R Program Notes
Biostatistics: A Guide to Design, Analysis, and Discovery Second Edition
by Ronald N. Forthofer, Eun Sul Lee, Mike Hernandez
Chapter 13: Linear Regression

Program Notes Outline

Program Note 13.1 – Simple Linear Regression

The following **R** commands are used to perform simple linear regression. The data below can be found in **Table 13.1** in the textbook.

R Commands:

```
sbp<- c(105, 90, 82, 96, 82, 74, 104, 100, 80, 98, 96, 86, 88, 128, 118,
        90, 108, 120, 114, 78, 116, 74, 80, 98, 90, 92, 80, 88, 104, 100,
        126, 108, 106, 98, 94, 88, 110, 124, 86, 120, 112, 100, 122, 122,
        110, 124, 122, 94, 110, 140)

height<- c(36, 37, 38, 38, 39, 39, 40, 40, 41, 42, 43, 44, 44, 44, 45, 46,
           48, 48, 49, 49, 50, 50, 51, 52, 53, 54, 54, 54, 55, 56, 57, 58,
           59, 59, 59, 60, 60, 60, 61, 61, 62, 63, 64, 64, 65, 65, 66, 67,
           67, 69)
```

R output:

```
> sbp
 [1] 105  90  82  96  82  74 104 100  80  98  96  86  88 128 118  90 108 120
114  78
[21] 116  74  80  98  90  92  80  88 104 100 126 108 106  98  94  88 110 124
86 120
[41] 112 100 122 122 110 124 122  94 110 140

> height
 [1] 36 37 38 38 39 39 40 40 41 42 43 44 44 44 45 46 48 48 49 49 50 50 51 52
53 54
[27] 54 54 55 56 57 58 59 59 59 60 60 60 61 61 62 63 64 64 65 65 66 67 67 69
```

The **R** function **lm( )** is used to fit linear regression models. With systolic blood pressure (**sbp**) as the dependent variable and **height** as the independent variable, the **R** commands provided below show the results of running a simple linear regression.

R commands:

```
lm(sbp~height)
```

R output:

```
Call:
lm(formula = sbp ~ height)

Coefficients:
(Intercept)       height
    61.1391       0.7688
```

A better way of doing this is to assign **lm(sbp~height)** to an object (ie. **sbpht**). Using the **summary( )** function, information on residuals, p-values corresponding to parameter estimates, and values for the adjusted and unadjusted coefficient of determination become available as shown below.

R commands:

```
sbpht<- lm(sbp~height)
summary(sbpht)
```

R output:

```
Call:
lm(formula = sbp ~ height)

Residuals:
     Min       1Q   Median       3Q      Max
-25.5781 -10.3829  -0.0405  11.6590  33.0346

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  61.1391    11.5381   5.299 2.89e-06 ***
height        0.7688     0.2163   3.555 0.000862 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 14.52 on 48 degrees of freedom
Multiple R-Squared: 0.2084,     Adjusted R-squared: 0.1919
F-statistic: 12.64 on 1 and 48 DF,  p-value: 0.0008617
```

To obtain the predicted values shown in **Table 13.1**, we start by using the function **names( )** first to view the information that is automatically provided by the **R** function.

R commands:

```
names(sbpht)
```

R output:

```
> names(sbpht)
 [1] "coefficients"  "residuals"     "effects"       "rank"
"fitted.values"
 [6] "assign"        "qr"            "df.residual"   "xlevels"       "call"
[11] "terms"         "model"
```

A list of the predicted values for systolic blood pressure can be obtained using the **R** command shown below. Refer to **Table 13.1**.

R commands:

```
sbpht$fitted.values
```

R output:

```
       1         2         3         4         5         6         7
8
 88.81516  89.58393  90.35271  90.35271  91.12149  91.12149  91.89027
91.89027
       9        10        11        12        13        14        15
16
 92.65905  93.42783  94.19660  94.96538  94.96538  94.96538  95.73416
96.50294
      17        18        19        20        21        22        23
24
 98.04050  98.04050  98.80928  98.80928  99.57805  99.57805 100.34683
101.11561
      25        26        27        28        29        30        31
32
101.88439 102.65317 102.65317 102.65317 103.42195 104.19072 104.95950
105.72828
      33        34        35        36        37        38        39
40
106.49706 106.49706 106.49706 107.26584 107.26584 107.26584 108.03462
108.03462
      41        42        43        44        45        46        47
48
108.80340 109.57217 110.34095 110.34095 111.10973 111.10973 111.87851
112.64729
      49        50
112.64729 114.18484
```

The standardized residuals and leverage values shown in **Table 13.2** can be obtained using the **library(MASS)** followed by the **R** functions **stdres( )** and **hatvalues( )**. The following formula is used to calculate the standardized residuals:

$$standized\ residual_i = \frac{residual_i}{s_{Y|X} * \sqrt{1 - leverage_i}}$$

where $s_{Y|X}$ is the *square root* of the sample variance of *Y* given *X*. The **R** commands below can be used to obtain standardized residuals, studentized residuals and leverage values.

R Commands:

```
library(MASS)
stdres(sbpht)
studres(sbpht)
hatvalues(sbpht)
```

R Output:

```
> library(MASS)
> stdres(sbpht)
          1           2           3           4           5           6
7
 1.16253316  0.02977066 -0.59552460  0.40263548 -0.64818243 -1.21667055
0.85789976
          8           9          10          11          12          13
14
 0.57452442 -0.89429999  0.32217559  0.12678112 -0.62897147 -0.48866036
2.31756197
         15          16          17          18          19          20
21
 1.55920356 -0.45465022  0.69456978  1.53144261  1.05842943 -1.44990777
1.14343744
         22          23          24          25          26          27
28
-1.78096459 -1.41607981 -0.21678826 -0.82693129 -0.74142920 -1.57659397 -
1.01981746
         29          30          31          32          33          34
35
 0.04024910 -0.29199306  1.46735169  0.15860880 -0.03475172 -0.59406822 -
0.87372647
         36          37          38          39          40          41
42
-1.34912341  0.19146437  1.17183842 -1.54585083  0.83943815  0.22472566 -
0.67449773
         43          44          45          46          47          48
49
 0.82365487  0.82365487 -0.07861756  0.91319616  0.71924344 -1.32949841 -
0.18874403
         50
 1.85426406
```

```
> studres(sbpht)
          1           2           3           4           5           6
7
 1.16690462  0.02945918 -0.59147770  0.39909380 -0.64422059 -1.22293511
0.85550037
          8           9          10          11          12          13
14
 0.57047314 -0.89240112  0.31914718  0.12547455 -0.62496594 -0.48475062
2.43348228
         15          16          17          18          19          20
21
 1.58349811 -0.45086119  0.69077672  1.55384247  1.05978629 -1.46721422
1.14719577
         22          23          24          25          26          27
28
-1.82359643 -1.43147141 -0.21462326 -0.82416372 -0.73790289 -1.60211841 -
1.02025200
         29          30          31          32          33          34
35
 0.03982830 -0.28919241  1.48569019  0.15698907 -0.03438825 -0.59002048 -
0.87153553
         36          37          38          39          40          41
42
-1.36105061  0.18953184  1.17651883 -1.56922361  0.83681304  0.22248952 -
0.67062039
         43          44          45          46          47          48
49
 0.82085136  0.82085136 -0.07779932  0.91158705  0.71557834 -1.34048936 -
0.18683695
         50
 1.90431861


> hatvalues(sbpht)
          1           2           3           4           5           6           7
0.08041274 0.07331188 0.06665483 0.06665483 0.06044158 0.06044158 0.05467214
          8           9          10          11          12          13          14
0.05467214 0.04934650 0.04446466 0.04002663 0.03603240 0.03603240 0.03603240
         15          16          17          18          19          20          21
0.03248197 0.02937535 0.02449351 0.02449351 0.02271830 0.02271830 0.02138689
         22          23          24          25          26          27          28
0.02138689 0.02049928 0.02005548 0.02005548 0.02049928 0.02049928 0.02049928
         29          30          31          32          33          34          35
0.02138689 0.02271830 0.02449351 0.02671253 0.02937535 0.02937535 0.02937535
         36          37          38          39          40          41          42
0.03248197 0.03248197 0.03248197 0.03603240 0.03603240 0.04002663 0.04446466
         43          44          45          46          47          48          49
0.04934650 0.04934650 0.05467214 0.05467214 0.06044158 0.06665483 0.06665483
         50
0.08041274
```

The **R** command below is used to calculate the standardized residuals using the formula shown above.

R commands:

```
# The MASS library allows the use of the hatvalues( ) function
library(MASS)

# The square root of the variance of Y given X
syx<- sqrt((sum((sbp-sbpht$fitted.values)**2))/48)

# Standardized Residuals
sbpht$residual/(syx*sqrt(1-hatvalues(sbpht)))
```

R output:

```
> sbpht$residual/(syx*sqrt(1-hatvalues(sbpht)))
          1           2           3           4           5           6
 1.16253316  0.02977066 -0.59552460  0.40263548 -0.64818243 -1.21667055
          7           8           9          10          11          12
 0.85789976  0.57452442 -0.89429999  0.32217559  0.12678112 -0.62897147
         13          14          15          16          17          18
-0.48866036  2.31756197  1.55920356 -0.45465022  0.69456978  1.53144261
         19          20          21          22          23          24
 1.05842943 -1.44990777  1.14343744 -1.78096459 -1.41607981 -0.21678826
         25          26          27          28          29          30
-0.82693129 -0.74142920 -1.57659397 -1.01981746  0.04024910 -0.29199306
         31          32          33          34          35          36
 1.46735169  0.15860880 -0.03475172 -0.59406822 -0.87372647 -1.34912341
         37          38          39          40          41          42
 0.19146437  1.17183842 -1.54585083  0.83943815  0.22472566 -0.67449773
         43          44          45          46          47          48
 0.82365487  0.82365487 -0.07861756  0.91319616  0.71924344 -1.32949841
         49          50
-0.18874403  1.85426406
```

At this point, we introduce an alternative approach that could have been used which starts by creating a data frame using the variables **sbp** and **height**. We show the **R** commands and output below. Notice that the data now appear in columns and comparisons between observations are much easier to make.

R commands:

```
sbp<- c(105, 90, 82, 96, 82, 74, 104, 100, 80, 98, 96, 86, 88, 128, 118,
        90, 108, 120, 114, 78, 116, 74, 80, 98, 90, 92, 80, 88, 104, 100,
        126, 108, 106, 98, 94, 88, 110, 124, 86, 120, 112, 100, 122, 122,
        110, 124, 122, 94, 110, 140)

height<- c(36, 37, 38, 38, 39, 39, 40, 40, 41, 42, 43, 44, 44, 44, 45, 46,
           48, 48, 49, 49, 50, 50, 51, 52, 53, 54, 54, 54, 55, 56, 57, 58,
           59, 59, 59, 60, 60, 60, 61, 61, 62, 63, 64, 64, 65, 65, 66, 67,
           67, 69)

sbphtdat<- data.frame(cbind(sbp,height))
sbphtdat
```

R output:

```
   sbp height
1  105     36
2   90     37
3   82     38
4   96     38
5   82     39
6   74     39
7  104     40
8  100     40
9   80     41
10  98     42
11  96     43
12  86     44
13  88     44
14 128     44
15 118     45
16  90     46
17 108     48
18 120     48
19 114     49
20  78     49
21 116     50
22  74     50
23  80     51
24  98     52
25  90     53
26  92     54
27  80     54
28  88     54
29 104     55
30 100     56
31 126     57
32 108     58
33 106     59
34  98     59
35  94     59
36  88     60
37 110     60
38 124     60
39  86     61
40 120     61
41 112     62
42 100     63
43 122     64
44 122     64
45 110     65
46 124     65
47 122     66
48  94     67
49 110     67
50 140     69
```

Program Note 13.2 – Confidence and Prediction Intervals

1. Confidence Intervals for Simple Linear Regression

To obtain confidence intervals for $\mu_{Y|X}$, we use the following formula:

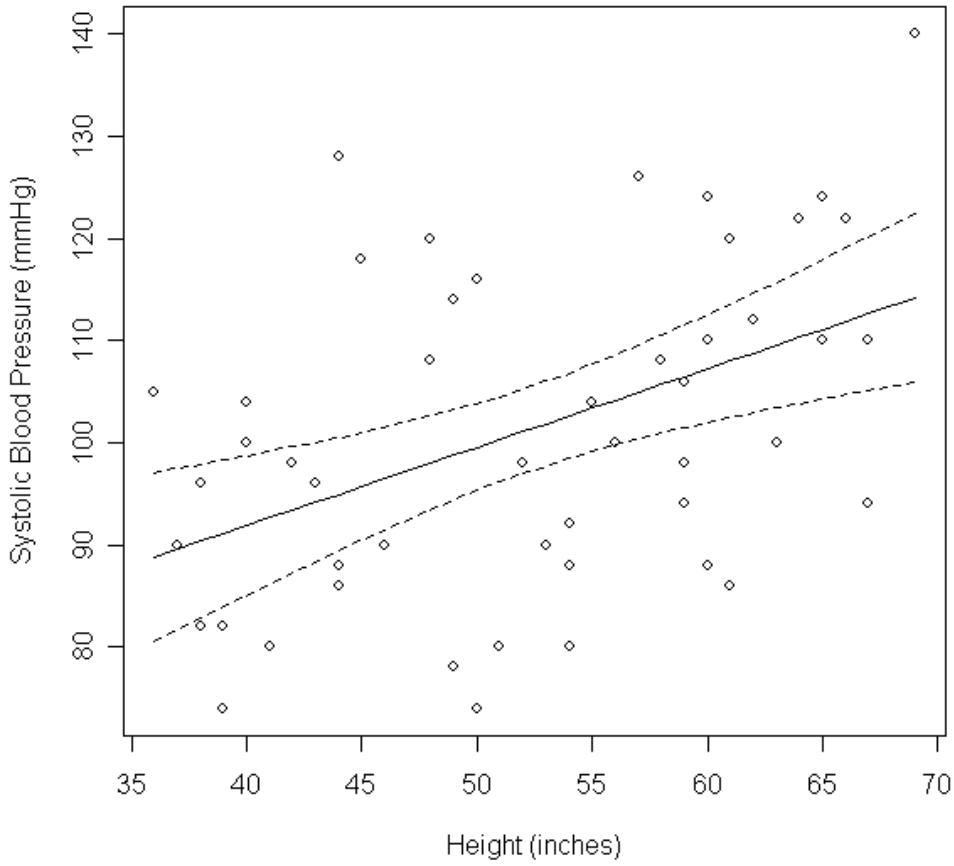
$$\hat{\mu}_{Y|X} \pm t_{n-2,1-\alpha/2} * est.s.e.\left(\hat{\mu}_{Y|X}\right)$$

The **R** commands below, we show the necessary components used to compute the lower and upper 95 percent confidence interval bounds as shown in **Figure 13.7** in the textbook.

R commands:

```
# The square root of the variance of Y given X
syx<- sqrt((sum((sbp-sbpht$fitted.values)**2))/48)

num<-(height-mean(height))**2

den<-sum((height-mean(height))**2)

semeanyx<- syx*sqrt((1/50)+(num/den))

lowci<- sbpht$fitted.values-qt(p=.975, df= 48, lower.tail=TRUE)*semeanyx

upperci<- sbpht$fitted.values+qt(p=.975, df= 48, lower.tail=TRUE)*semeanyx

plot(sbp~height, ylab=("Systolic Blood Pressure (mmHg)"), xlab=("Height
(inches)"))
points(lowci~height, type="l", lty=2)
points(sbpht$fitted.values~height, type="l" ,lty=1)
points(upperci~height, type="l", lty=2)
```

R output:



2. Prediction Intervals for Simple Linear Regression

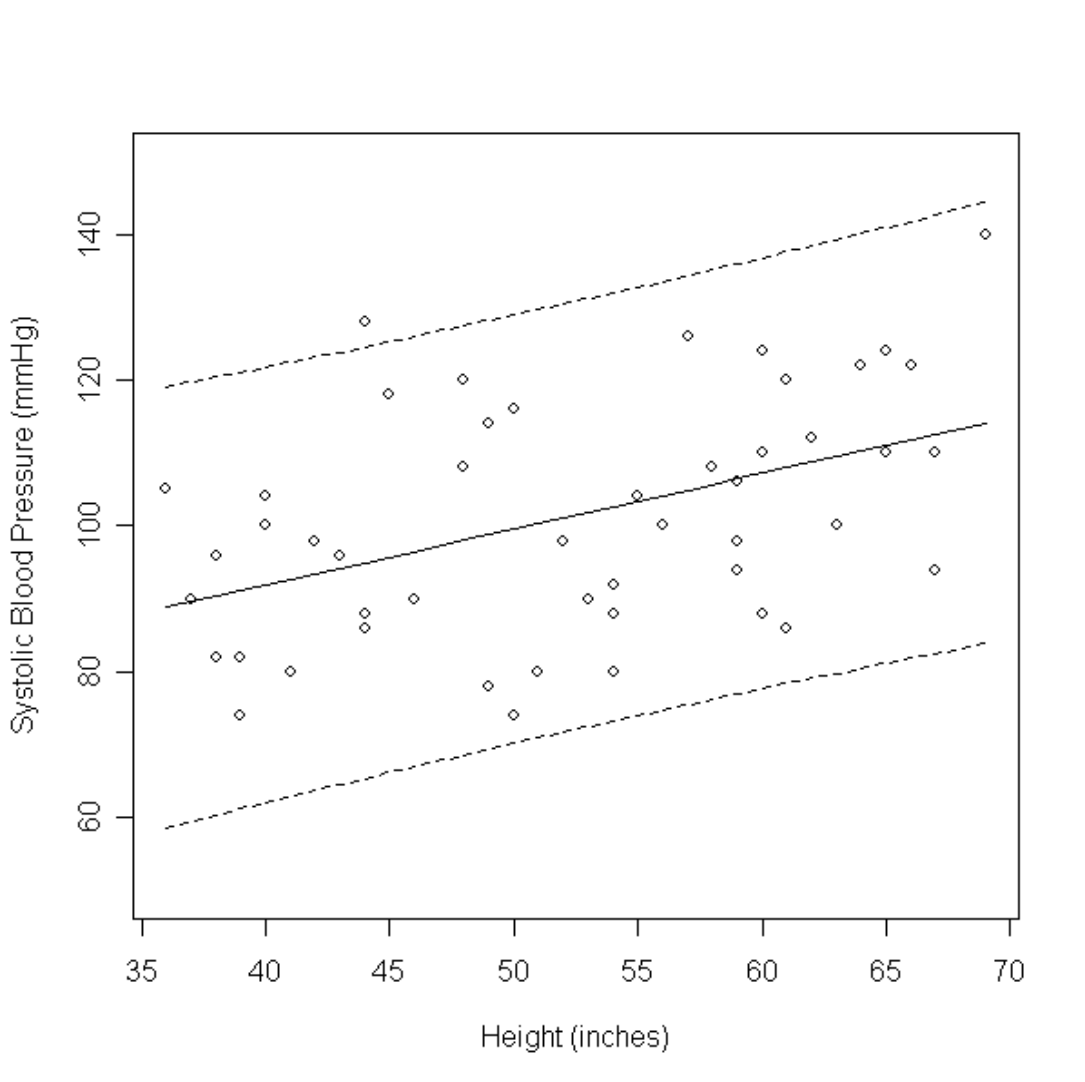
The **R** commands below, we show the necessary components used to compute the lower and upper 95 percent prediction interval bounds as shown in **Figure 13.8** in the textbook.

R commands:

```
# The square root of the variance of Y given X
syx<- sqrt((sum((sbp-sbpht$fitted.values)**2))/48)

num<-(height-mean(height))**2

den<-sum((height-mean(height))**2)

semeanyx<- syx*sqrt(1+(1/50)+(num/den))

lowpi<- sbpht$fitted.values-qt(p=.975, df= 48, lower.tail=TRUE)*semeanyx
```

```
upperpi<- sbpht$fitted.values+qt(p=.975, df= 48, lower.tail=TRUE)*semeanyx

plot(sbp~height, ylim=c(50,150), ylab=("Systolic Blood Pressure (mmHg)"),
xlab=("Height (inches)"))
points(lowpi~height, type="l", lty=2)
points(sbpht$fitted.values~height, type="l" ,lty=1)
points(upperpi~height, type="l", lty=2)
```

R output:



Program Note 13.3 – Multiple Regression Analysis, Variable Selection Procedures, and Residual Plots

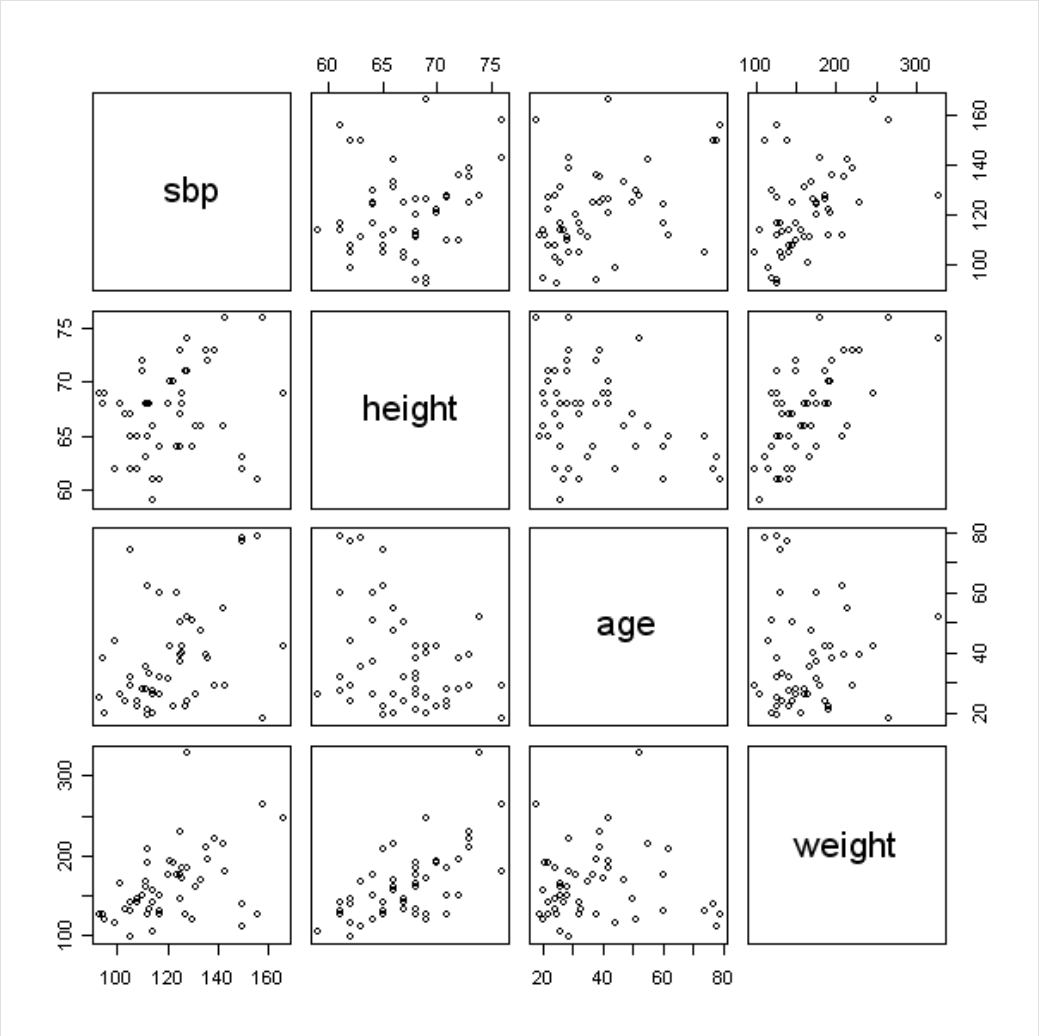
1. Multiple Regression Analysis

See **Figure 13.9 Scatterplot matrix for systolic blood pressure, weight, age, and height** in the textbook.

R commands:

```
# Read in the data from the book's companion website
nhanes<-read.table('http://www.biostat-edu.com/files/Table13-6.txt',head=T)
# Create the scatterplot matrix
pairs(~height+age+weight,data=nhanes)
```

R output:



R commands:

```
#Fitting the full model
lm(sbp~height+age+weight,data=nhanes)
```

R output:

```
Call:
lm(formula = sbp ~ height + age + weight, data = nhanes)

Coefficients:
(Intercept)       height          age       weight
    53.9626       0.3845       0.4381       0.1544
```

R commands:

```
#Fitting the full model
summary(lm(sbp~height+age+weight,data=nhanes))
```

R output:

```
Call:
lm(formula = sbp ~ height + age + weight, data = nhanes)

Residuals:
     Min       1Q   Median       3Q      Max
-27.8208  -7.8398   0.1813   7.4544  28.9848

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 53.96258   41.54436   1.299  0.20045
height       0.38447    0.67246   0.572  0.57028
age          0.43808    0.13192   3.321  0.00176 **
weight       0.15435    0.05969   2.586  0.01294 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.76 on 46 degrees of freedom
Multiple R-Squared: 0.3767,     Adjusted R-squared: 0.336
F-statistic: 9.267 on 3 and 46 DF,  p-value: 6.618e-05
```

2. Variable Selection Procedures

Stepwise methods can be applied using the function **step( )** in **R**. With the option **direction**, the methods that can be specified are “backwards”, “forwards”, or “both”. The option **scope**

defines the range of variables that can be examined. In the first example below, forward stepwise selection is used.

R commands:

```
#Using forward stepwise selection
step(lm(sbp~1,data=nhanes),scope= ~height+age+weight,direction=c("forward"))
```

R output:

```
Start:  AIC=283.67
sbp ~ 1

         Df Sum of Sq     RSS     AIC
+ weight  1    3021.9 10959.0   273.5
+ age     1    2161.9 11818.9   277.3
+ height  1     641.9 13339.0   283.3
<none>                13980.9   283.7

Step:  AIC=273.49
sbp ~ weight

         Df Sum of Sq     RSS     AIC
+ age     1    2182.6  8776.3   264.4
<none>                10959.0   273.5
+ height  1     155.5 10803.4   274.8

Step:  AIC=264.39
sbp ~ weight + age

         Df Sum of Sq    RSS    AIC
<none>                8776.3  264.4
+ height  1      61.9 8714.4  266.0

Call:
lm(formula = sbp ~ weight + age, data = nhanes)

Coefficients:
(Intercept)       weight          age
    77.1846       0.1773       0.4064
```

In the next example below, backward stepwise selection is used.

R commands:

```
#Using backward stepwise selection
step(lm(sbp~height+age+weight,data=nhanes),direction=c("backward"))
```

R output:

```
Start:  AIC=266.04
sbp ~ height + age + weight
```

```
         Df Sum of Sq     RSS     AIC
- height  1      61.9  8776.3   264.4
<none>                8714.4   266.0
- weight  1    1266.8  9981.2   270.8
- age     1    2089.0 10803.4   274.8

Step:  AIC=264.39
sbp ~ age + weight

         Df Sum of Sq     RSS     AIC
<none>                8776.3   264.4
- age     1    2182.6 10959.0   273.5
- weight  1    3042.6 11818.9   277.3

Call:
lm(formula = sbp ~ age + weight, data = nhanes)

Coefficients:
(Intercept)          age       weight
    77.1846       0.4064       0.1773
```

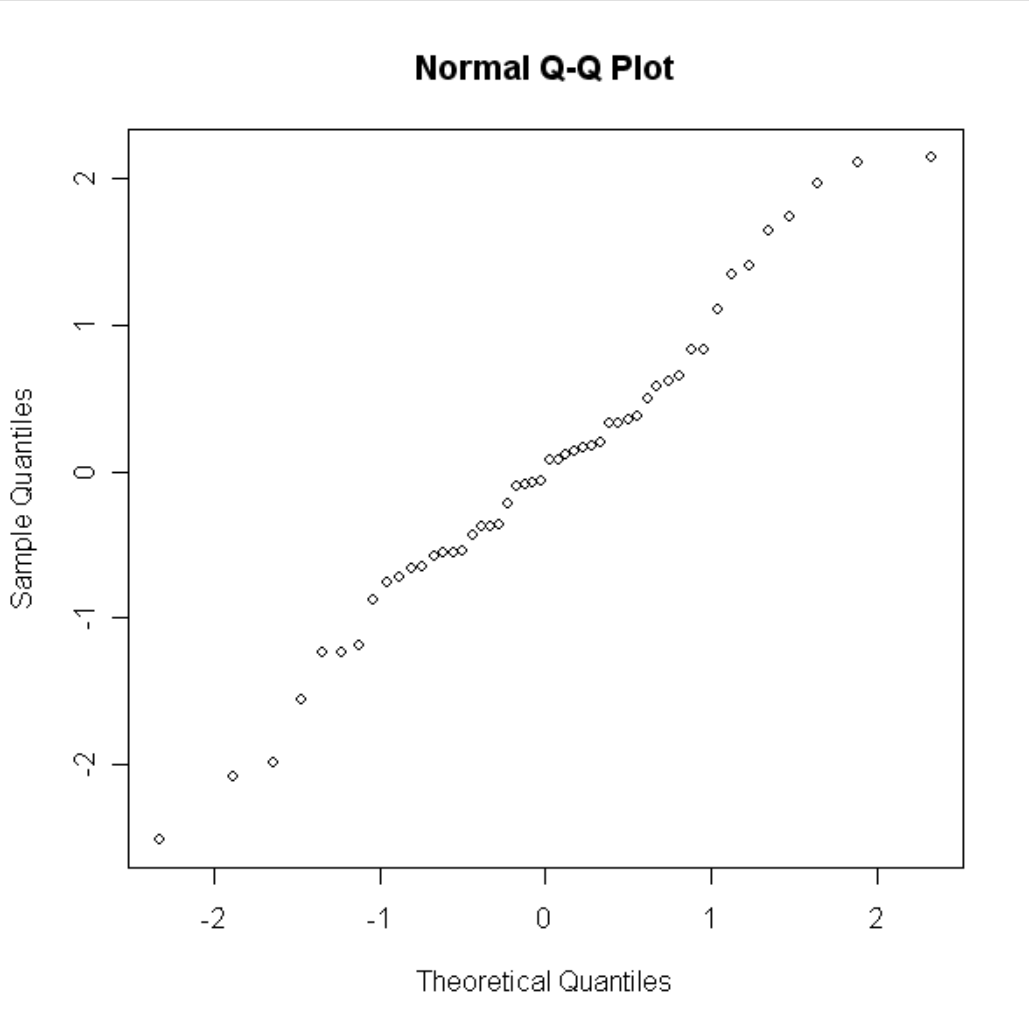
3. Residual Plots

(1) A normal probability plot of the residuals:

R commands:

```
# Read in the data from a directory on your computer
nhanes<-read.table('C:/Table13-6.txt',head=T)
# Model with Age and Weight
model<-lm(sbp~age+weight,data=nhanes)
# Normal Quantile Plot of Standarized Residuals
qqnorm(rstandard(model))
```

R output:

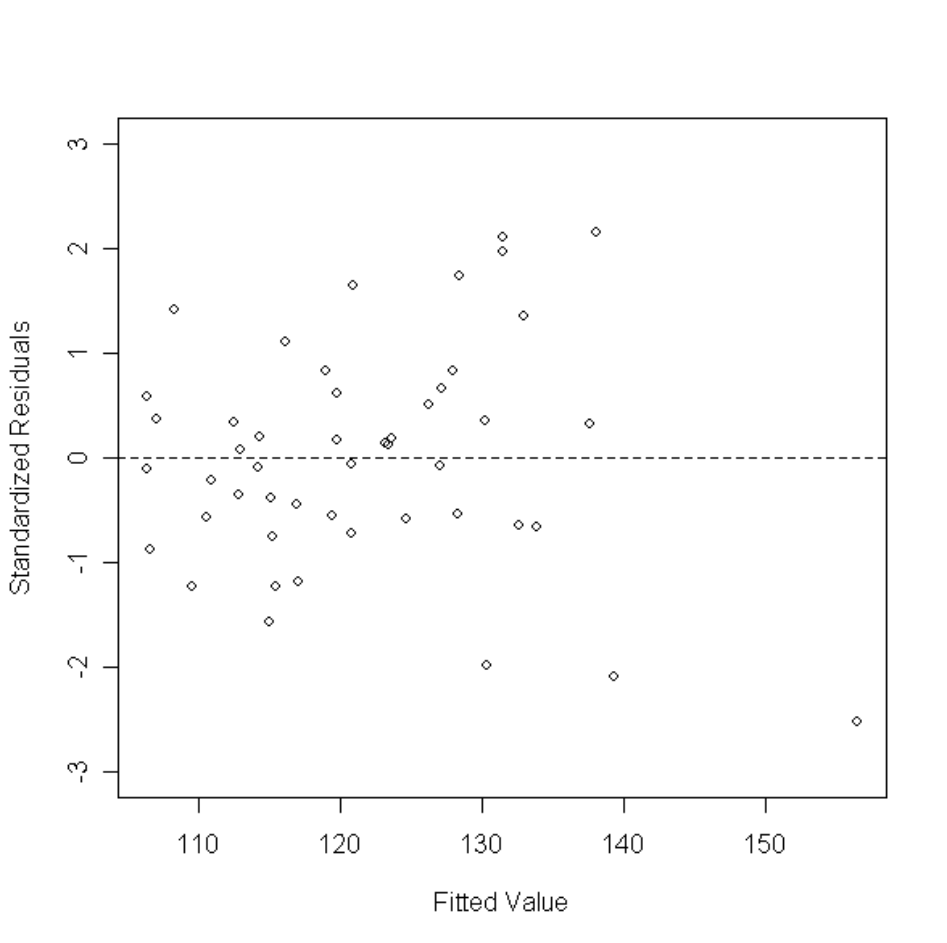


(2) A plot of the residuals against the fitted values:

R commands:

```
# Read in the data from a directory on your computer
nhanes<-read.table('C:/Table13-6.txt',head=T)
# Model with age and weight
lm(sbp~age+weight,data=nhanes)
# Plot of Standardized Residuals versus Fitted Value
plot(fitted.values(model),rstandard(model),ylim=c(-3,3),xlab="Fitted Value",
ylab="Standardized Residuals")
abline(h=0,lty=2)
```

R output:

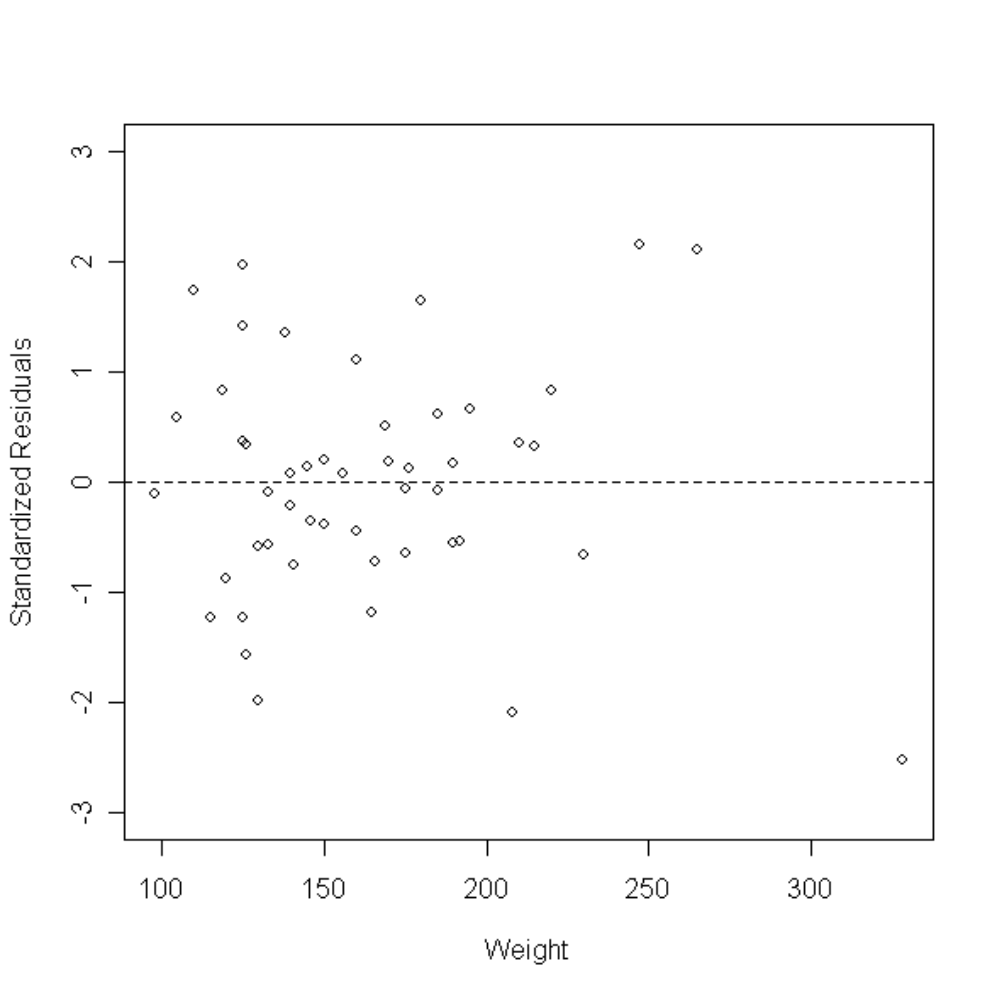


(3) A plot of standardized residuals against each independent variable in the model:

R commands:

```
# Read in the data from a directory on your computer
nhanes<-read.table('C:/Table13-6.txt',head=T)
# Model with age and weight
lm(sbp~age+weight,data=nhanes)
# Plot of Standardized Residuals versus Weight
plot(nhanes$weight,rstandard(model),ylim=c(-3,3), xlab="Weight",
ylab="Standardized Residuals")
abline(h=0,lty=2)
```

R output:



R commands:

```
# Read in the data from a directory on your computer
nhanes<-read.table('C:/Table13-6.txt',head=T)
# Model with age and weight
model<-lm(sbp~age+weight,data=nhanes)
# Plot of Standarized Residuals versus Age
plot(nhanes$age,rstandard(model),ylim=c(-3,3),xlab="Age", ylab="Standardized
Residuals")
abline(h=0,lty=2)
```

R output: