Fall Week 1 Lab

CSCI E-106 TA Session

9/12/2019

Contents

d. The manufacturer has suggested that the mean required time should not increase by more than 14 minutes for each additional copier that is serviced on a service call. Conduct a test to decide whether this standard is being satisfied by Tri-City. Control the risk of a Type I error at .05. State the alternatives, decision rule, and conclusion. What is the $P - value$ of the test? . . . 14
e. Does b_0 give any relevant information here about the “start-up” time on calls-i.e., about the time required before service work is begun on the copiers at a customer location? 15

Slide 17 - Vectors

```
x <- c(1, 2, 3)
y <- seq(1:3)
z <- rep(3, 14)

print(x)
```

```
## [1] 1 2 3
```

```
print(y)
```

```
## [1] 1 2 3
```

```
print(z)
```

```
##  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3
```

Slide 20 - Operations on Vectors: Example

```
x <- c(1,3,5,10)
x <- x * 2
print(x)
```

```
## [1]  2  6 10 20
```

Slide 21 - Lists

```
hd<- list(name="John Smith", id=1111, grade="B+", age=35)
hd
```

```
## $name
## [1] "John Smith"
##
## $id
## [1] 1111
##
```

```
## $grade
## [1] "B+"
##
## $age
## [1] 35
```

```r
#find the second component of the list
hd[[2]]
```

```
## [1] 1111
```

Slide 25 - Data Frame

```r
students <- c("John", "Mary", "Ethan", "Dora")
score<- c(76, 82, 84, 67)
grade <- c("B", "A", "A", "C")
class <- data.frame(students, score, grade)

class
```

```
##   students score grade
## 1     John    76     B
## 2     Mary    82     A
## 3    Ethan    84     A
## 4     Dora    67     C
```

Slide 27 - Loading Data into R

```r
#see where R will read and save files
getwd()
```

```
## [1] "/cloud/project"
```

Slide 32 - The For() Loop

```r
x = 1:10

for(i in 1:5)
{
    x[i] <- x[i+2] + x[i+1]
}

print(x)
```

```
##  [1]  5  7  9 11 13  6  7  8  9 10
```

```r
# Example: Write for loop function that calculates the cumulative total for each row or element

# declare a matrix

m = matrix(1:3, nrow=3, ncol=5)
print(m)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    1    1    1
## [2,]    2    2    2    2    2
## [3,]    3    3    3    3    3
```

```r
cumSumRow = rep(0, dim(m)[1])

for (i in (1:dim(m)[1]))
{
    for (j in (1:length(m[i,])))
    {
        cumSumRow[i] = (cumSumRow[i] + m[i,j])
    }

}

print(cumSumRow)
```

```
## [1]  5 10 15
```

Slide 35 - Sorting

```r
data(mtcars)

# sort by mpg
newdata = mtcars[order(mtcars$mpg),]
print(newdata)
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
```

```
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

```r
# sort by mpg and cyl
newdata <- mtcars[order(mtcars$mpg, mtcars$cyl),]
print(newdata)
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

```r
#sort by mpg (ascending) and cyl (descending)
newdata <- mtcars[order(mtcars$mpg, -mtcars$cyl),]
print(newdata)
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
```

Slide 40 - Example

```r
set.seed(123)

obs = rnorm(10000)

print(quantile(obs, .99))
```

```
##      99%
## 2.338748
```

1.20. Copier maintenance.

The Tri-City Office Equipment Corporation sells an imported copier on a franchise basis and performs preventive maintenance and repair service on this copier. The data below have been collected from 45 recent calls on users to perform routine preventive maintenance service; for each call, X is the number of copiers serviced and Y is the total number of minutes spent by the service person. Assume that first-order regression model (1.1) is appropriate.

Loading Data

```
# Assigning the data to a data frame object
# called df20 and then we want to name the columns
df20 = read.delim("CH01PR20.txt", header=FALSE, sep="")
colnames(df20) = c("y", "x")
```

a. Obtain the estimated regression function.

Solution

```
lmFit20 = lm(y~x, df20)
print(summary(lmFit20))
```

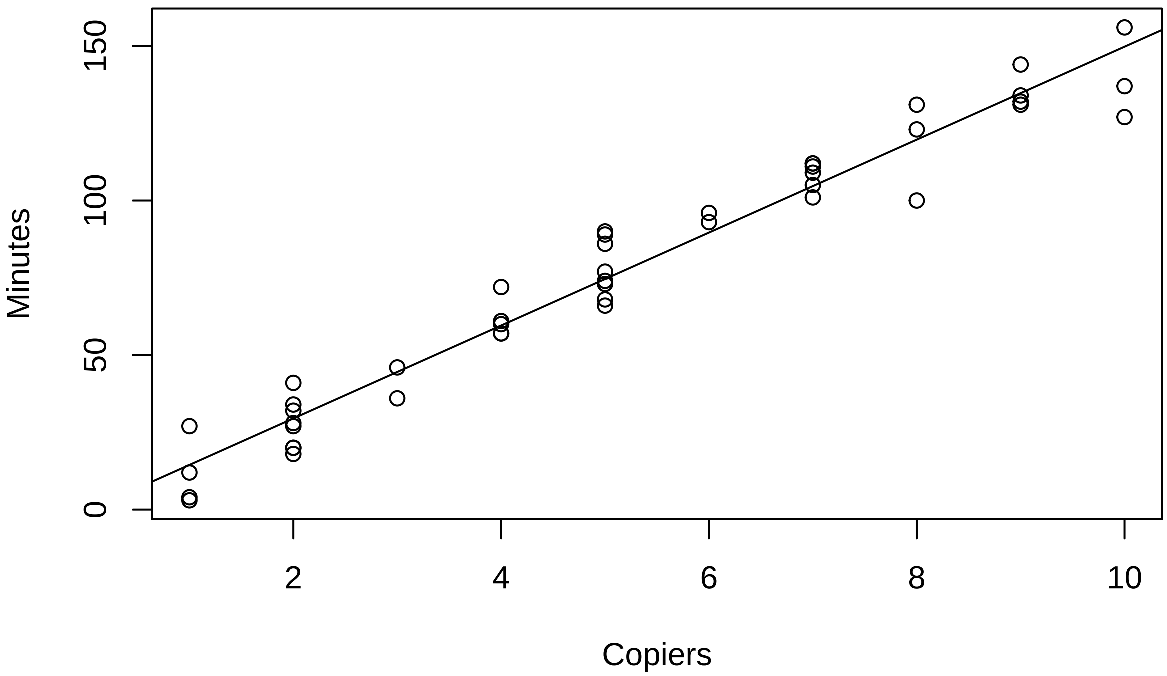
```
##
## Call:
## lm(formula = y ~ x, data = df20)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7723  -3.7371   0.3334   6.3334  15.4039
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5802     2.8039  -0.207    0.837
## x            15.0352     0.4831  31.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 43 degrees of freedom
## Multiple R-squared:  0.9575, Adjusted R-squared:  0.9565
## F-statistic: 968.7 on 1 and 43 DF,  p-value: < 2.2e-16
```

We see that to obtain our estimated regression function we would use the estimate column values so our answer would become: y = -.5802 + 15.0352x

b. Plot the estimated regression function and the data. How well does the estimated regression function fit the data?

Solution

```
plot(df20$x, df20$y, xlab="Copiers", ylab="Minutes")
abline(lmFit20)
```



By just looking at our data we do have a well fit line here.

c. Interpret b_o in your estimated regression function. Does b_o provide any relevant information here? Explain.

Solution

```r
yHat = predict(lmFit20, data.frame(x=0))
print(yHat)
```

```
##          1
## -0.5801567
```

b_0 does not give us any sensible information since the predicted value is a negative number

d. Obtain a point estimate of the mean service time when $X = 5$ copiers are serviced.

Solution

```r
yHat = predict(lmFit20, data.frame(x=5))
print(yHat)
```

```
##        1
## 74.59608
```

1.24. Refer to *Copier maintenance* Problem 1.20.

a Obtain the residuals e_i and the sum of the squared residuals $\sum e_i^2$. What is the relation between the sum of the squared residuals here and the quantity Q in (1.8)?

Solution

```
yHat = predict(lmFit20)

resids = (df20$y - yHat)
print(resids)
```

```
##          1          2          3          4          5          6
##  -9.4903394  0.4391645  1.4744125 11.5096606 -2.4550914 -12.7723238
##          7          8          9         10         11         12
##  -6.5960836 14.4039164 -10.4550914  2.5096606  9.2629243  6.2276762
##         13         14         15         16         17         18
##   3.3686684 -8.5255875 12.4391645 -19.7018277  0.3334204 11.2981723
##         19         20         21         22         23         24
## -22.7723238 -2.5608355 -8.5960836 -3.6665796  4.3334204 -0.5960836
##         25         26         27         28         29         30
##  -0.7370757  7.3334204 -11.4903394 -1.5960836  6.3334204  6.3686684
##         31         32         33         34         35         36
##   3.2981723 15.4039164 -9.4903394 -1.4903394 -11.4550914 -2.5608355
##         37         38         39         40         41         42
##  11.4039164 -2.7370757  7.3334204 12.5449086 -3.7370757  4.5096606
##         43         44         45
##  -2.4903394  1.4391645  2.4039164
```

```
# Same thing, different syntax
# resids = lmFit20$residuals
# print(resids)

#Sum of the squared residuals
SSE = (sum(resids^2))
print(SSE)
```

```
## [1] 3416.377
```

b. Obtain point estimates of σ^2 and σ. In what units is σ expressed?

Solution

```
# Degrees of freedom residual
dfResid = lmFit20$df.residual
MSE = (SSE/dfResid)
print(MSE)
```

```
## [1] 79.45063
```

```
# Obtain original units
sqrt(MSE)
```

```
## [1] 8.913508
```

```r
# We can also see sigma ^2 by using the anova function
anova(lmFit20)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1  76960   76960  968.66 < 2.2e-16 ***
## Residuals 43   3416      79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#mean is expressed in the units of minutes.
```

1.27. Muscle mass.

A person's muscle mass is expected to decrease with age. To explore this rela- tionship in women, a nutritionist randomly selected 15 women from each lO-year age group, beginning with age 40 and ending with age 79. The results follow; X is age, and Y is a measure of muscle mass. Assume that first-order regression model (1.1) is appropriate.

Loading Data

```r
df27 = read.delim("CH01PR27.txt", header=FALSE, sep="")
colnames(df27) = c("y", "x")
```

a. Obtain the estimated regression function. Plot the estimated regression function and the data. Does a linear regression function appear to give a good fit here? Does your plot support the anticipation that muscle mass decreases with age?

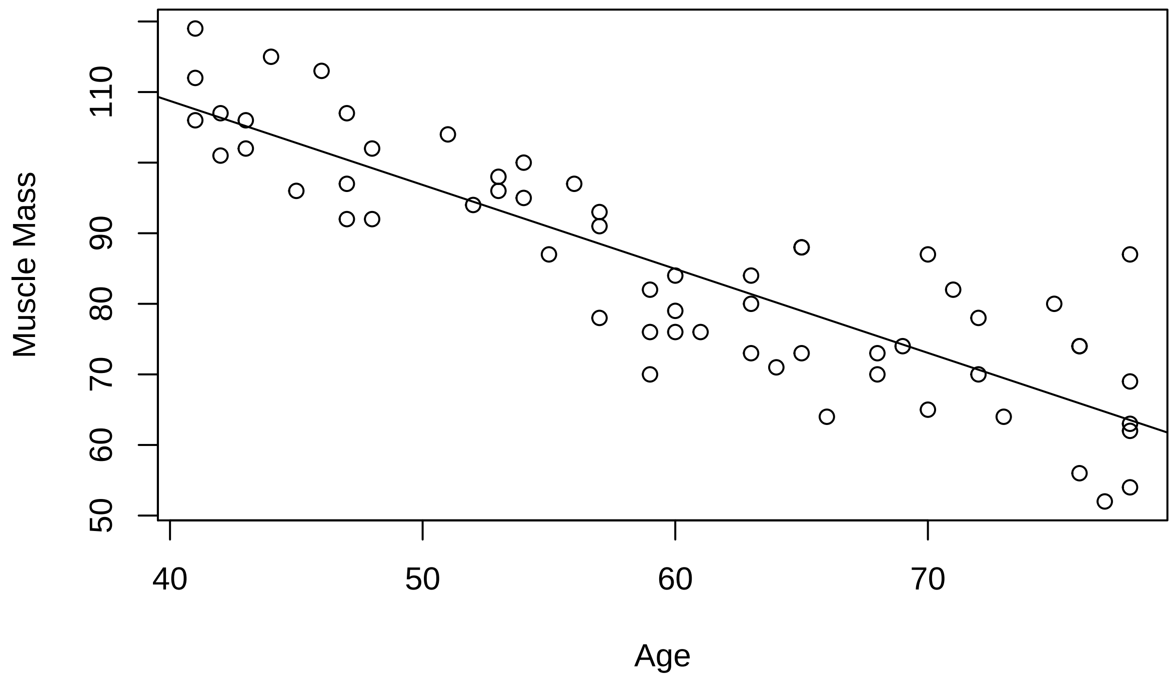
Solution

```r
lmFit27 = lm(y~x, df27)
print(summary(lmFit27))
```

```
##
## Call:
## lm(formula = y ~ x, data = df27)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1368  -6.1968  -0.5969   6.7607  23.4731
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 156.3466     5.5123   28.36   <2e-16 ***
## x            -1.1900     0.0902  -13.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 8.173 on 58 degrees of freedom
## Multiple R-squared:  0.7501, Adjusted R-squared:  0.7458
## F-statistic: 174.1 on 1 and 58 DF,  p-value: < 2.2e-16
```

```r
plot(df27$x, df27$y, xlab="Age", ylab="Muscle Mass")
abline(lmFit27)
```



So $y = 156.34 - 1.19x$ Assessing the plot, it does appear to support the anticipation that muscle mass decreases with age.

b. Obtain the following: (1) a point estimate of the difference in the mean muscle mass for women differing in age by one year, (2) a point estimate of the mean muscle mass for women aged $X = 60$ years, (3) the value of the residual for the eighth case, (4) a point estimate of σ^2.

Solution

```r
# 1
beta1 = lmFit27$coefficients[2]
print(beta1)
```

```
##         x
## -1.189996
```

```r
# 2
yHat = predict(lmFit27, data.frame(x=60))
print(yHat)
```

```
##        1
## 84.94683
```

```r
# 3
yHat = predict(lmFit27)
```

```r
resids = (df27$y - yHat)
print(resids)
```

```
##           1           2           3           4           5           6
##   0.8232429  -1.5567482  -3.4167751  11.3932294  -6.7967661  11.4432518
##           7           8           9          10          11          12
##  -8.4167751   4.4432518  -7.2267796   2.7732204   0.6332473   6.5832249
##          13          14          15          16          17          18
##  -3.1767571  11.0132384  -5.3667527  -3.8968110   2.4831800   7.2931845
##          19          20          21          22          23          24
##  -4.1368289 -10.5168200   2.9131935   4.7231980  -0.4667975   2.7231980
##          25          26          27          28          29          30
##   7.9131935  -0.9468334 -16.1368289   8.3432070 -10.1368289   4.4831800
##          31          32          33          34          35          36
##  -2.4268693  -8.3768469  -8.9468334  -1.3768469   2.6231531  -9.1868514
##          37          38          39          40          41          42
## -13.8068603   9.0031442  -5.9468334   9.0031442  -5.9968558  -0.2368738
##          43          44          45          46          47          48
##  -7.7568379  13.9531217  -5.4268693   5.4730858  -9.5269142  -1.5269142
##          49          50          51          52          53          54
##   7.3331128  -8.0468783  -5.4768917   8.0930948  23.4730858  -0.5269142
##          55          56          57          58          59          60
##  10.1431172  12.9030993 -12.7169097  -9.9069052  -0.6668872   8.0930948
```

```r
# Same thing, different syntax
# resids = lmFit27$residuals
# print(resids)

print(resids[8])
```

```
##        8
## 4.443252
```

```r
# 4
# Degrees of freedom residual
dfResid = lmFit27$df.residual
SSE = (sum(resids^2))

MSE = (SSE/dfResid)
print(MSE)
```

```
## [1] 66.80082
```

2.05. Reference to Copier maintenance

The Tri-City Office Equipment Corporation sells an imported copier on a franchise basis and performs preventive maintenance and repair service on this copier. The data below have been collected from 45 recent calls on users to perform routine preventive maintenance service; for each call, X is the number of copiers serviced and Y is the total number of minutes spent by the service person. Assume that first-order regression model (1.1) is appropriate.

```r
# Load data
df20 = read.delim("CH01PR20.txt", header=FALSE, sep="")
colnames(df20) = c("y", "x")
```

a. Estimate the change in the mean service time when the number of copiers serviced increases by one. Use a 90 percent confidence interval. Interpret your confidence interval.

```r
# Create linear model from data
lmFit20 = lm(y~x, df20)
print(summary(lmFit20))
```

```
##
## Call:
## lm(formula = y ~ x, data = df20)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.7723  -3.7371   0.3334   6.3334  15.4039
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5802     2.8039  -0.207    0.837
## x            15.0352     0.4831  31.123   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.914 on 43 degrees of freedom
## Multiple R-squared:  0.9575, Adjusted R-squared:  0.9565
## F-statistic: 968.7 on 1 and 43 DF,  p-value: < 2.2e-16
```

So the estimated change in the mean service time when the number of copiers serviced increases by one is: 15.0352 which is the estimated slope (b_1) and the s(b_1) is .4831

```r
#Finding our t-distribution
#In this case we want to first get our degrees of freedom which we can see from our summary is: 43
#or we can use our anova function to find the same
anova(lmFit20)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df Sum Sq Mean Sq F value    Pr(>F)
## x          1  76960   76960  968.66 < 2.2e-16 ***
## Residuals 43   3416      79
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#We can find our t-distribution with our degrees of freedom
tdist = qt(1-.10/2, 43)
print(tdist)
```

```
## [1] 1.681071
```

```r
#Use a 90 Percent Confidence Interval
confint(lmFit20, level=.9)
```

```
##                   5 %      95 %
## (Intercept) -5.29378  4.133467
## x           14.22314 15.847352
```

```r
#or you can write it all out without a function:
c(15.0352-1.6811*0.4831, 15.0352+1.6811*0.4831)
```

```
## [1] 14.22306 15.84734
```

So our b_1 interval would be: 14.223 <= b_1 <= 15.847 90% of the time. with t(.95,43): 1.681

b. Conduct a t test to determine whether or not there is a linear association between X and Y here; control the a risk at $.10$. State the alternatives, decision rule, and conclusion. What is the $P - value$ of your test?

H_0: $B_1 = 0$, H_a: $B_1 != 0

So our test statistic is: t = b_1 / SD(b_1)= (15.0352 - 0) / .4831 Or you can see it above under t value of x

```r
t = ( 15.0352 - 0 ) / .4831
print(t)
```

```
## [1] 31.12233
```

The decision rule would be: reject H_0 if $t > 1.681$ or reject H_0 if the p-value $< .1$

The conclusion: Reject the hypothesis

P-Value: P($t_4$3 > 31.123) < 0.000001 or as seen in our summary: 2.2e-16

```r
#Calculate the p-value
p = 1-pt(31.122,43)
print(p)
```

```
## [1] 0
```

c. Are your results in parts (a) and (b) consistent? Explain.

Yes they are consistent because we see that the 90% confidence interval of b_1 doesn't include 0 so the hypothesis that $b_1 = 0$ at a 10% sig. level will be rejected.

d. The manufacturer has suggested that the mean required time should not increase by more than 14 minutes for each additional copier that is serviced on a service call. Conduct a test to decide whether this standard is being satisfied by Tri-City. Control the risk of a Type I error at $.05$. State the alternatives, decision rule, and conclusion. What is the $P - value$ of the test?

H_0: $B_1 <= 14$, H_a: $B_1 > 14

So our test statistic is: t = b_1 / SD(b_1)= (15.0352 - 14) / .4831

The decision rule would be: reject H_0 if $t > 1.681$ or reject H_0 if the p-value $< .05$

```r
#test statistic:
t1 = ( 15.0352 - 14 ) / .4831
print(t1)
```

```
## [1] 2.142828
```

The conclusion: Reject the hypothesis

```r
#p-value calculation:
p = 1 - pt(2.148,43)
print(p)
```

```
## [1] 0.01869529
```

P-Value: P($t_43 > 2.1428$) = 0.0189

e. Does b_0 give any relevant information here about the "start-up" time on calls- i.e., about the time required before service work is begun on the copiers at a customer location?

No b_0 does not give any relevant information since it is negative and would not provide anything meaningful.