Stat Computing HW1

Michael Ackerman

September 8th

```
gender <- c('M','M','F','M','F','F','M','F','M')
age <- c(34, 64, 38, 63, 40, 73, 27, 51, 47)
smoker <- c('no','yes','no','no','yes','no','no','no','yes')
exercise <- factor(c('moderate','frequent','some','some','moderate','none','none','moderate','moderate'),
                    levels=c('none','some','moderate','frequent'), ordered=TRUE
)
los <- c(4,8,1,10,6,3,9,4,8)
x <- data.frame(gender, age, smoker, exercise, los)
x
```

```
##   gender age smoker exercise los
## 1      M  34     no moderate   4
## 2      M  64    yes frequent   8
## 3      F  38     no     some   1
## 4      M  63     no     some  10
## 5      F  40    yes moderate   6
## 6      F  73     no     none   3
## 7      M  27     no     none   9
## 8      F  51     no moderate   4
## 9      M  47    yes moderate   8
```

Model:

```
model <- lm(los ~ gender + age + smoker + exercise, dat=x)
```

1) It appears that has being a male has the greatest effect on your length of stay with a value of 4.509.

2)

```
mod <- lm(los~gender)
summary(mod)
```

```
##
## Call:
## lm(formula = los ~ gender)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -3.8   -0.5    0.2    1.2    2.5
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.500      1.099   3.186   0.0154 *
## genderM        4.300      1.474   2.917   0.0224 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 2.197 on 7 degrees of freedom
## Multiple R-squared:  0.5487, Adjusted R-squared:  0.4842
## F-statistic:  8.51 on 1 and 7 DF,  p-value: 0.02243
```

3)

```
mod$coef
```

```
## (Intercept)     genderM
##         3.5         4.3
```

4)

```
sqrt(diag(vcov(mod)))
```

```
## (Intercept)     genderM
##    1.098701    1.474061
```

5)

```
mod <- lm(los ~ gender, dat=x)
mod.c <- coef(summary(mod))
teststat <- mod.c[,1]/mod.c[,2]
2*pt(teststat['genderM'], 7, lower.tail = FALSE)
```

```
##    genderM
## 0.02243214
```

6)

```
fitted(mod)-predict(mod)
```

```
##             1             2             3             4             5
## -5.329071e-15 -1.776357e-15 -1.332268e-15 -1.776357e-15 -1.332268e-15
##             6             7             8             9
## -1.332268e-15 -1.776357e-15 -1.332268e-15 -1.776357e-15
```

It looks like the functions work a tiny bit different, yet create essentially the same vector. **7)**

```
newdata <- data.frame(gender=c('F','M','F'))
predict(mod,newdata)
```

```
##   1   2   3
## 3.5 7.8 3.5
```

8)

```
residuals <- fitted(mod)- x$los
residuals
```

```
##    1    2    3    4    5    6    7    8    9
##  3.8 -0.2  2.5 -2.2 -2.5  0.5 -1.2 -0.5 -0.2
```

9)

```
residuals(mod)
```

```
##    1    2    3    4    5    6    7    8    9
## -3.8  0.2 -2.5  2.2  2.5 -0.5  1.2  0.5  0.2
```

Same result in absolute value.

10)

```
sum(residuals^2) - deviance(mod)
```

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

These functions do the same thing.

11)

```
sqrt(deviance(mod)/df.residual(mod))
```

```
## [1] 2.197401
```

This is the same as the Residual Standard Error shown by the original model.

12)

```
xmen <- x[gender== "M",]
women <- x[gender=="F",]
```

X-Men is the proper labeling.

13)

```
var(xmen$los)
```

```
## [1] 5.2
```

```
var(women$los)
```

```
## [1] 4.333333
```

14)

```r
t.test(women$los, xmen$los)
```

```
##
##  Welch Two Sample t-test
##
## data:  women$los and xmen$los
## t = -2.9509, df = 6.8146, p-value = 0.02205
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.7647486 -0.8352514
## sample estimates:
## mean of x mean of y
##       3.5       7.8
```

```r
t.test(women$los, xmen$los, var.equal = T)
```

```
##
##  Two Sample t-test
##
## data:  women$los and xmen$los
## t = -2.9171, df = 7, p-value = 0.02243
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.7856014 -0.8143986
## sample estimates:
## mean of x mean of y
##       3.5       7.8
```

The equal variance assumption causes that p-value in the t-test comparison to be the same as the full model with the genderM variable.

```r
##this is nice and consice.
t.test(los ~ gender, dat=x, var.equal=TRUE)
```

```
##
##  Two Sample t-test
##
## data:  los by gender
## t = -2.9171, df = 7, p-value = 0.02243
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -7.7856014 -0.8143986
## sample estimates:
## mean in group F mean in group M
##             3.5             7.8
```