HW1_Reg_lian

Jiayi Lian

September 25, 2019

P1

It seems to be a good fit. Because the scatter plot shows that x and y approxmately have an linear relationship and the regression line is just located at the center of y's with repect to each x.

```
# b) Estimate the expected umber of broken ampules. When x=1
Beta_est[1,1]+Beta_est[2,1]*1
```

```
## [1] 14.2
```

```
# c) Estimate the increase in the expected number of broken ampules when there are two transfers as comp
Beta_est[1,1]+Beta_est[2,1]*2-Beta_est[1,1]-Beta_est[2,1]*1
```

```
## [1] 4
```

```
# Verify that your regression line goes through the point (xbar, ybar)
mean(freight$transfers)*Beta_est[2,1]+Beta_est[1,1]
```

```
## [1] 14.2
```

```
mean(freight$broken)
```

```
## [1] 14.2
```

Yes, the line goes throught the point.

P2

```
model<-lm(Y~freight$transfers)
summary(model)
```

```
##
## Call:
## lm(formula = Y ~ freight$transfers)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##   -2.2   -1.2    0.3    0.8    1.8
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        10.2000     0.6633  15.377 3.18e-07 ***
## freight$transfers   4.0000     0.4690   8.528 2.75e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.483 on 8 degrees of freedom
## Multiple R-squared:  0.9009, Adjusted R-squared:  0.8885
## F-statistic: 72.73 on 1 and 8 DF,  p-value: 2.749e-05
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Y
##                    Df Sum Sq Mean Sq F value    Pr(>F)
## freight$transfers  1  160.0   160.0  72.727 2.749e-05 ***
## Residuals          8   17.6     2.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
H= X %*% solve(t(X) %*% X) %*% t(X)
sigma2_est<- t(Y) %*% (diag(1,10)-H) %*% Y/8
Sxx<-sum((freight$transfers-mean(freight$transfers))^2)
# a
Beta_est[2,1]+sqrt(sigma2_est/Sxx)*c(qt(0.025,8),qt(0.975,8))
```

```
## Warning in sqrt(sigma2_est/Sxx) * c(qt(0.025, 8), qt(0.975, 8)): Recycling array of length 1 in array
##   Use c() or as.vector() instead.
```

```
## [1] 2.918388 5.081612
```

```r
# b
qt(0.975,8)
```

```
## [1] 2.306004
```

```r
Beta_est[2,1]/sqrt(sigma2_est/10)
```

```
##          [,1]
## [1,] 8.528029
```

```r
(1-pt(Beta_est[2,1]/sqrt(sigma2_est/10),8))*2
```

```
##              [,1]
## [1,] 2.748669e-05
```

```r
# c
qt(0.95,8)
```

```
## [1] 1.859548
```

```r
1-pt(Beta_est[2,1]/sqrt(sigma2_est/10),8)
```

```
##              [,1]
## [1,] 1.374335e-05
```

```r
# d
Beta_est[1,1]+sqrt(sigma2_est*solve(t(X) %*% X)[1,1])*c(qt(0.025,8),qt(0.975,8))
```

```
## Warning in sqrt(sigma2_est * solve(t(X) %*% X)[1, 1]) * c(qt(0.025, 8), : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## [1]  8.67037 11.72963
```

```r
# e
(Beta_est[1,1]-9)/sqrt(sigma2_est*solve(t(X) %*% X)[1,1])
```

```
##          [,1]
## [1,] 1.809068
```

```
1-pt((Beta_est[1,1]-9)/sqrt(sigma2_est*solve(t(X) %*% X)[1,1]),8)
```

```
##              [,1]
## [1,] 0.05402227
```

P3

```
Xnew1<-matrix(c(1,2),1,2)
Xnew2<-matrix(c(1,4),1,2)
# a
mid<- Xnew1 %*% solve(t(X)%*%X) %*% t(Xnew1)
Xnew1 %*% Beta_est +sqrt(sigma2_est*mid)*c(qt(0.0025,8),qt(0.9975,8))
```

```
## Warning in sqrt(sigma2_est * mid) * c(qt(0.0025, 8), qt(0.9975, 8)): Recycling array of length 1 in a
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew1 %*% Beta_est + sqrt(sigma2_est * mid) * c(qt(0.0025, 8), : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## [1] 15.65779 20.74221
```

```
Xnew2 %*% Beta_est +sqrt(sigma2_est*mid)*c(qt(0.0025,8),qt(0.9975,8))
```

```
## Warning in sqrt(sigma2_est * mid) * c(qt(0.0025, 8), qt(0.9975, 8)): Recycling array of length 1 in a
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew2 %*% Beta_est + sqrt(sigma2_est * mid) * c(qt(0.0025, 8), : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## [1] 23.65779 28.74221
```

```
Xnew1 %*% Beta_est +sqrt(sigma2_est*mid)*c(qt(0.005,8),qt(0.995,8))
```

```
## Warning in sqrt(sigma2_est * mid) * c(qt(0.005, 8), qt(0.995, 8)): Recycling array of length 1 in ar
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew1 %*% Beta_est + sqrt(sigma2_est * mid) * c(qt(0.005, 8), : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## [1] 15.97429 20.42571
```

```
Xnew2 %*% Beta_est +sqrt(sigma2_est*mid)*c(qt(0.005,8),qt(0.995,8))
```

```
## Warning in sqrt(sigma2_est * mid) * c(qt(0.005, 8), qt(0.995, 8)): Recycling array of length 1 in ar
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew2 %*% Beta_est + sqrt(sigma2_est * mid) * c(qt(0.005, 8), : Recycling array of length
##   Use c() or as.vector() instead.
```

```
## [1] 23.97429 28.42571
```

```
# b
Xnew1 %*% Beta_est +sqrt(sigma2_est*(mid+1))*c(qt(0.005,8),qt(0.995,8))
```

```
## Warning in sqrt(sigma2_est * (mid + 1)) * c(qt(0.005, 8), qt(0.995, 8)): Recycling array of length 1
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew1 %*% Beta_est + sqrt(sigma2_est * (mid + 1)) * c(qt(0.005, : Recycling array of leng
##   Use c() or as.vector() instead.
```

```
## [1] 12.74814 23.65186
```

```
# c
Xnew1 %*% Beta_est +sqrt(sigma2_est*(mid+1))*c(qt(0.01/6,8),qt(1-0.01/6,8))
```

```
## Warning in sqrt(sigma2_est * (mid + 1)) * c(qt(0.01/6, 8), qt(1 - 0.01/6, : Recycling array of lengt
##   Use c() or as.vector() instead.
```

```
## Warning in Xnew1 %*% Beta_est + sqrt(sigma2_est * (mid + 1)) * c(qt(0.01/6, : Recycling array of leng
##   Use c() or as.vector() instead.
```

```
## [1] 11.50195 24.89805
```

P4

```
# a
SSe=8*sigma2_est
SST=sum((freight$broken-mean(freight$broken))^2)
SSr=SST-SSe
SSr/SSe*8
```

```
##          [,1]
## [1,] 72.72727
```

```
# b
qf(0.95,1,8)
```

```
## [1] 5.317655
```

```
# c
Beta_est[2,1]/sqrt(sigma2_est/10)
```

```
##          [,1]
## [1,] 8.528029
```

```
# d
SSr/SST
```

```
##           [,1]
## [1,] 0.9009009
```

```
sqrt(SSr/SST)
```

```
##          [,1]
## [1,] 0.949158
```