Stat 344 – HW 7

Trey Tipton

March 18, 2022

Problem 6.6

c.)

```
x <- c(1,2,3,4)
y <- c(2,5,6,8)

beta1 <- (sum(x*y))/(sum(x^2))
beta1
```

```
## [1] 2.066667
```

```
lm(y ~ 0 + x)
```

```
##
## Call:
## lm(formula = y ~ 0 + x)
##
## Coefficients:
##     x
## 2.067
```

```
lm(y ~ -1 + x)
```

```
##
## Call:
## lm(formula = y ~ -1 + x)
##
## Coefficients:
##     x
## 2.067
```

```
summary(lm(y ~ 0 + x))
```

```
##
## Call:
## lm(formula = y ~ 0 + x)
##
## Residuals:
##        1        2        3        4
## -0.06667  0.86667 -0.20000 -0.26667
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x  2.06667    0.09813   21.06 0.000234 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5375 on 3 degrees of freedom
## Multiple R-squared:  0.9933, Adjusted R-squared:  0.991
## F-statistic: 443.5 on 1 and 3 DF,  p-value: 0.0002342
```

e.)

```
e <- y - (beta1*x)
sumq <- sum(e^2)
esigma <- sqrt(sumq/(length(y)-1))
esigma
```

```
## [1] 0.5374838
```

```
se <- esigma/(sqrt(sum(x^2)))
se
```

```
## [1] 0.09813068
```

Problem 6.34

```
model <- lm(distance ~ projectileWt, data = Trebuchet2)
x <- Trebuchet2$projectileWt
se.beta1 <- x - mean(x)
summary(model)
```

```
##
## Call:
## lm(formula = distance ~ projectileWt, data = Trebuchet2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1961 -0.2929  0.1631  0.4392  0.7869
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.62939    0.81878  12.982 3.39e-09 ***
## projectileWt -0.09460    0.01713  -5.524 7.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.743 on 14 degrees of freedom
## Multiple R-squared:  0.6855, Adjusted R-squared:  0.663
## F-statistic: 30.51 on 1 and 14 DF,  p-value: 7.495e-05
```

```
sse <- sum(resid(model)^2)
s <- sqrt(sse/(length(x) - 2))
s
```

```
## [1] 0.7429528
```

```
sxx <- sum((x - mean(x))^2)
meansq <- mean(x)^2
se.beta1 <- s/sqrt(sxx)
```

```
se.beta1
```

```
## [1] 0.01712611
```

```
se.beta0 <- s*(sqrt((1/length(x)) + (meansq/sxx)))
se.beta0
```

```
## [1] 0.8187797
```

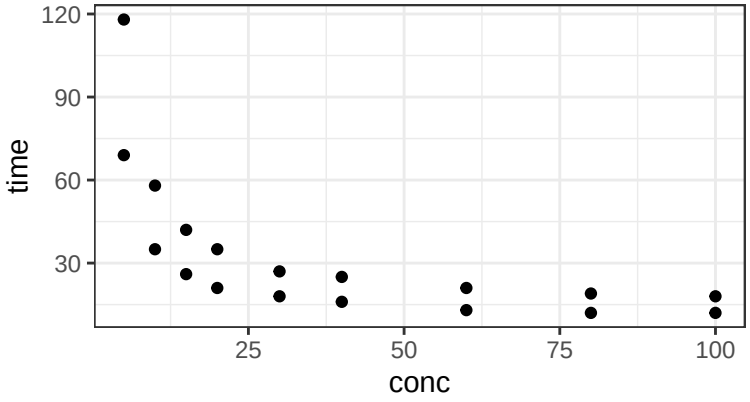
Problem 6.59

a.) Response variable: Time

Predictor variable: Concentration

I chose time as the response variable because it seems like the percentage concentrations of plasma would affect how long it takes for the blood to clot.

```
gf_point(time ~ conc, data = clot)
```



b.)

```
clot.model <- lm(time ~ conc, data = clot)
summary(clot.model)
```

```
##
## Call:
## lm(formula = time ~ conc, data = clot)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.540 -12.049  -5.275   8.859  67.931
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  52.5791     8.2295   6.389 8.97e-06 ***
## conc         -0.5020     0.1619  -3.100  0.00688 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.54 on 16 degrees of freedom
## Multiple R-squared:  0.3753, Adjusted R-squared:  0.3362
## F-statistic: 9.612 on 1 and 16 DF,  p-value: 0.006876
```

Equation: $\hat{\beta}_1$ = -0.5020 $\hat{\beta}_0$ = 52.5791 σ = 21.54

$$Y_{time} = 52.5791 - 0.5020 x_{conc} + \epsilon$$

, $\epsilon \sim$ Norm(0, 21.54)

c.)

```
xbar <- mean(clot$conc)
ybar <- mean(clot$time)
sy <- sd(clot$time)
sx <- sd(clot$conc)

r <- cor(clot$time ~ clot$conc)

beta1 <- r*(sy/sx)
beta1
```

```
## [1] -0.5019774
```

```
beta0 <- ybar - beta1*xbar
beta0
```

```
## [1] 52.5791
```

```
resids <- clot$time - (beta0 + beta1*clot$conc)
sum <- sum(resids^2)
sigma <- sqrt(sum/length(clot$time))
sigma
```

```
## [1] 20.30932
```

d.)

```
ll <- function(theta, x){
  beta0 <- theta[1]
  beta1 <- theta[2]
  sigma <- theta[3]
  resids <- clot$time - (beta0 + beta1*clot$conc)
  if (sigma < 0) return(NA)
  dnorm(resids, mean = 0, sd = sigma, log = TRUE)
}


library(maxLik)
```

```
## Loading required package: miscTools
```

```
##
## Please cite the 'maxLik' package as:
## Henningsen, Arne and Toomet, Ott (2011). maxLik: A package for maximum likelihood estimation in R. C
##
## If you have questions, suggestions, or comments regarding the 'maxLik' package, please use a forum o
## https://r-forge.r-project.org/projects/maxlik/
```

```
maxLik(logLik = ll, start = c(beta0 = 0, beta1 = 1, sigma = .5), x = clot)
```

```
## Maximum Likelihood estimation
## Newton-Raphson maximisation, 23 iterations
## Return code 8: successive function values within relative tolerance limit (reltol)
```

```
## Log-Likelihood: -79.74033 (3 free parameter(s))
## Estimate(s): 52.57985 -0.5019891 20.31005
```