02_multipleReg

```
#### Example: Indian
# Anthropologists conducted a study to determine the long-term effects of an
# environmental change on systolic blood pressure. They measured the blood
# pressure and several other characteristics of 39 Indians who migrated from a
# very primitive environment high in the Andes into the mainstream of Peruvian
# society at a lower altitude. All of the Indians were males at least 21 years of
# age, and were born at a high altitude.
# filename
####################################################
fn.data <- "http://statacumen.com/teach/ADA2/ADA2_notes_Ch02_indian.dat"
indian <- read.table(fn.data, header=TRUE)
indian
```

```
##    id age yrmig   wt   ht chin fore calf pulse sysbp diabp
## 1   1  21     1 71.0 1629  8.0  7.0 12.7    88   170    76
## 2   2  22     6 56.5 1569  3.3  5.0  8.0    64   120    60
## 3   3  24     5 56.0 1561  3.3  1.3  4.3    68   125    75
## 4   4  24     1 61.0 1619  3.7  3.0  4.3    52   148   120
## 5   5  25     1 65.0 1566  9.0 12.7 20.7    72   140    78
## 6   6  27    19 62.0 1639  3.0  3.3  5.7    72   106    72
## 7   7  28     5 53.0 1494  7.3  4.7  8.0    64   120    76
## 8   8  28    25 53.0 1568  3.7  4.3  0.0    80   108    62
## 9   9  31     6 65.0 1540 10.3  9.0 10.0    76   124    70
## 10 10  32    13 57.0 1530  5.7  4.0  6.0    60   134    64
## 11 11  33    13 66.5 1622  6.0  5.7  8.3    68   116    76
## 12 12  33    10 59.1 1486  6.7  5.3 10.3    73   114    74
## 13 13  34    15 64.0 1578  3.3  5.3  7.0    88   130    80
## 14 14  35    18 69.5 1645  9.3  5.0  7.0    60   118    68
## 15 15  35     2 64.0 1648  3.0  3.7  6.7    60   138    78
## 16 16  36    12 56.5 1521  3.3  5.0 11.7    72   134    86
## 17 17  36    15 57.0 1547  3.0  3.0  6.0    84   120    70
## 18 18  37    16 55.0 1505  4.3  5.0  7.0    64   120    76
## 19 19  37    17 57.0 1473  6.0  5.3 11.7    72   114    80
## 20 20  38    10 58.0 1538  8.7  6.0 13.0    64   124    64
## 21 21  38    18 59.5 1513  5.3  4.0  7.7    80   114    66
## 22 22  38    11 61.0 1653  4.0  3.3  4.0    76   136    78
## 23 23  38    11 57.0 1566  3.0  3.0  3.0    60   126    72
## 24 24  39    21 57.5 1580  4.0  3.0  5.0    64   124    62
## 25 25  39    24 74.0 1647  7.3  6.3 15.7    64   128    84
## 26 26  39    14 72.0 1620  6.3  7.7 13.3    68   134    92
## 27 27  41    25 62.5 1637  6.0  5.3  8.0    76   112    80
## 28 28  41    32 68.0 1528 10.0  5.0 11.3    60   128    82
## 29 29  41     5 63.4 1647  5.3  4.3 13.7    76   134    92
## 30 30  42    12 68.0 1605 11.0  7.0 10.7    88   128    90
## 31 31  43    25 69.0 1625  5.0  3.0  6.0    72   140    72
## 32 32  43    26 73.0 1615 12.0  4.0  5.7    68   138    74
## 33 33  43    10 64.0 1640  5.7  3.0  7.0    60   118    66
## 34 34  44    19 65.0 1610  8.0  6.7  7.7    74   110    70
## 35 35  44    18 71.0 1572  3.0  4.7  4.3    72   142    84
## 36 36  45    10 60.2 1534  3.0  3.0  3.3    56   134    70
## 37 37  47     1 55.0 1536  3.0  3.0  4.0    64   116    54
```

```
## 38 38  50    43 70.0 1630  4.0  6.0 11.7    72   132    90
## 39 39  54    40 87.0 1542 11.3 11.7 11.3    92   152    88
```

```r
nrow(indian)
```

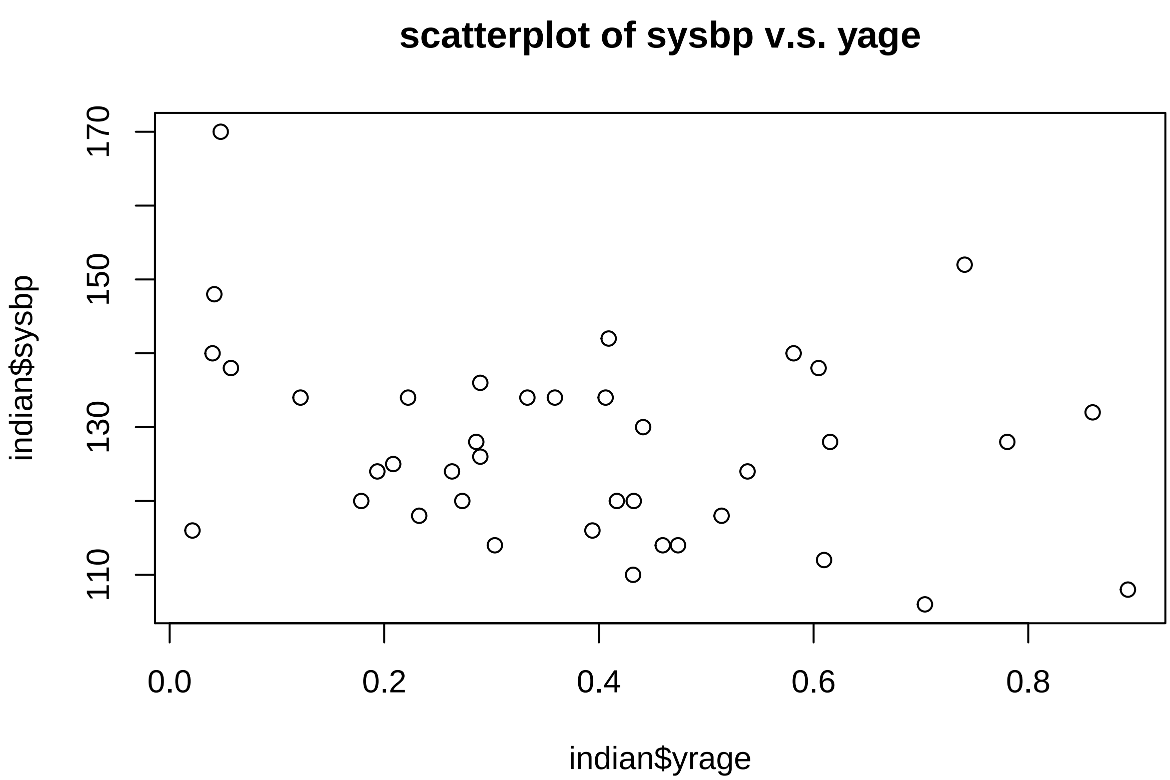
```
## [1] 39
```

```r
# examine the structure of the dataset, is it what you expected?
  # a data.frame containing integers, numbers, and factors
str(indian)
```

```
## 'data.frame':    39 obs. of  11 variables:
##  $ id   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age  : int  21 22 24 24 25 27 28 28 31 32 ...
##  $ yrmig: int  1 6 5 1 1 19 5 25 6 13 ...
##  $ wt   : num  71 56.5 56 61 65 62 53 53 65 57 ...
##  $ ht   : int  1629 1569 1561 1619 1566 1639 1494 1568 1540 1530 ...
##  $ chin : num  8 3.3 3.3 3.7 9 3 7.3 3.7 10.3 5.7 ...
##  $ fore : num  7 5 1.3 3 12.7 3.3 4.7 4.3 9 4 ...
##  $ calf : num  12.7 8 4.3 4.3 20.7 5.7 8 0 10 6 ...
##  $ pulse: int  88 64 68 52 72 72 64 80 76 60 ...
##  $ sysbp: int  170 120 125 148 140 106 120 108 124 134 ...
##  $ diabp: int  76 60 75 120 78 72 76 62 70 64 ...
```

```r
# Description of variables
#    id = individual id
#   age = age in years           yrmig = years since migration
#    wt = weight in kilos           ht = height in mm
#  chin = chin skin fold in mm    fore = forearm skin fold in mm
#  calf = calf skin fold in mm   pulse = pulse rate-beats/min
# sysbp = systolic bp            diabp = diastolic bp


# Create the "fraction of their life" variable
#   yrage = years since migration divided by age
indian$yrage <- indian$yrmig / indian$age
plot(indian$yrage,indian$sysbp,main="scatterplot of sysbp v.s. yage")  #weak linear relationship
```




```r
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot1.jpg")
```

```
## jpeg
##    3
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
####################################################
# fit the simple linear regression model
lm.sysbp.yrage <- lm(sysbp ~ yrage, data = indian)
# use Anova() from library(car) to get ANOVA table (Type 3 SS, df)
library(car)
```

```
## Loading required package: carData
```

```r
Anova(lm.sysbp.yrage, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: sysbp
##             Sum Sq Df   F value  Pr(>F)
## (Intercept) 178221  1 1092.9484 < 2e-16 ***
## yrage          498  1    3.0544 0.08881 .
## Residuals     6033 37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm.sysbp.yrage)  #compare to function anova
```

```
## Analysis of Variance Table
```

```
##
## Response: sysbp
##           Df Sum Sq Mean Sq F value  Pr(>F)
## yrage      1  498.1  498.06  3.0544 0.08881 .
## Residuals 37 6033.4  163.06
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# use summary() to get t-tests of parameters (slope, intercept)
summary(lm.sysbp.yrage)
```

```
##
## Call:
## lm(formula = sysbp ~ yrage, data = indian)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.161 -10.987  -1.014   6.851  37.254
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  133.496      4.038  33.060   <2e-16 ***
## yrage        -15.752      9.013  -1.748   0.0888 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.77 on 37 degrees of freedom
## Multiple R-squared:  0.07626,    Adjusted R-squared:  0.05129
## F-statistic: 3.054 on 1 and 37 DF,  p-value: 0.08881
```

```
####################################################
# fit the multiple linear regression model, (" + wt" added)
lm.sysbp.yrage.wt <- lm(sysbp ~ yrage + wt, data = indian)
library(car)
Anova(lm.sysbp.yrage.wt, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: sysbp
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 1738.2  1  18.183 0.0001385 ***
## yrage       1314.7  1  13.753 0.0006991 ***
## wt          2592.0  1  27.115 7.966e-06 ***
## Residuals   3441.4 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm.sysbp.yrage.wt)  #sequential ss
```

```
## Analysis of Variance Table
##
## Response: sysbp
##           Df Sum Sq Mean Sq F value    Pr(>F)
## yrage      1  498.1  498.06  5.2102   0.02846 *
## wt         1 2592.0 2592.01 27.1149 7.966e-06 ***
## Residuals 36 3441.4   95.59
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(lm.sysbp.yrage.wt)
```

```
##
## Call:
## lm(formula = sysbp ~ yrage + wt, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4330  -7.3070   0.8963   5.7275  23.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.8959    14.2809   4.264 0.000138 ***
## yrage       -26.7672     7.2178  -3.708 0.000699 ***
## wt            1.2169     0.2337   5.207 7.97e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.777 on 36 degrees of freedom
## Multiple R-squared:  0.4731, Adjusted R-squared:  0.4438
## F-statistic: 16.16 on 2 and 36 DF,  p-value: 9.795e-06
```

```r
## 95% confidence intervals for beta_0 and beta_1
confint(lm.sysbp.yrage.wt, level=.95)
```

```
##                   2.5 %     97.5 %
## (Intercept)  31.9329542  89.858895
## yrage       -41.4055934 -12.128836
## wt            0.7429167   1.690796
```

```r
## 95% CI's for E(Y) when (x1,x2)=(0.05,58), (x1,x2)=(0.03,64),
newdata <- data.frame(yrage=c(0.05,0.03), wt=c(58,64))
predict(lm.sysbp.yrage.wt, newdata=newdata, interval="confidence", level=.95)
```

```
##        fit      lwr      upr
## 1 130.1352 124.3695 135.9010
## 2 137.9717 131.7534 144.1901
```

```r
## 95% CI's for E(Y) using Bonferroni Correction
newdata <- data.frame(yrage=c(0.05,0.03), wt=c(58,64))
predict(lm.sysbp.yrage.wt, newdata=newdata, interval="confidence", level=1-.05/3)
```

```
##        fit      lwr      upr
## 1 130.1352 122.9965 137.2739
## 2 137.9717 130.2726 145.6708
```

```r
## 95% Prediction Intervals for Y_new when (x1,x2)=(0.05,58), (x1,x2)=(0.03,64),
newdata <- data.frame(yrage=c(0.05,0.03), wt=c(58,64))
predict(lm.sysbp.yrage.wt, newdata=newdata, interval="prediction", level=.95)
```

```
##        fit      lwr      upr
## 1 130.1352 109.4849 150.7855
## 2 137.9717 117.1905 158.7529
```

```r
## 95% PI's for Y_new using Bonferroni Correction
newdata <- data.frame(yrage=c(0.05,0.03), wt=c(58,64))
predict(lm.sysbp.yrage.wt, newdata=newdata, interval="prediction", level=1-.05/3)
```

```
##        fit      lwr      upr
## 1 130.1352 104.5675 155.7030
## 2 137.9717 112.2419 163.7015
```

```r
####################################################
# fit the multiple linear regression model, (" + wt" added first, then fraction)
lm.sysbp.yrage.wt2 <- lm(sysbp ~ wt+yrage, data = indian)
Anova(lm.sysbp.yrage.wt2, type=3)
```

```
## Anova Table (Type III tests)
##
## Response: sysbp
##             Sum Sq Df F value    Pr(>F)
## (Intercept) 1738.2  1  18.183 0.0001385 ***
## wt          2592.0  1  27.115 7.966e-06 ***
## yrage       1314.7  1  13.753 0.0006991 ***
## Residuals   3441.4 36
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(lm.sysbp.yrage.wt2)  #sequential ss
```

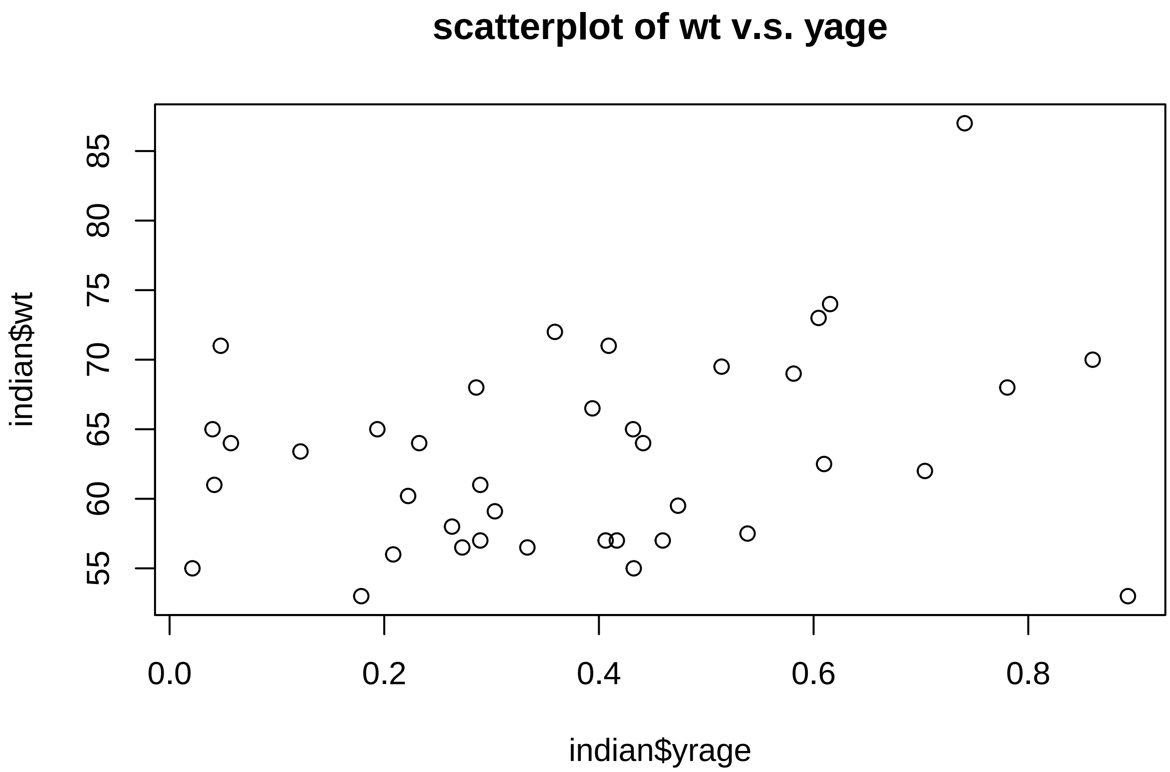
```
## Analysis of Variance Table
##
## Response: sysbp
##           Df Sum Sq Mean Sq F value    Pr(>F)
## wt         1 1775.4 1775.38  18.572 0.0001210 ***
## yrage      1 1314.7 1314.69  13.753 0.0006991 ***
## Residuals 36 3441.4   95.59
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
summary(lm.sysbp.yrage.wt2)
```

```
##
## Call:
## lm(formula = sysbp ~ wt + yrage, data = indian)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.4330  -7.3070   0.8963   5.7275  23.9819
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.8959    14.2809   4.264 0.000138 ***
## wt            1.2169     0.2337   5.207 7.97e-06 ***
## yrage       -26.7672     7.2178  -3.708 0.000699 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.777 on 36 degrees of freedom
## Multiple R-squared:  0.4731, Adjusted R-squared:  0.4438
```

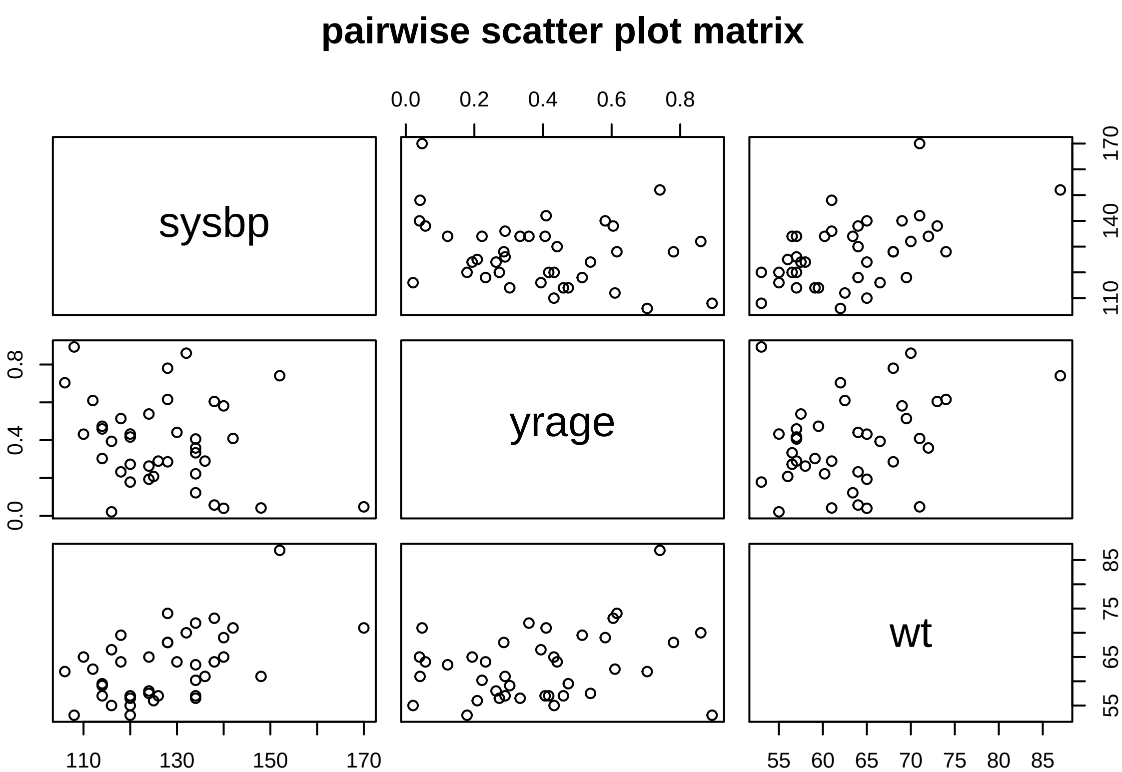
```
## F-statistic: 16.16 on 2 and 36 DF,  p-value: 9.795e-06
```

```
plot(indian$yrage,indian$wt,main="scatterplot of wt v.s. yage")
```




```
####################################################
##check model assumptions

##scatterplot matrix, check for possible non-linear relationships
pairs(sysbp ~ yrage+wt,data=indian, main="pairwise scatter plot matrix")
```




```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot2.jpg")
```

```
## jpeg
##    3
```

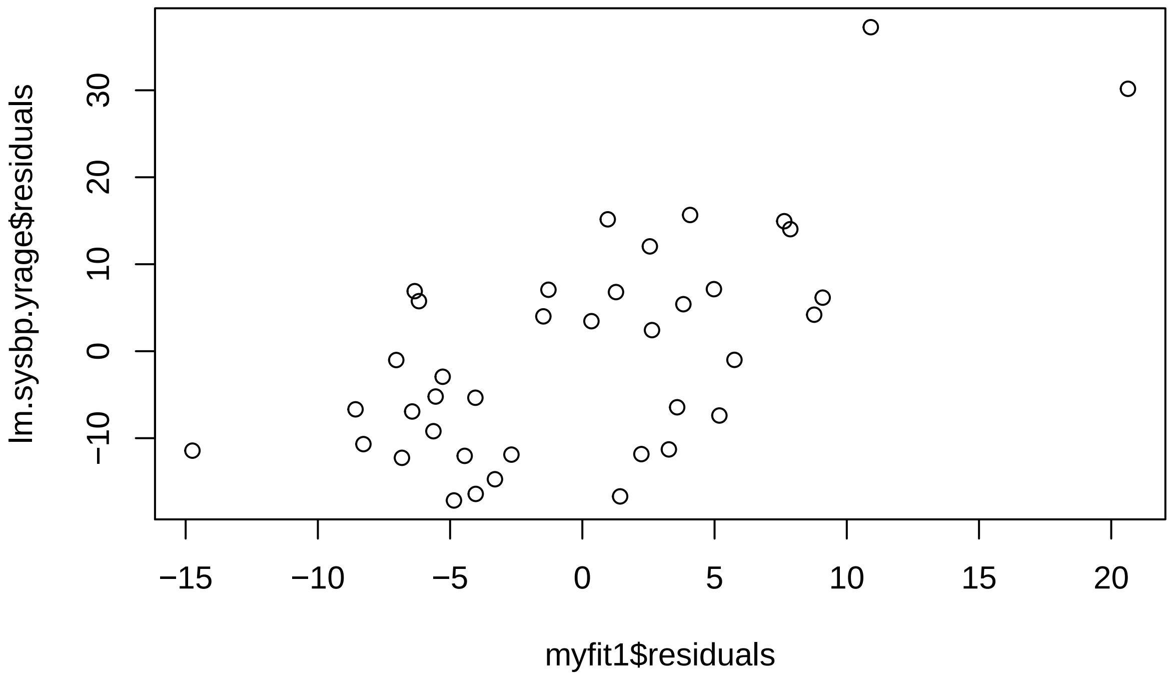
```
dev.off()
```

```
## pdf
##   2
```

```
cor(cbind(indian$sysbp,indian$yrage,indian$wt))
```

```
##            [,1]       [,2]      [,3]
## [1,]  1.0000000 -0.2761457 0.5213643
## [2,] -0.2761457  1.0000000 0.2930830
## [3,]  0.5213643  0.2930830 1.0000000
```

```
##added variable plot
#recall lm.sysbp.yrage <- lm(sysbp ~ yrage, data = indian)
myfit1 <- lm(wt ~ yrage, data = indian)
plot(myfit1$residuals,lm.sysbp.yrage$residuals)
```



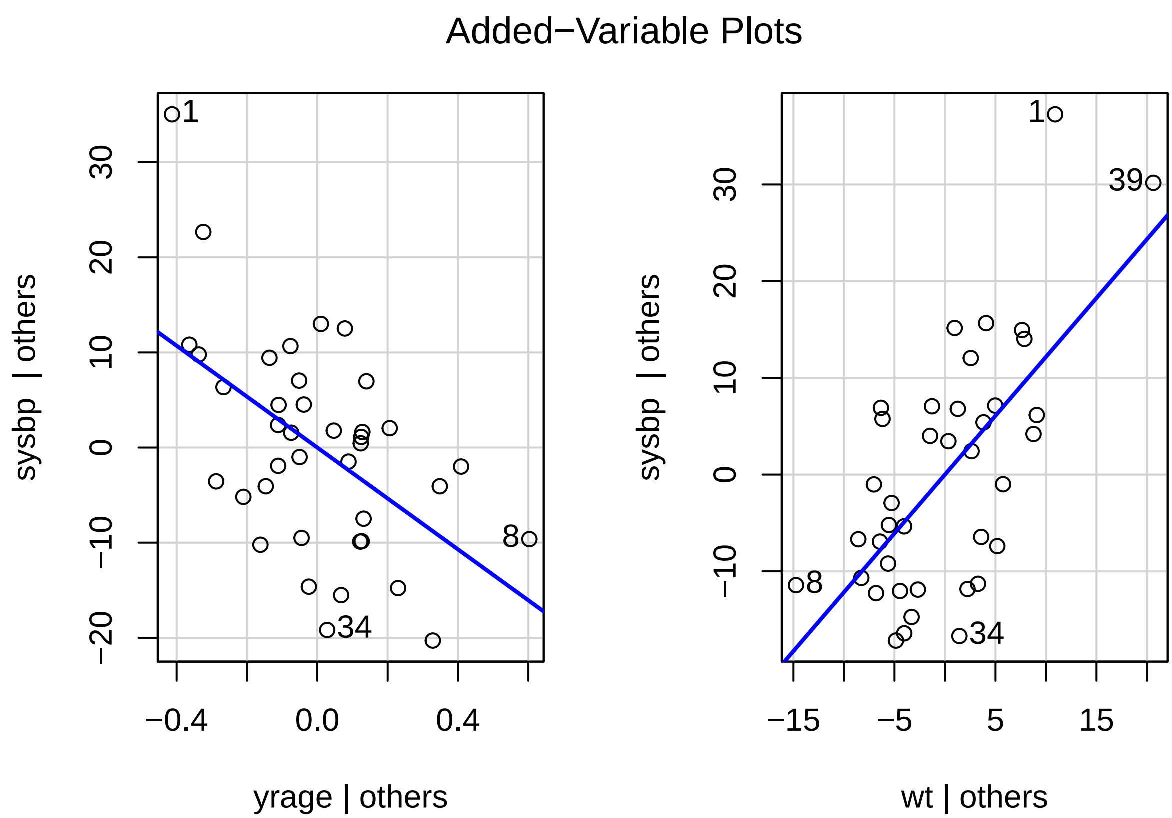

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot3.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#or a simple way
library(car)
avPlots(lm.sysbp.yrage.wt)
```




```
#see a linear relationship, wt is important to add to the model
#diagnostic plots
par(mfrow=c(2,3))
plot(lm.sysbp.yrage.wt,which=c(1,4))
plot(indian$yrage, lm.sysbp.yrage.wt$residuals, main="Residuals vs yrage")
# horizontal line at zero
abline(h = 0, col = "gray75")
plot(indian$wt, lm.sysbp.yrage.wt$residuals, main="Residuals vs wt")
# horizontal line at zero
abline(h = 0, col = "gray75")
library(car)
qqPlot(lm.sysbp.yrage.wt$residuals, las = 1, main="QQ Plot")
```

```
## [1]  1 34
```

```
lev<-hatvalues(lm.sysbp.yrage.wt)
plot(lev)
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot4.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

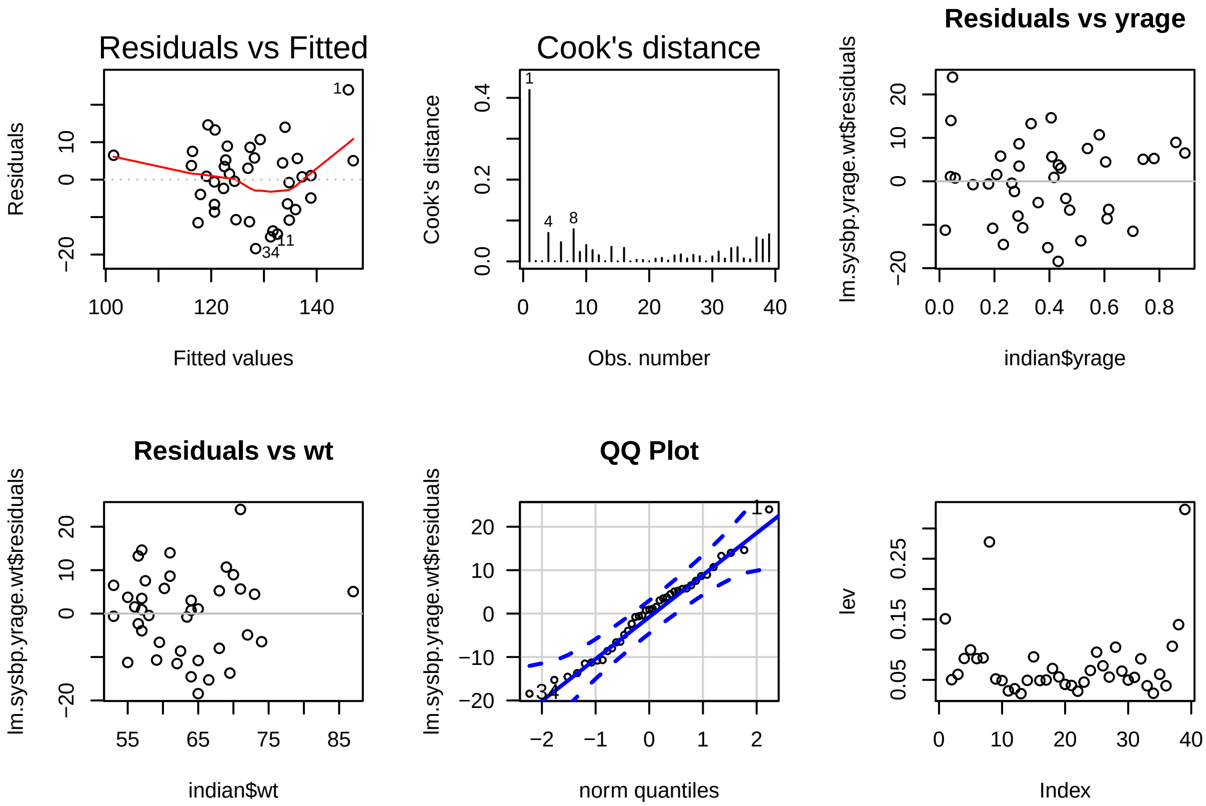
```
## pdf
##   2
```

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```




```
bptest(sysbp~yrage+wt,data=indian,studentize=FALSE)   #test constant variance
```

```
##
##  Breusch-Pagan test
##
## data:  sysbp ~ yrage + wt
## BP = 2.3797, df = 2, p-value = 0.3043
```

```
shapiro.test(rstandard(lm.sysbp.yrage.wt))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  rstandard(lm.sysbp.yrage.wt)
## W = 0.98081, p-value = 0.733
```

```
#####################################################
##outliers
rstudent(lm.sysbp.yrage.wt)  ##gives rstudent values
```

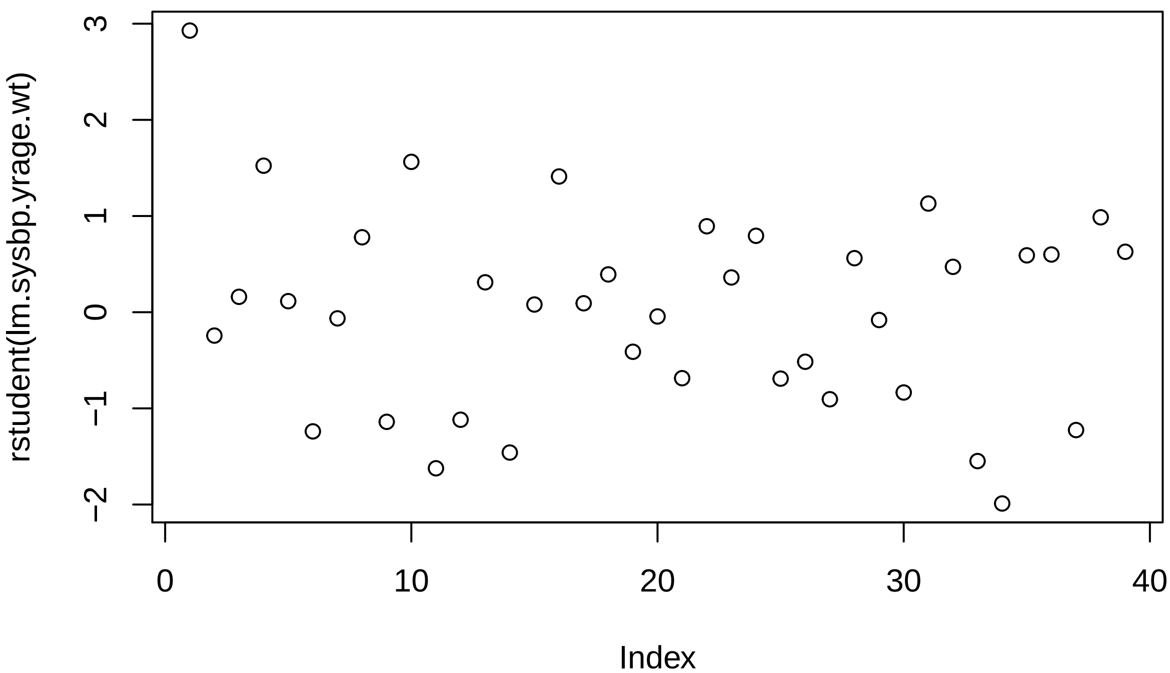
```
##           1           2           3           4           5           6
##  2.92834234 -0.24318932  0.15980796  1.52347845  0.11470330 -1.23933893
##           7           8           9          10          11          12
## -0.06430383  0.77912370 -1.14006156  1.56336665 -1.62309601 -1.11820762
```

```
##           13          14          15          16          17          18
##  0.31068368 -1.45951018  0.07971536  1.41112410  0.09273330  0.39295645
##           19          20          21          22          23          24
## -0.41160726 -0.04427766 -0.68618697  0.89368330  0.36123695  0.79446998
##           25          26          27          28          29          30
## -0.69093206 -0.51533483 -0.90531641  0.56176832 -0.08136759 -0.83518488
##           31          32          33          34          35          36
##  1.12990037  0.47145833 -1.54823185 -1.98907839  0.59115467  0.59988613
##           37          38          39
## -1.22542691  0.98671600  0.62822330
```

```
outlierTest(lm.sysbp.yrage.wt)  ##Reports the Bonferroni p-value for the most extreme observation.
```

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##   rstudent unadjusted p-value Bonferonni p
## 1 2.928342          0.0059575      0.23234
```

```
par(mfrow=c(1,1))
plot(rstudent(lm.sysbp.yrage.wt))
```




```
indian[1,]
```

```
##   id age yrmig wt   ht chin fore calf pulse sysbp diabp      yrage
## 1  1  21     1 71 1629    8    7 12.7    88   170    76 0.04761905
```

```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot5.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
##leverage, x outliers
xoutliers <- which(lev > 2*3/39)
xoutliers
```

```
##  8 39
##  8 39
```

```
lev[xoutliers]
```

```
##         8        39
## 0.2776435 0.3314108
```

```
xoutliers2 <- which(lev > 3*3/39)
xoutliers2
```

```
##  8 39
##  8 39
```

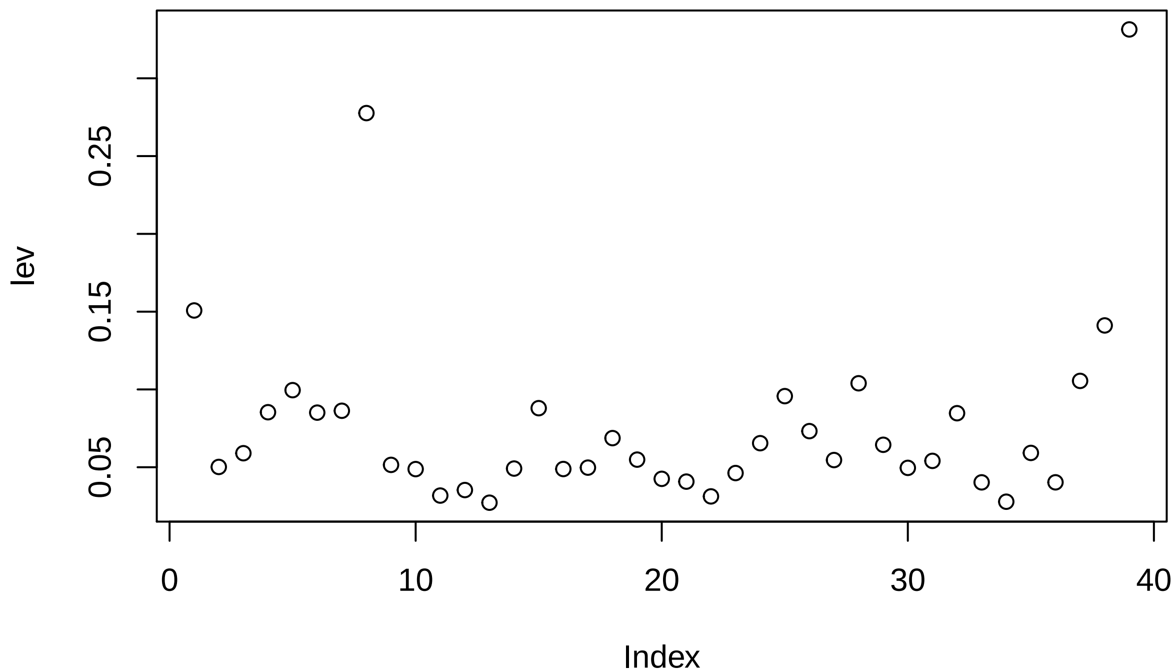
```
indian[8,]
```

```
##   id age yrmig wt   ht chin fore calf pulse sysbp diabp     yrage
## 8  8  28    25 53 1568  3.7  4.3    0    80   108    62 0.8928571
```

```
indian[39,]
```

```
##    id age yrmig wt   ht chin fore calf pulse sysbp diabp     yrage
## 39 39  54    40 87 1542 11.3 11.7 11.3    92   152    88 0.7407407
```

```
plot(lev)
```




```
#cooks distance
cooks.distance(lm.sysbp.yrage.wt)
```

```
##            1            2            3            4            5
## 4.192095e-01 1.069485e-03 5.486670e-04 6.963747e-02 4.984699e-04
##            6            7            8            9           10
## 4.692733e-02 1.338771e-04 7.863108e-02 2.333088e-02 4.018174e-02
```

```
##            11           12           13           14           15
## 2.757758e-02 1.516266e-02 9.230339e-04 3.557985e-02 2.100991e-04
##            16           17           18           19           20
## 3.317841e-02 1.543065e-04 3.889378e-03 3.356893e-03 2.986433e-05
##            21           22           23           24           25
## 6.767459e-03 8.637264e-03 2.161711e-03 1.488762e-02 1.708655e-02
##            26           27           28           29           30
## 7.137663e-03 1.586255e-02 1.244151e-02 1.562861e-04 1.223323e-02
##            31           32           33           34           35
## 2.414600e-02 7.009472e-03 3.227358e-02 3.489925e-02 7.464976e-03
##            36           37           38           39
## 5.128108e-03 5.822101e-02 5.336894e-02 6.632523e-02
```

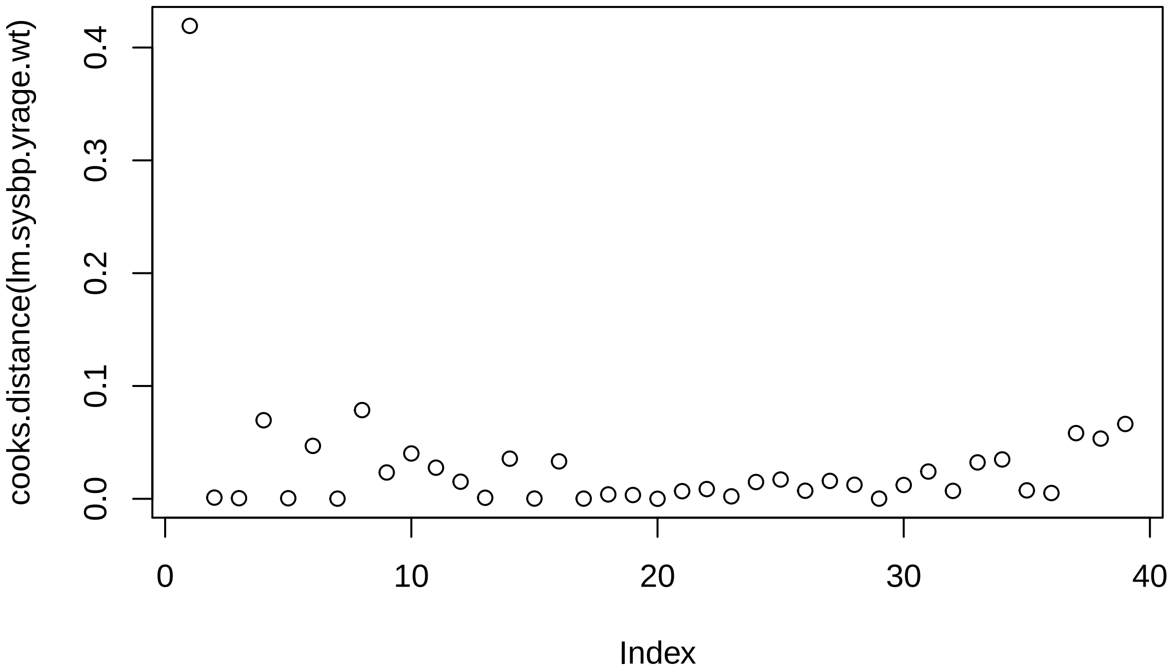
```
max(cooks.distance(lm.sysbp.yrage.wt))
```

```
## [1] 0.4192095
```

```
order(cooks.distance(lm.sysbp.yrage.wt))[39]
```

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

```
plot(cooks.distance(lm.sysbp.yrage.wt))
```



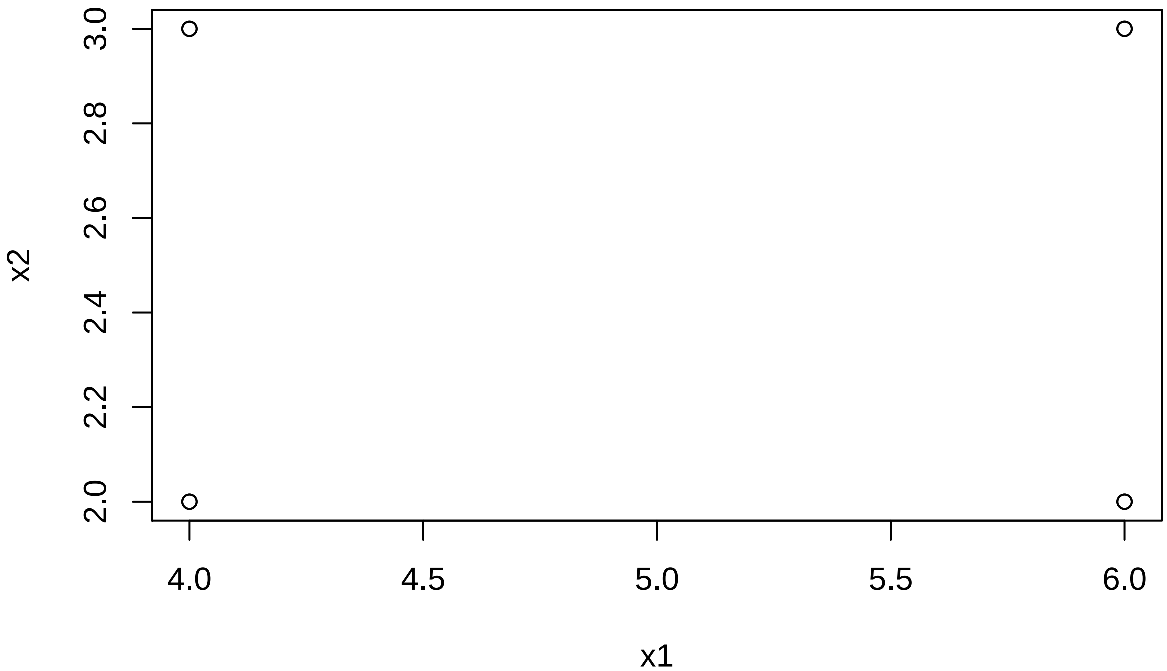

```
highcook <- which((cooks.distance(lm.sysbp.yrage.wt)) > qf(0.5,3,36))
cooks.distance(lm.sysbp.yrage.wt)[highcook]
```

```
## named numeric(0)
```

```
################## Two extreme cases ####################

########## extreme case when the predictor variables are uncorrelated #######
ex.data1<-read.table(file="~/Desktop/jenn/teaching/stat440540/data/CH7/CH07TA06.txt")
x1 <- ex.data1$V1
x2 <- ex.data1$V2
y <- ex.data1$V3
```

```r
plot(x1,x2)
```




```r
cor(cbind(y, x1, x2))  ##notice that x1 and x2 are uncorrelated
```

```
##            y        x1        x2
## y  1.0000000 0.7419309 0.6384057
## x1 0.7419309 1.0000000 0.0000000
## x2 0.6384057 0.0000000 1.0000000
```

```r
## Fit the model  y = b0 + b1*x1 + b2*x2
myfit1 <- lm(y ~ x1 + x2, data=ex.data1)
summary(myfit1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = ex.data1)
##
## Residuals:
##      1      2      3      4      5      6      7      8
##  1.625 -1.375 -1.625  1.375 -2.125  1.875  0.625 -0.375
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3750     4.7405   0.079 0.940016
## x1            5.3750     0.6638   8.097 0.000466 ***
## x2            9.2500     1.3276   6.968 0.000937 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.877 on 5 degrees of freedom
## Multiple R-squared:  0.958,  Adjusted R-squared:  0.9412
## F-statistic: 57.06 on 2 and 5 DF,  p-value: 0.000361
```

```r
anova(myfit1)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## x1         1 231.125 231.125  65.567 0.0004657 ***
## x2         1 171.125 171.125  48.546 0.0009366 ***
## Residuals  5  17.625   3.525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Fit the model  y = b0 + b1*x2 + b2*x1
myfit1_1 <- lm(y ~ x2 + x1, data=ex.data1)
summary(myfit1_1)
```

```
##
## Call:
## lm(formula = y ~ x2 + x1, data = ex.data1)
##
## Residuals:
##      1      2      3      4      5      6      7      8
##  1.625 -1.375 -1.625  1.375 -2.125  1.875  0.625 -0.375
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.3750     4.7405   0.079 0.940016
## x2            9.2500     1.3276   6.968 0.000937 ***
## x1            5.3750     0.6638   8.097 0.000466 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.877 on 5 degrees of freedom
## Multiple R-squared:  0.958,  Adjusted R-squared:  0.9412
## F-statistic: 57.06 on 2 and 5 DF,  p-value: 0.000361
```

```r
anova(myfit1_1)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## x2         1 171.125 171.125  48.546 0.0009366 ***
## x1         1 231.125 231.125  65.567 0.0004657 ***
## Residuals  5  17.625   3.525
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## Fit the model  y = b0 + b1*x1
myfit2 <- lm(y ~ x1, data=ex.data1)
summary(myfit2)
```

```
##
## Call:
## lm(formula = y ~ x1, data = ex.data1)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.750 -3.750  0.125  4.500  6.000
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   23.500     10.111   2.324   0.0591 .
## x1             5.375      1.983   2.711   0.0351 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.609 on 6 degrees of freedom
## Multiple R-squared:  0.5505, Adjusted R-squared:  0.4755
## F-statistic: 7.347 on 1 and 6 DF,  p-value: 0.03508
```

```
anova(myfit2)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## x1         1 231.12 231.125   7.347 0.03508 *
## Residuals  6 188.75  31.458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Fit the model  y = b0 + b1*x2
myfit3 <- lm(y ~ x2, data=ex.data1)
summary(myfit3)
```

```
##
## Call:
## lm(formula = y ~ x2, data = ex.data1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.000 -4.688 -0.250  5.250  7.250
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   27.250     11.608   2.348   0.0572 .
## x2             9.250      4.553   2.032   0.0885 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 6 degrees of freedom
## Multiple R-squared:  0.4076, Adjusted R-squared:  0.3088
## F-statistic: 4.128 on 1 and 6 DF,  p-value: 0.08846
```

```
anova(myfit3)
```

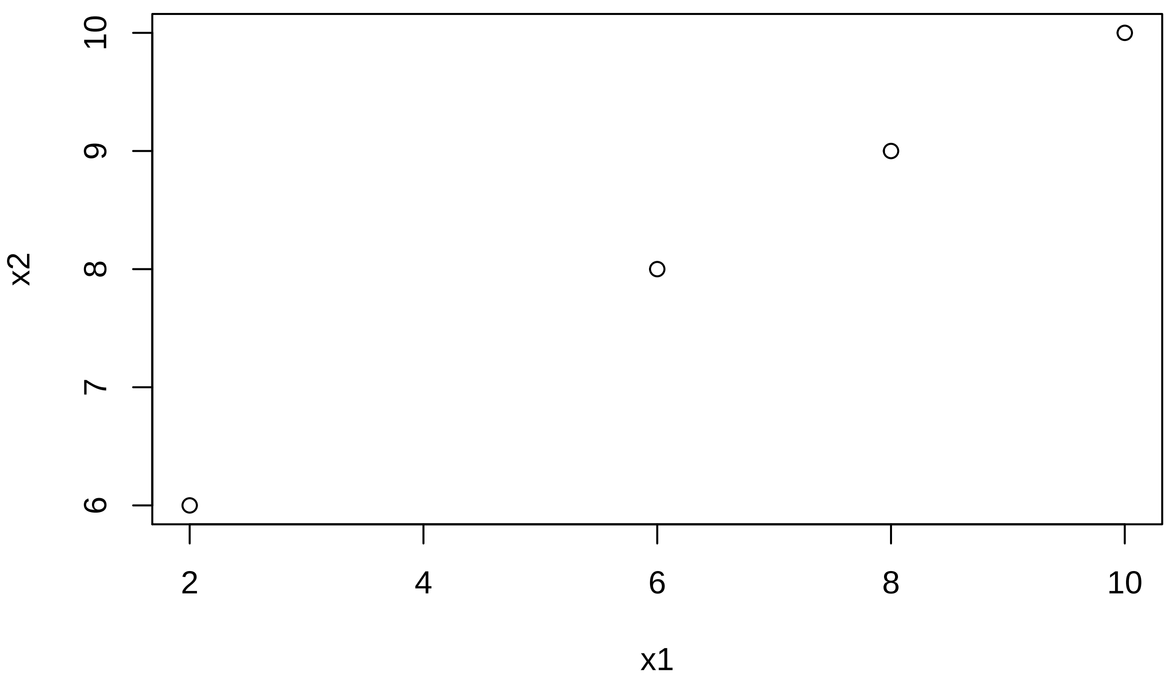
```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value  Pr(>F)
## x2         1 171.12 171.125  4.1276 0.08846 .
```

```
## Residuals  6 248.75  41.458
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
########## extreme case when the predictor variables are perfectly correlated #######
ex.data2<-read.table(file="~/Desktop/jenn/teaching/stat440540/data/CH7/CH07TA08.txt")
x1 <- ex.data2$V1
x2 <- ex.data2$V2
y <- ex.data2$V3
ex.data2
```

```
##   V1 V2  V3
## 1  2  6  23
## 2  8  9  83
## 3  6  8  63
## 4 10 10 103
```

```
plot(x1,x2) ##notice that x1 and x2 are perfectly correlated by x2= 5+.5 x1
```




```
## Fit the model  y = b0 + b1*x1 + b2*x2
myfit4 <- lm(y ~ x1 + x2, data=ex.data2)
summary(myfit4)
```

```
## Warning in summary.lm(myfit4): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = ex.data2)
##
## Residuals:
##          1          2          3          4
##  1.470e-15 -5.515e-15 -1.834e-16  4.228e-15
##
## Coefficients: (1 not defined because of singularities)
```

```
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 3.000e+00  6.065e-15 4.946e+14   <2e-16 ***
## x1          1.000e+01  8.493e-16 1.177e+16   <2e-16 ***
## x2                 NA         NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.024e-15 on 2 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.386e+32 on 1 and 2 DF,  p-value: < 2.2e-16
```

```
anova(myfit4)
```

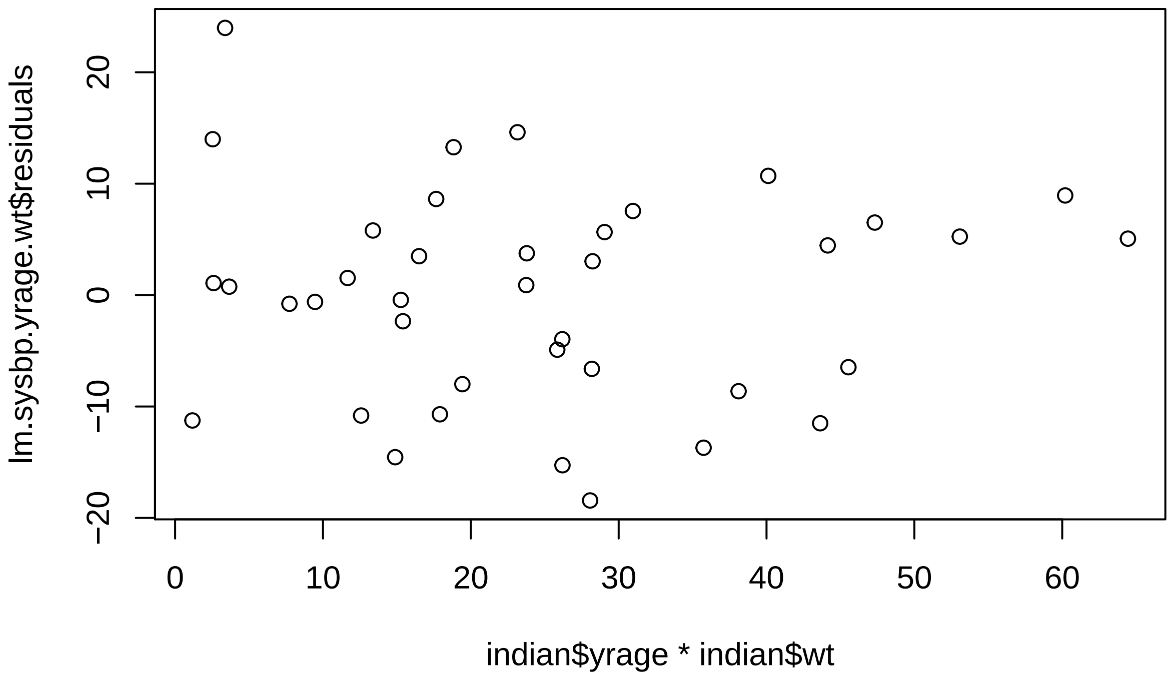
```
## Warning in anova.lm(myfit4): ANOVA F-tests on an essentially perfect fit
## are unreliable

## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq    F value    Pr(>F)
## x1         1   3500    3500 1.3865e+32 < 2.2e-16 ***
## Residuals  2      0       0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
####################################################
#check for multicollinearity
vif(lm.sysbp.yrage.wt)
```

```
##    yrage       wt
## 1.093969 1.093969
```

```
##check if interaction term need to be included
plot(indian$yrage*indian$wt,lm.sysbp.yrage.wt$residuals)
```



```
dev.copy(jpeg,filename="~/Desktop/jenn/teaching/ADA2/lecture notes/plots/plot6.jpg")
```

```
## jpeg
##    3
```

```
dev.off()
```

```
## pdf
##   2
```

```
#random pattern, do not need to add interaction term
```