Assignment 1

Ya-Chen Lin

Create a Data Set

```
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
```

Question 1:

Looking at the output, which coefficient seems to have the highest effect on los?

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

```
##
## Call:
## lm(formula = los ~ gender + age + smoker + exercise, data = x)
##
## Coefficients:
## (Intercept)      genderM          age    smokeryes   exercise.L
##    0.588144     4.508675     0.033377     2.966623    -2.749852
##  exercise.Q   exercise.C
##   -0.710942     0.002393
```

Based on the coeffecient, genderM seems to have the highest effect on los since the coeffecient value is the highest. Q1.Create a model using [los] and [gender] and assign it to the variable mod. Run the summary function with mod as its argument.

```
mod <- lm (los ~ gender, dat = x)
summary (mod)
```

```
##
## Call:
## lm(formula = los ~ gender, data = x)
##
## 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
```

Question 1:

What is the estimate for the intercept? What is the estimate for gender? Use the [coef] function.

```r
coef(mod)
```

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

The estimate for the intercept is 3.5 and the estimate for gender is 4.3.

Question 2:

The second column of coef are standard errors. These can be calculated by taking the sqrt of the diag of the vcov of the summary of mod. Calculate the standard errors.

```r
sqrt(diag(vcov(summary(mod))))
```

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

The standard errors for intercept is 1.098701 and the standard error for genderM is 1.474061 The third column of coef are test statistics. These can be calculated by dividing the first column by the second column.

```r
mod <- lm(los ~ gender, dat=x)
mod.c <- coef(summary(mod))
mod.c[,1]/mod.c[,2]
```

```
## (Intercept)     genderM
##    3.185581    2.917110
```

Question 3:

Use the pt function to calculate the p value for gender. The first argument should be the test statistic for gender. The second argument is the degrees-of-freedom. Also, set the lower.tail argument to FALSE. Finally multiple this result by two.

```
ttest <- (mod.c[,1]/mod.c[,2])
pvalue <- pt(ttest, df =7, lower.tail= FALSE)
print(2*pvalue)
```

```
## (Intercept)     genderM
##  0.01537082  0.02243214
```

Predicted Values
The estimates can be used to create predicted values.

```
3.5+(x$gender=='M')*4.3
```

```
## [1] 7.8 7.8 3.5 7.8 3.5 3.5 7.8 3.5 7.8
```

Question 1:

It is even easier to see the predicted values by passing the model mod to the predict or fitted functions. Try it out.
[predict]

```
predict(mod)
```

```
##   1   2   3   4   5   6   7   8   9
## 7.8 7.8 3.5 7.8 3.5 3.5 7.8 3.5 7.8
```

[fitted]

```
fitted(mod)
```

```
##   1   2   3   4   5   6   7   8   9
## 7.8 7.8 3.5 7.8 3.5 3.5 7.8 3.5 7.8
```

Yes. These two functions are easier to see the predicted values.

Question 2:

predict can also use a new data set. Pass newdat as the second argument to predict.

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

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

Question 1:

Use one of the methods to generate predicted values. Subtract the predicted value from the x$los column.

```
prevalue <- predict(mod)
x$los - prevalue
```

```
##    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
```

Question 2:

Try passing mod to the residuals function

```
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
```

Question 3:

Square the residuals, and then sum these values. Compare this to the result of passing mod to the deviance function.

```
rvalue <- residuals(mod)
sum((rvalue)^2)
```

```
## [1] 33.8
```

```
deviance(mod)
```

```
## [1] 33.8
```

```
sum((rvalue)^2) == deviance(mod)
```

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

As we can see from the result, the values from squaring the residuals and sum the values are exactly same as directly using deviance function. We can judge either by numerically the same (33.8) or if we set the two functions equal to each other and we get TRUE as the result.

```
df.residual(mod)
```

```
## [1] 7
```

Question 1:

Calculate standard error by dividing the deviance by the degrees-of-freedom, and then taking the square root. Verify that this matches the output labeled “Residual standard error” from summary(mod).

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

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

```
summary(mod)
```

```
##
## Call:
## lm(formula = los ~ gender, data = x)
##
## 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
```

As we can see that the standard error we calculated is 2.197401 (can be round up to 2.197) and in the summary the residual standard error is 2.197. Thus, these two values are the same.

```
predict(mod, se.fit=TRUE)$residual.scale
```

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

And the number also matches the output above.

Question 1:

Create a subset of x by taking all records where gender is ‘M’ and assigning it to the variable men. Do the same for the variable women.

```
men <- subset(x, gender == "M", select = los)
women <- subset(x, gender == 'F', select = los)
```

Question 1:

By default a two-sampled t-test assumes that the two groups have unequal variances. You can calculate variance with the var function. Calculate variance for los for the men and women data sets.

```r
var(men)
```

```
##     los
## los 5.2
```

```r
var(women)
```

```
##          los
## los 4.333333
```

The variance for los function for men is 5.2 and the variance for los function for women is approximately 4.333

Question 1:

Call the t.test function, where the first argument is los for women and the second argument is los for men. Call it a second time by adding the argument var.equal and setting it to TRUE. Does either produce output that matches the p value for gender from the model summary?

```r
t.test(women, men)
```

```
##
##  Welch Two Sample t-test
##
## data:  women and men
## 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, men, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  women and men
## 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
```

```r
summary(mod)
```

```
## 
## Call:
## lm(formula = los ~ gender, data = x)
## 
## 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
```

We can see that from the first call, the p-value is 0.2205 and the second call is 0.224 where the second call p-value matches the p-value for gender in the summary (0.224)