STAT4355HW3

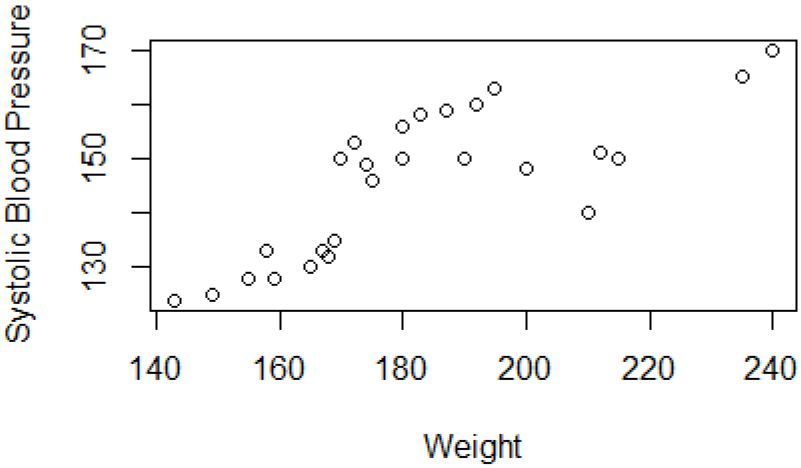
Code ▾

Ramesh Kanakala

a)

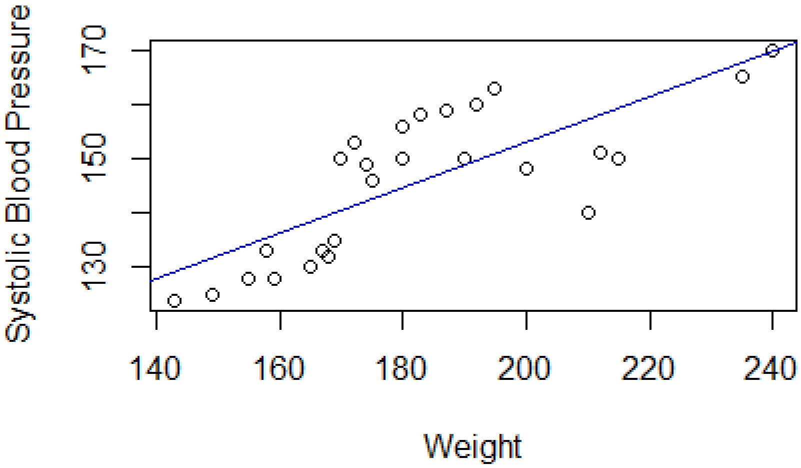
Hide

```
#import data
sysbp=read.csv("hw2_systolic_bp.csv")
y=sysbp$sys.bp
x=sysbp$ï..weight
#plot both data and abline
plot(x, y, xlab="Weight", ylab="Systolic Blood Pressure")
```



Hide

```
lmsb1=lm(y~x)
plot(x, y, xlab="Weight", ylab="Systolic Blood Pressure")
abline(lmsb1, col="blue")
```



b)

Hide

```
#coefficent of determination
rsq <- summary(lmsb1)$r.squared
print(paste("Coefficient of Determination = ", rsq))
```

```
[1] "Coefficient of Determination =  0.59828724501484"
```

Hide

```
#R^2 = SSR/SST => SSR = R^2*SST
sst = sum((y - mean(y))^2)
print(paste("SST = ", sst))
```

```
[1] "SST =  4502.15384615385"
```

Hide

```
ssr = sum((fitted(lmsb1) - mean(y))^2)
print(paste("SSR = ", ssr))
```

```
[1] "SSR =  2693.58122124835"
```

c)

Hide

```
#99% confidence interval on ...
#slope and intercept
confint(lmsb1, level = 0.99)
```

```
                 0.5 %      99.5 %
(Intercept) 32.9955223 105.2132233
x            0.2232037   0.6156267
```

Hide

```
#mean response at x = 160
predict(lmsb1, data.frame(x = 160), interval = "confidence", conf.level = .99)
```

```
       fit      lwr      upr
1 136.2108 131.4268 140.9948
```

Hide

```
#mean response at x = 180
predict(lmsb1, data.frame(x = 180), interval = "confidence", conf.level = .99)
```

```
       fit      lwr      upr
1 144.5991 141.0679 148.1303
```

Hide

```
#mean response at x = 200
predict(lmsb1, data.frame(x = 200), interval = "confidence", conf.level = .99)
```

```
       fit      lwr      upr
1 152.9874 148.6489 157.3259
```

Hide

```
#26 future observations whose weights are identical to the males’ in the dataset
predout <- predict(lmsb1, data.frame(x = x), interval = "predict", predict.level = .99)
predout
```

```
        fit      lwr      upr
1  138.3079 119.8767 156.7390
2  139.1467 120.7530 157.5405
3  144.5991 126.3380 162.8602
4  134.1137 115.4293 152.7982
5  158.0204 139.2672 176.7736
6  142.5020 124.2127 160.7913
7  148.7933 130.5026 167.0839
8  157.1816 138.4924 175.8708
9  152.9874 134.5532 171.4216
10 131.5972 112.7091 150.4854
11 135.3720 116.7750 153.9690
12 139.9855 121.6247 158.3464
13 140.4050 122.0589 158.7510
14 141.2438 122.9238 159.5637
15 135.7914 117.2214 154.3614
16 139.5661 121.1894 157.9429
17 142.0826 123.7842 160.3810
18 145.8574 127.5995 164.1152
19 159.2786 140.4215 178.1358
20 150.8903 132.5421 169.2386
21 144.5991 126.3380 162.8602
22 129.0807 109.9516 148.2099
23 169.7640 149.6932 189.8349
24 167.6669 147.8858 187.4481
25 149.6321 131.3218 167.9424
26 147.5350 129.2653 165.8047
```

d)

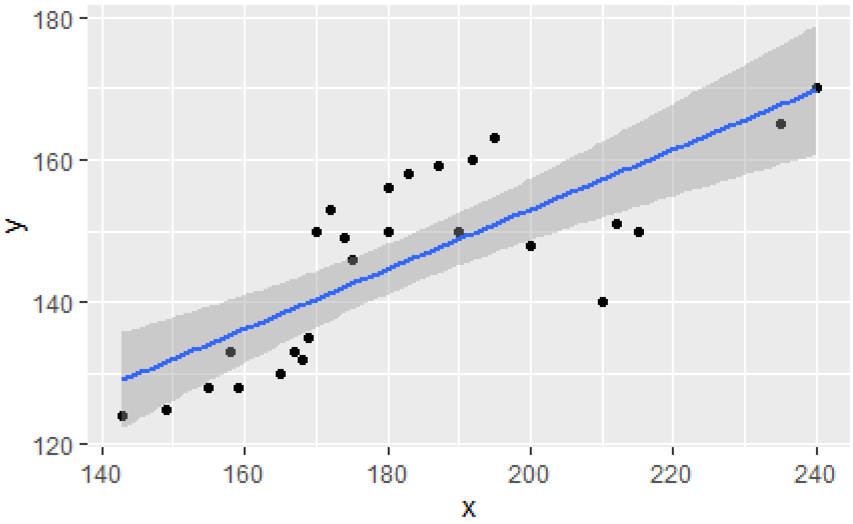
Hide

```
#merge data points and prediction outcome object
pred.bp <- cbind(sysbp, predout)
library("ggplot2")
```

```
Use suppressPackageStartupMessages() to eliminate
package startup messages
```

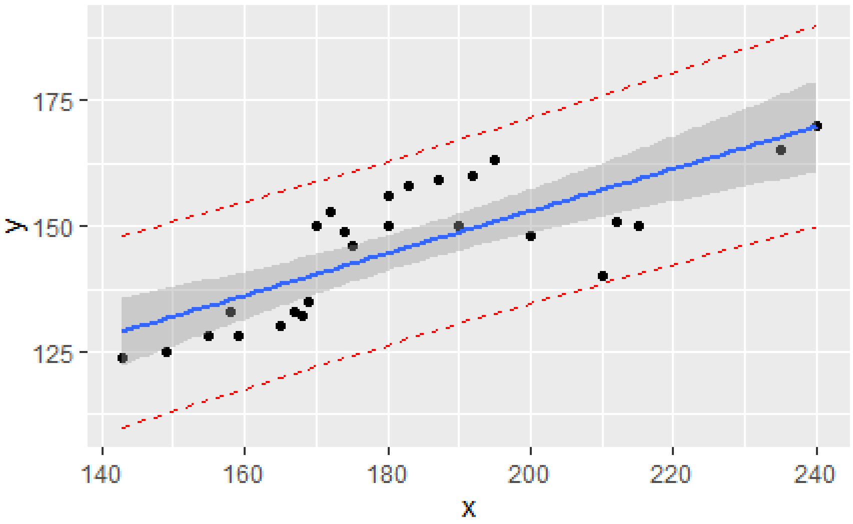
Hide

```
#confidence interval curve
p <- ggplot(pred.bp, aes(x, y)) + geom_point() + stat_smooth(method = lm)
p
```



Hide

```
#prediction interval curve
p + geom_line(aes(y = lwr), color = "red", linetype = "dashed")+
geom_line(aes(y = upr), color = "red", linetype = "dashed")
```


175
150
125
y
140
160
180
200
220
240
x