Lab 9 Solutions

Naomi Giertych

11/6/2018

Lab Activity Part 1

```
bottle_2016 <- read.csv('bottle_2016.csv', header = T)

lm1 <- lm(T_degC ~ Depthm + Salnty + O2ml_L,
          data = bottle_2016)
summary(lm1)$coefficients[, c("Estimate", "Std. Error")]
```

```
##                  Estimate   Std. Error
## (Intercept) -78.591784806 3.6966365630
## Depthm       -0.003844847 0.0001576979
## Salnty        2.481605712 0.1077711182
## O2ml_L        1.955793588 0.0240747199
```

```
# confidence interval for B3 (relationship between oxygen)
z_star <- qnorm(0.975)

pred <- summary(lm1)$coefficients["O2ml_L", c("Estimate")]
SE <- summary(lm1)$coefficients["O2ml_L", c("Std. Error")]

cat("CI = [", round(pred-z_star*SE,3),
    ", ", round(pred+z_star*SE,3), "]", sep = "")
```

```
## CI = [1.909, 2.003]
```

Lab Activity Part 2

```
# read in the crime dataset
crime <- read.delim("crime.txt", comment.char = "#")

# Ha: B1 (pct_25_hs_grad) not equal to 0

  # fit the model
lm1 <- lm(total_crime_count ~ ., data = crime)
summary(lm1)$coefficients[, c("Estimate", "Std. Error")]
```

```
##                          Estimate Std. Error
## (Intercept)           489.6485970 472.365924
## annual_police_funding  10.9806703   3.077784
## pct_25_hs_grad         -6.0885294   6.543685
## pct_19_not_hs_grad      5.4803042  10.053499
## pct_24_college          0.3770443   4.417396
## pct_25_college          5.5004712  13.753907
```

```r
# conduct the hypothesis test

2*pnorm(summary(lm1)$coefficients["pct_25_hs_grad", "Estimate"],
      mean = 0,
      sd = summary(lm1)$coefficients["pct_25_hs_grad", "Std. Error"])
```

```
## [1] 0.3521415
```

Lab Activity Part 2 cont.

```r
# confidence interval and prediction interval for new value
  # new value
x_star <- c(1, 38, 50, 19, 15, 12)
pred <- x_star %*%coef(lm1)
V <- summary(lm1)$cov.unscaled
s <- summary(lm1)$sigma
SE <- s*sqrt(x_star%*%V%*%x_star)
c <- qnorm(0.975)

  # confidence interval
cat("CI = [", round(pred-c*SE,3),
    ", ", round(pred+c*SE,3), "]", sep = "")
```

```
## CI = [621.788, 934.761]
```

```r
  # prediction interval
SE_pred <- s*sqrt(1 + x_star%*%V%*%x_star)
cat("PI = [", round(pred-c*SE_pred,3),
    ", ", round(pred+c*SE_pred,3), "]", sep = "")
```

```
## PI = [257.89, 1298.66]
```

Lab Activity Part 3

```r
# simulate rolling a weighted die
# simulate rolling a die 100 times
die_roll <- function(x){
  r <- runif(x, min = 0, max = 7)
  value <- ceiling(r)
  if(value > 5) {value = 6}
  return(value)
}
die_rolls <- replicate(100, die_roll(1))

# simpler solution
sides <- c(1:6)
die_rolls2 <- sample(100, sides, replace = TRUE)

# get the count of rolls for each value
die_prop <- aggregate(data.frame(count = die_rolls),
                      list(value = die_rolls), FUN = length)
```

```r
# calculate our test statistic
die_prop$exp <- 100*(1/7)
die_prop$exp[die_prop$value == 6] = 100*(2/7)
chi_sqr <- sum((die_prop$count - die_prop$exp)^2/sum(die_prop$exp))
# calculate our p-value
# chi-squared has 5 degrees of freedom
1 - pchisq(chi_sqr, 5)
```

```
## [1] 0.9906346
```

Exit Ticket

```r
# weighted coin

# coin tosses
coin_toss <- function(x){
  r <- runif(x, min = 0, max = 3)
  value <- ceiling(r)
  if(value > 1) {value = 2}
  return(value)
}
coin_tosses <- replicate(100, coin_toss(1))

# get the count of rolls for each value
coin_prop <- aggregate(data.frame(count = coin_tosses),
                       list(value = coin_tosses), FUN = length)
# calculate our test statistic
coin_prop$exp <- 100*(1/3)
coin_prop$exp[coin_prop$value == 2] = 100*(2/3)
chi_sqr <- sum((coin_prop$count - coin_prop$exp)^2/sum(coin_prop$exp))
# calculate our p-value
# chi-squared has 1 degrees of freedom
1 - pchisq(chi_sqr, 1)
```

```
## [1] 0.924886
```

Exit Ticket cont.

```r
# from the summary results of the linear model fit above, we know that we have 44 degrees of freedom
t_star <- qt(0.995, 44)

pred <- summary(lm1)$coefficients["pct_25_hs_grad", c("Estimate")]
SE <- summary(lm1)$coefficients["pct_25_hs_grad", c("Std. Error")]

cat("CI = [", round(pred-t_star*SE,3),
    ", ", round(pred+t_star*SE,3), "]", sep = "")
```

```
## CI = [-23.706, 11.529]
```

The t-distribution may be more appropriate because our sample size is relatively small with only 50 observations in the crime dataset.