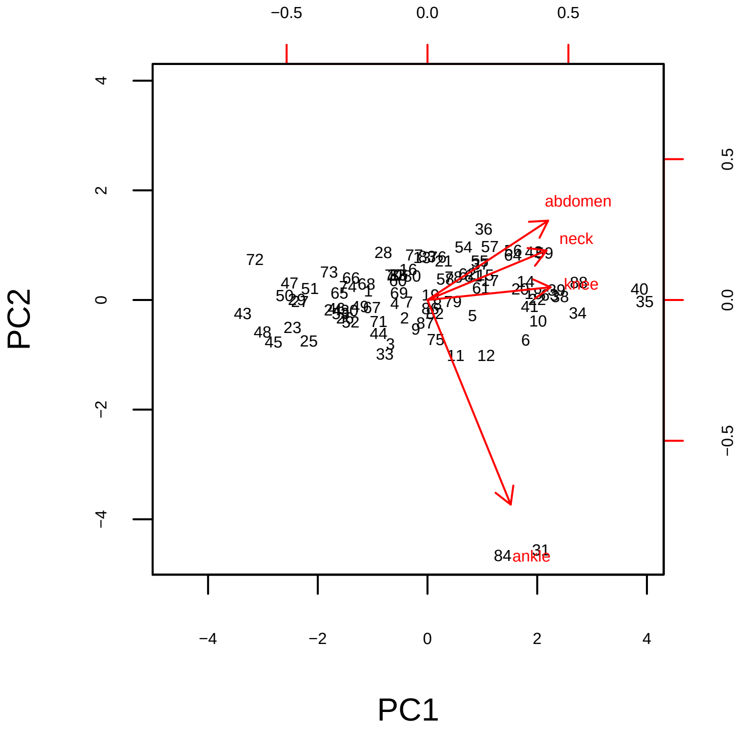
```r
screeplot(p)
```



66% of variability explaide by the 1st PC, 85% by the first 2 PCs and 94% by the first 3 PCs.

c)

```
biplot(p, scale=0, cex=c(.5,.5), cex.axis=.5)
```



The first component is a weighted average of the variables. It is an overall measure of size. The second component is a contrast of neck and abdomen with ankle. It is a measure of the difference between top size and ankle. The visible outliers are 84 and 31, with the big ankle values.

d)

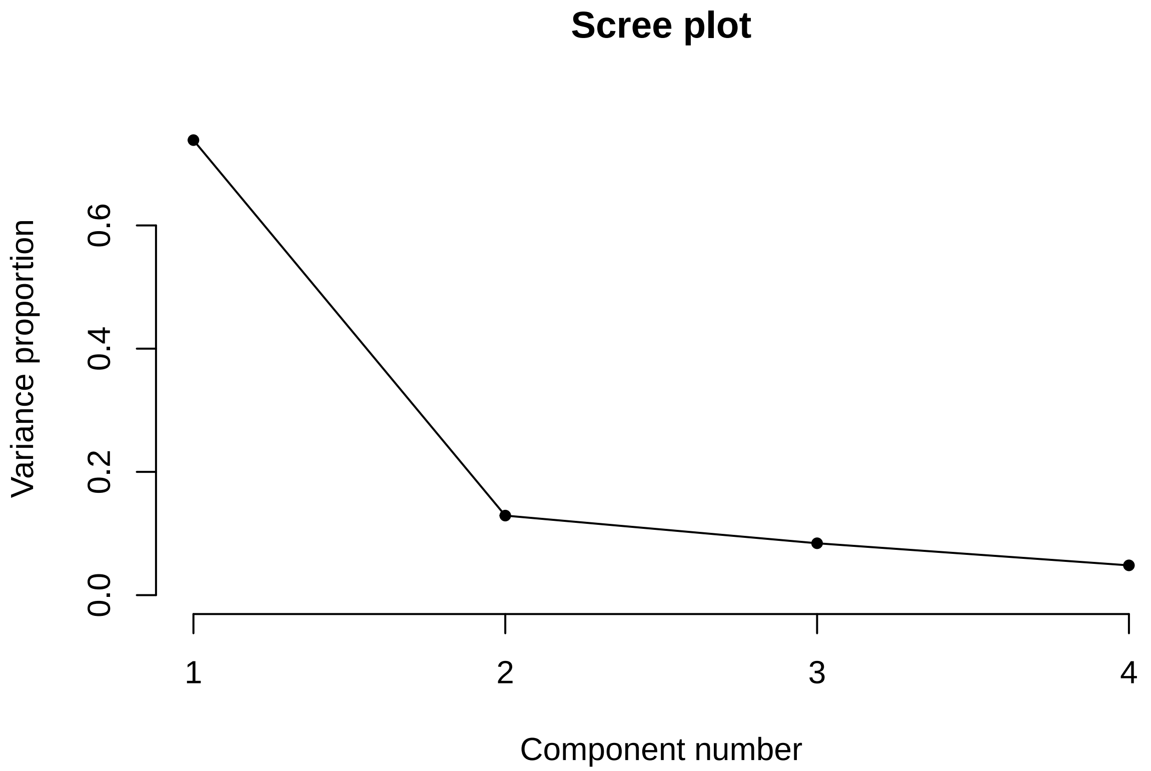
```r
p<- prcomp(bfat[-c(31,84),], scale=TRUE)
p$rotation[,1:2]
```

```
##               PC1        PC2
## neck    0.5002906  0.3230518
## abdomen 0.5005250  0.5460175
## knee    0.5251301 -0.1447403
## ankle   0.4726758 -0.7593106
```

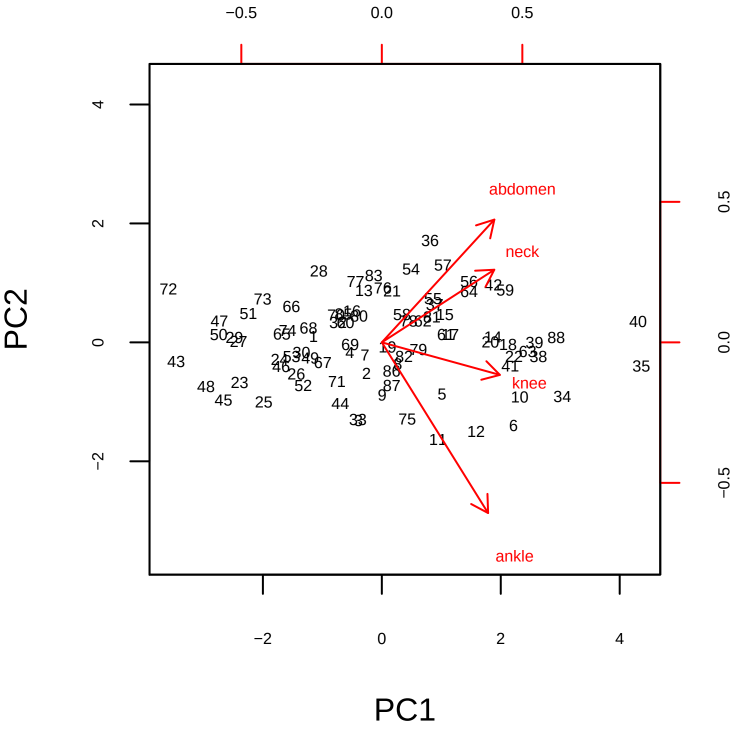
```r
summary(p)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7186 0.7186 0.58020 0.43962
## Proportion of Variance 0.7384 0.1291 0.08416 0.04832
## Cumulative Proportion  0.7384 0.8675 0.95168 1.00000
```

```r
screeplot(p)
```



```
biplot(p, scale=0, cex=c(.5,.5), cex.axis=.5)
```



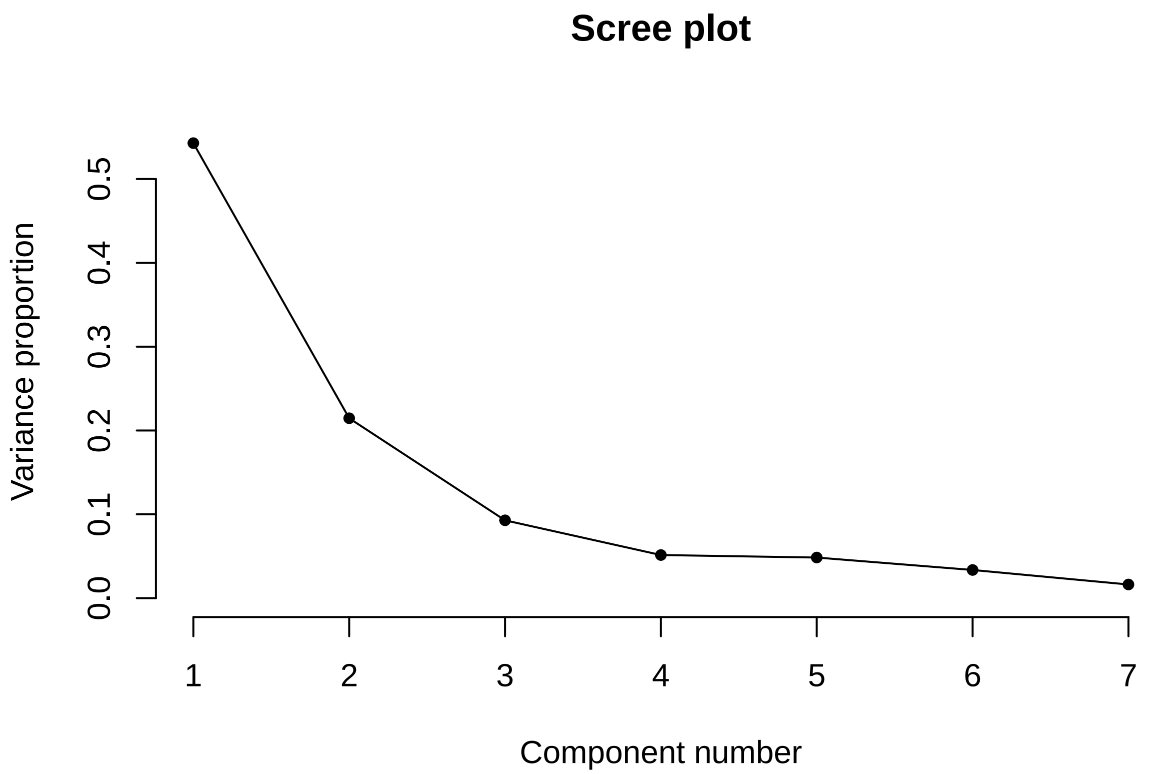
The first component is a weighted average of the variables. It is an overall measure of size. The second component is a contrast of neck and abdomen with knee and ankle. It is a measure of the difference between top size and lower size. The high weight people stick out on the first component, but are not that extreme.

Q3

```
# read in the correlation data as a vector

crimcorr <- matrix(c(
  1.000, 0.402, 0.396, 0.301, 0.305, 0.339, 0.340,
  0.402, 1.000, 0.618, 0.150, 0.135, 0.206, 0.183,
  0.396, 0.618, 1.000, 0.321, 0.289, 0.363, 0.345,
  0.301, 0.150, 0.321, 1.000, 0.846, 0.759, 0.661,
  0.305, 0.135, 0.289, 0.846, 1.000, 0.797, 0.800,
  0.339, 0.206, 0.363, 0.759, 0.797, 1.000, 0.736,
  0.340, 0.183, 0.345, 0.661, 0.800, 0.736, 1.000), nrow = 7, byrow = TRUE)

colnames(crimcorr)<- c("Head-L","Head-B","Face-B",
                       "L-Fing","L-Fore","L-Foot",
                       "Height")
V <- eigen(crimcorr)

V$values/sum(V$values)
```

```
## [1] 0.54278208 0.21461832 0.09282963 0.05143670 0.04845178 0.03360759
## [7] 0.01627391
```

```
# can hack it to use the screeplot above, else make your own.
V$sdev <- sqrt(V$values)
screeplot(V)
```



Proportion variance explained by the 1st PC is 0.54, first two is 0.76 etc.

First PC is a measure of overall size of the person. Second PC contrasts head measurements with the rest. Third PC is the head length etc.

Q4

a)

Regression, inference,n = 500, 1 response, 4 predictors. All predictors are quantitative except country which is categorical. Inflexible better for inference.

b)

classification, prediction,n = 20, 1 response (binary), 12 predictors. All described predictors are quantitative Inflexible better because so many predictors relative to n.

c)

Regression, inference,n = unknown, 1 response (quantitative), 2 predictors, birthweight quantitative and gender categorical.Inference. Inflexible better because to understand predictors response association

d)

Classification, prediction,n = 32, 1 response (categorical, 3 classes), 56 predictors, structure unknown. Inflexible becuase n so large relative to p.

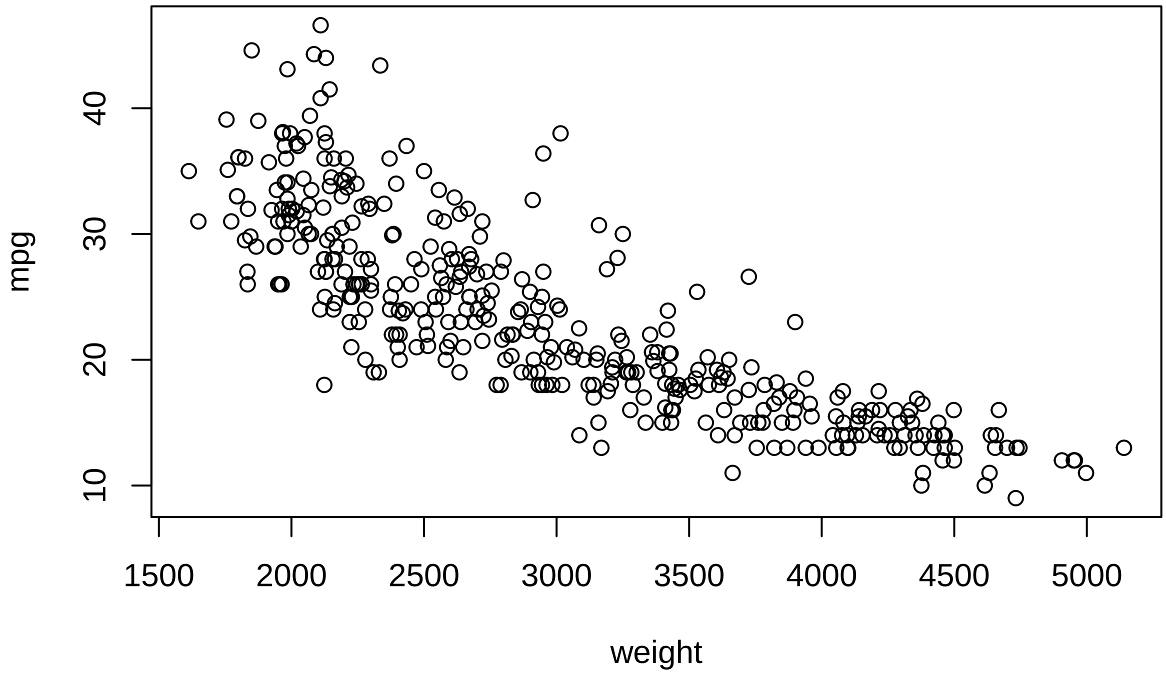
Q5

a)

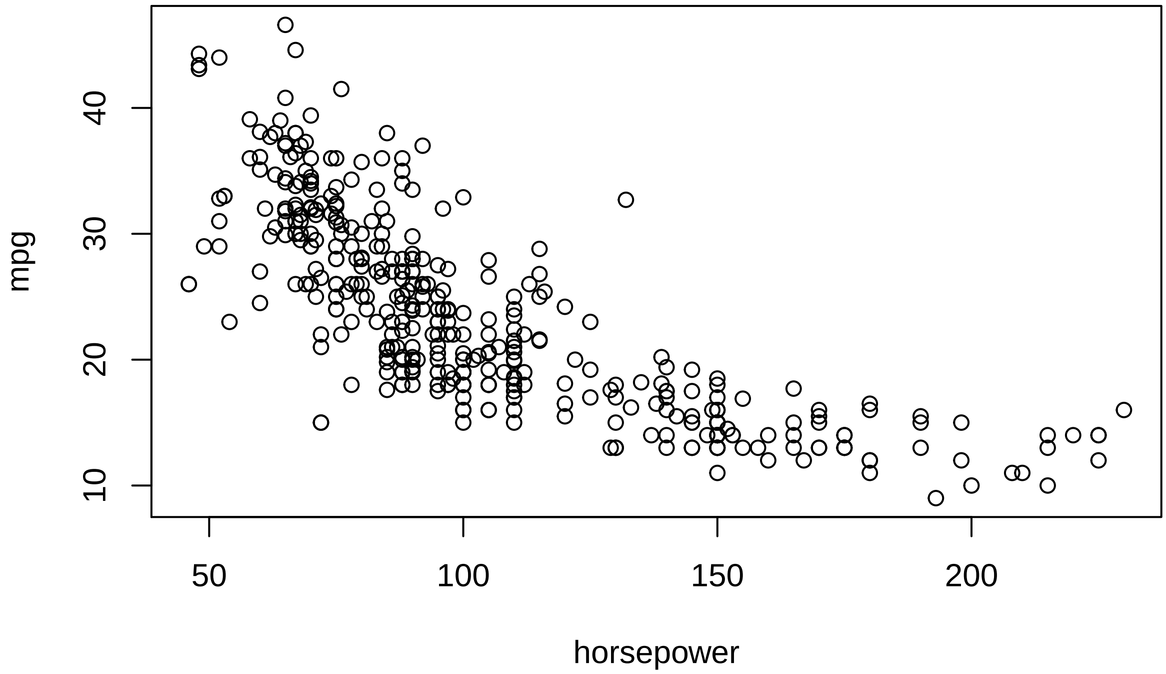
```r
# install.packages("ISLR") #home computer, first time only
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.4.3
```

```r
Auto <-Auto[complete.cases(Auto[,c(1,4,5)]),] # to remove NAs
plot(mpg ~ weight, data=Auto)
```



```r
plot(mpg ~ horsepower, data=Auto)
```



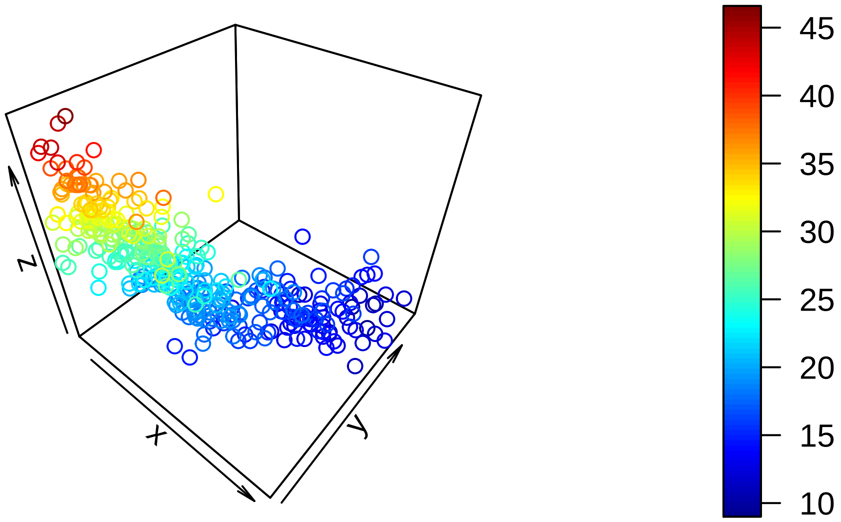
mpg goes down as weight goes up, plot shows curvature. mpg goes down as hp goes up, plot shows curvature.

b)

```
library(plot3D) # install package
```

```
## Warning: package 'plot3D' was built under R version 3.4.3
```

```
scatter3D(Auto$weight,Auto$horsepower,Auto$mpg)
```



```
library(plot3Drgl)
```

```
## Warning: package 'plot3Drgl' was built under R version 3.4.3
```

```
## Loading required package: rgl
```

```
scatter3Drgl(Auto$weight,Auto$horsepower,Auto$mpg)
```

plot shows that points lie on a surface, not a plane, so a linear fit is not appropriate

c)

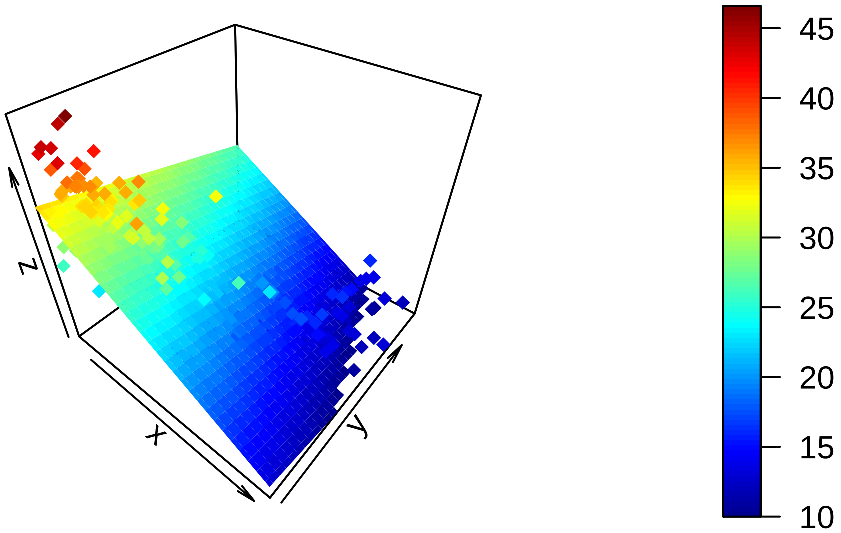
```
set.seed(123)
train <- sample(nrow(Auto), round(.8*nrow(Auto)))
AutoTrain <- Auto[train,]
AutoTest <- Auto[-train,]
```

```
f1 <- lm(mpg~weight+horsepower, data=AutoTrain)
summary(f1)
```

```
##
## Call:
## lm(formula = mpg ~ weight + horsepower, data = AutoTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4101  -2.7431  -0.4644   2.5079  16.0258
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.1626326  0.8950964   51.57  < 2e-16 ***
## weight      -0.0060579  0.0005553  -10.91  < 2e-16 ***
## horsepower  -0.0431712  0.0120932   -3.57 0.000414 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.339 on 311 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7068
## F-statistic: 378.3 on 2 and 311 DF,  p-value: < 2.2e-16
```

Both predictors are significant as p values are so small # d)

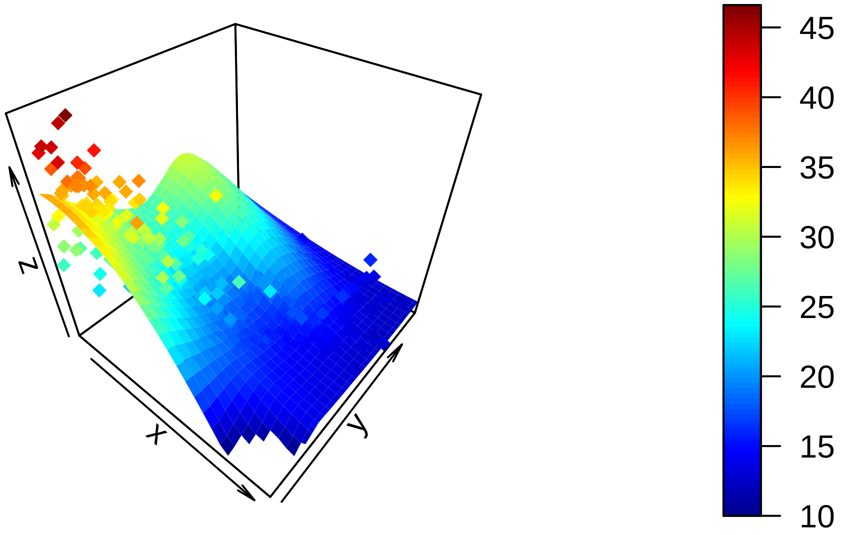
```
wt1 <- seq(1610 ,5140, length.out = 30)
hp1 <- seq(45, 230, length.out = 30)
pred <- predict(f1, expand.grid(weight=wt1, horsepower=hp1))
pred <- matrix(pred,30,30)
library(plot3D)
scatter3D(AutoTrain$weight,AutoTrain$horsepower,AutoTrain$mpg, pch = 18, surf = list(x = wt1, y = hp1, 
```



You can see for high values of z=mpg, points lie far away and mostly above from the fitted planes so the linear fit does not look appropriate.

e)

```
f2 <- loess(mpg~weight+horsepower, data=AutoTrain)
pred <- predict(f2, expand.grid(weight=wt1, horsepower=hp1))
pred <- matrix(pred,30,30)

scatter3D(AutoTrain$weight,AutoTrain$horsepower,AutoTrain$mpg, pch = 18,
      surf = list(x = wt1, y = hp1, z = pred))
```



It looks to capture the pattern of the association, but a smoother surface might be better.

f)

```
mean(residuals(f1)^2)
```

```
## [1] 18.64557
```

```
mean(residuals(f2)^2)
```

```
## [1] 16.12921
```

the train mse is smaller for f2, so the fit is closer to the observed data

g)

```
pred1 <- predict(f1, AutoTest)
mean((pred1 - AutoTest$mpg)^2)
```

```
## [1] 14.81678
```

```
pred2 <- predict(f2, AutoTest)
mean((pred2 - AutoTest$mpg)^2, na.rm=T)
```

```
## [1] 14.70665
```

the test MSE is about the same for both fits. Choose simpler model (f1)