Assignment 1: Application

Lien Dao

2/27/2022

Indicators data

```
library(datasets)
library(readr)
url <- "https://gattonweb.uky.edu/sheather/book/docs/datasets/indicators.txt"
indicators <- read.table(url, header = TRUE)
print(indicators)
```

```
##              MetroArea PriceChange LoanPaymentsOverdue
## 1              Atlanta         1.2                4.55
## 2               Boston        -3.4                3.31
## 3              Chicago        -0.9                2.99
## 4               Dallas         0.8                4.26
## 5               Denver        -0.7                3.56
## 6              Detroit        -9.7                4.71
## 7             LasVegas        -6.1                4.90
## 8           LosAngeles        -4.8                3.05
## 9  MiamiFt.Lauderdale        -6.4                5.63
## 10  MinneapolisStPaul        -3.4                3.01
## 11             NewYork        -3.8                3.29
## 12             Phoenix        -7.3                3.26
## 13            Portland         3.8                1.93
## 14            SanDiego        -7.8                3.45
## 15        SanFrancisco        -4.1                2.29
## 16             Seattle         6.9                1.65
## 17               Tampa        -8.8                4.60
## 18        WashingtonDC        -7.2                3.14
```

Question 6

Simple linear regression model

```r
indicators_lm <- lm(PriceChange ~ LoanPaymentsOverdue, data = indicators)
summary(indicators_lm)
```

```
##
## Call:
## lm(formula = PriceChange ~ LoanPaymentsOverdue, data = indicators)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6541 -3.3419 -0.6944  2.5288  6.9163
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.5145     3.3240   1.358   0.1933
## LoanPaymentsOverdue  -2.2485     0.9033  -2.489   0.0242 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.954 on 16 degrees of freedom
## Multiple R-squared:  0.2792, Adjusted R-squared:  0.2341
## F-statistic: 6.196 on 1 and 16 DF,  p-value: 0.02419
```

a) 95% CI

```r
confint_95 <- confint(indicators_lm)[2, ]
confint_95
```

```
##      2.5 %     97.5 %
## -4.1634543 -0.3335853
```

b) 95% CI for E[Y|X=4]

```r
predict(indicators_lm, data.frame(LoanPaymentsOverdue = 4), interval = "confidence")
```

```
##         fit       lwr       upr
## 1 -4.479585 -6.648849 -2.310322
```

Question 7

Function to calculate basic statistics

```r
RSS <- function(x,y){
  mean_x <- mean(x)
  mean_y <- mean(y)

  Sxy = sum((x - mean(x)) * (y - mean(y)))
  Sxx = sum((x - mean(x))^2)
  Syy = sum((y - mean(y))^2)

  beta_1_hat = Sxy / Sxx
  beta_0_hat = mean(y) - beta_1_hat * mean(x)
  result <- data.frame(mean_x, mean_y,
                       Sxx, Syy, Sxy,
                       beta_0_hat, beta_1_hat)
}
```

Results

```r
rss_all <- RSS(indicators$LoanPaymentsOverdue, indicators$PriceChange)
rss_all
```

```
##     mean_x    mean_y      Sxx      Syy       Sxy beta_0_hat beta_1_hat
## 1 3.532222 -3.427778 19.16011 347.0161 -43.08189   4.514494   -2.24852
```

Question 8

a) Calculate predicted y

```
indicators$pred_y <- rss_all[1,6] + rss_all[1,7]*indicators$LoanPaymentsOverdue
```

Function to get ANOVA table's elements

```
ANOVA <- function(n,x,y,y_pred){
  df_reg <- 1
  df_res <- n-2
  df_total <- n-1

  SS_reg = sum((y_pred - mean(y))^2)
  SSE = sum((y - y_pred)^2)
  SST = sum((y - mean(y))^2)

  MSR = SS_reg/df_reg
  MSE = SSE/df_res

  F_stat = MSR/MSE

  p_value <- pf(F_stat, df_reg, df_res, lower.tail = FALSE)
  result <- data.frame(df_reg, df_res, df_total,
                       SS_reg, SSE, SST,
                       MSR, MSE, NA,
                       F_stat, NA, NA,
                       p_value, NA, NA)
}
```

ANOVA table

```
num_value <- 18
anova_all <- ANOVA(num_value,indicators$LoanPaymentsOverdue,indicators$PriceChange,indicators$pred_y)
anova_table <- matrix(anova_all,ncol = 5)
dimnames(anova_table) <- list("  Group" = c("Regression", "Residual", "Total"),
                              "ANOVA" = c("Df", "SS", "MS", "F stat", "P value"))
anova_table
```

```
##              ANOVA
##   Group      Df SS       MS       F stat   P value
##   Regression 1  96.87048 96.87048 6.196101 0.02419411
##   Residual   16 250.1456 15.6341  NA       NA
##   Total      17 347.0161 NA       NA       NA
```

b) Unbiased estimator of Var(Y|X) is S^2 (or MSE)

```
unbiased_s <- anova_all[1,8]
unbiased_s
```

```
## [1] 15.6341
```

Question 9

a) Standard error of b1 hat

```
se_b1_hat <- sqrt(unbiased_s/rss_all[1,3])
se_b1_hat
```

```
## [1] 0.9033113
```

b) Ho: b1=0; Ha: b1 >< 0

```
t_stat <- abs(rss_all[1,7]/se_b1_hat)
t_stat
```

```
## [1] 2.489197
```

```
t_critical <- qt(p=0.05/2, df=(num_value-2), lower.tail=FALSE)
t_critical
```

```
## [1] 2.119905
```

```
as.logical(t_stat>t_critical)
```

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

Conclusion: We reject the null hypothesis and conclude that there a linear association between X and Y at 0.05 significant level

Question 10

a) 95% CI for b1

```
lwb <- rss_all[1,7] - t_critical*se_b1_hat
upb <- rss_all[1,7] + t_critical*se_b1_hat
confint_b1 <- c(lwb, upb)
confint_b1
```

```
## [1] -4.1634543 -0.3335853
```

Conclusion: We reject the null hypothesis because 0 does not belong to the confidence interval (we have the same conclusion as before)

b) 95% CIs for population regression line and prediction intervals of Y

Function to obtain 95% CIs for population regression line

```
confi_reg <- function(n,x_real,x,y){
  t_critical <- qt(p=0.05/2, df=(num_value-2), lower.tail=FALSE)
  y_pred <- rss_all[1,6] + rss_all[1,7]*x_real
  lwb <- y_pred - t_critical*sqrt(unbiased_s*((1/n)+(x_real-mean(x))^2/rss_all[1,3]))
  upb <- y_pred + t_critical*sqrt(unbiased_s*((1/n)+(x_real-mean(x))^2/rss_all[1,3]))
  result <- c(lwb,upb)
}
```

x = 2.1

```
x1 <- confi_reg(num_value,2.1,indicators$LoanPaymentsOverdue,indicators$PriceChange)
x1
```

```
## [1] -3.587521  3.172725
```

x = 3.3

```
x2 <- confi_reg(num_value,3.3,indicators$LoanPaymentsOverdue,indicators$PriceChange)
x2
```

```
## [1] -4.9307295 -0.8805135
```

x = 4.0

```
x3 <- confi_reg(num_value,4.0,indicators$LoanPaymentsOverdue,indicators$PriceChange)
x3
```

```
## [1] -6.648849 -2.310322
```

x = 4.4

```
x4 <- confi_reg(num_value,4.4,indicators$LoanPaymentsOverdue,indicators$PriceChange)
x4
```

```
## [1] -7.960598 -2.797388
```

Function to obtain prediction intervals of Y

```
confi_Y <- function(n,x_real,x,y){
  t_critical <- qt(p=0.05/2, df=(num_value-2), lower.tail=FALSE)
  y_pred <- rss_all[1,6] + rss_all[1,7]*x_real
  lwb <- y_pred - t_critical*sqrt(unbiased_s*(1+(1/n)+(x_real-mean(x))^2/rss_all[1,3]))
  upb <- y_pred + t_critical*sqrt(unbiased_s*(1+(1/n)+(x_real-mean(x))^2/rss_all[1,3]))
  result <- c(lwb,upb)
}
```

x = 2.1

```
y_x1 <- confi_Y(num_value,2.1,indicators$LoanPaymentsOverdue,indicators$PriceChange)
y_x1
```

```
## [1] -9.245366  8.830570
```

x = 3.3

```
y_x2 <- confi_Y(num_value,3.3,indicators$LoanPaymentsOverdue,indicators$PriceChange)
y_x2
```

```
## [1] -11.528886   5.717643
```

x = 4.0

```
y_x3 <- confi_Y(num_value,4.0,indicators$LoanPaymentsOverdue,indicators$PriceChange)
y_x3
```

```
## [1] -13.137838   4.178667
```

x = 4.4

```
y_x4 <- confi_Y(num_value,4.4,indicators$LoanPaymentsOverdue,indicators$PriceChange)
y_x4
```

```
## [1] -14.149644   3.391658
```

End