Regression Script

Ananthan Ambikairajah

11 February, 2022

Contents

1 Exercise 1 - Advertisement and Sales

1.1 Load data and inspect

```
album_data <- read.delim("Album Sales 1.dat", header = TRUE)
head(album_data)
```

```
##    adverts sales
## 1   10.256   330
## 2  985.685   120
## 3 1445.563   360
## 4 1188.193   270
## 5  574.513   220
## 6  568.954   170
```

```
tail(album_data)
```

```
##      adverts sales
## 195  700.929   250
## 196  910.851   190
## 197  888.569   240
## 198  800.615   250
## 199 1500.000   230
## 200  785.694   110
```

```
summary(album_data)
```

```
##     adverts             sales      
##  Min.   :   9.104   Min.   : 10.0  
##  1st Qu.: 215.918   1st Qu.:137.5  
##  Median : 531.916   Median :200.0  
##  Mean   : 614.412   Mean   :193.2  
##  3rd Qu.: 911.226   3rd Qu.:250.0  
##  Max.   :2271.860   Max.   :360.0  
```

```
str(album_data)
```

```
## 'data.frame':    200 obs. of  2 variables:
##  $ adverts: num  10.3 985.7 1445.6 1188.2 574.5 ...
##  $ sales  : int  330 120 360 270 220 170 70 210 200 300 ...
```

1.2 Run simple linear regression

```
album_lm_0 <- lm(sales ~ 1, data = album_data); summary(album_lm_0)
```

```
## 
## Call:
```

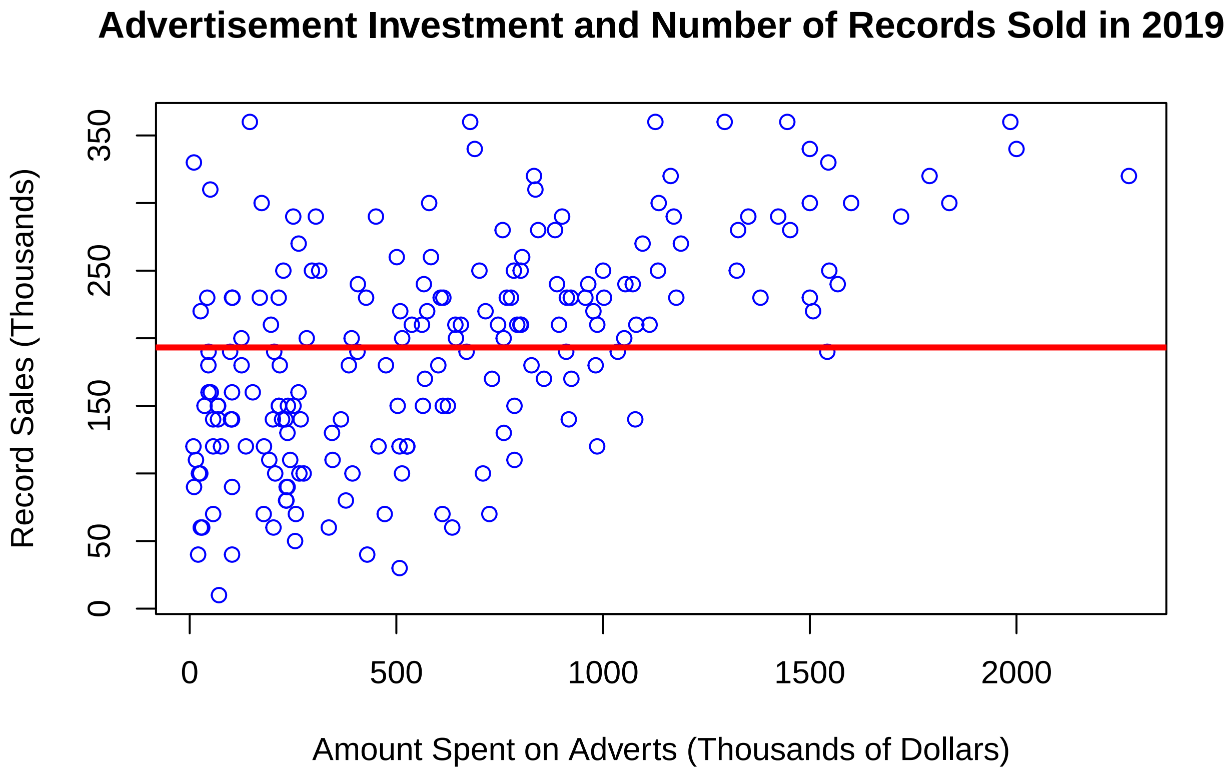
```
## lm(formula = sales ~ 1, data = album_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -183.2  -55.7    6.8   56.8  166.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  193.200      5.706   33.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 80.7 on 199 degrees of freedom
```

```
album_lm_1 <- lm(sales ~ 1 + adverts, data = album_data); summary(album_lm_1)
```

```
##
## Call:
## lm(formula = sales ~ 1 + adverts, data = album_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.949  -43.796   -0.393   37.040  211.866
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.341e+02  7.537e+00  17.799   <2e-16 ***
## adverts     9.612e-02  9.632e-03   9.979   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.99 on 198 degrees of freedom
## Multiple R-squared:  0.3346, Adjusted R-squared:  0.3313
## F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16
```

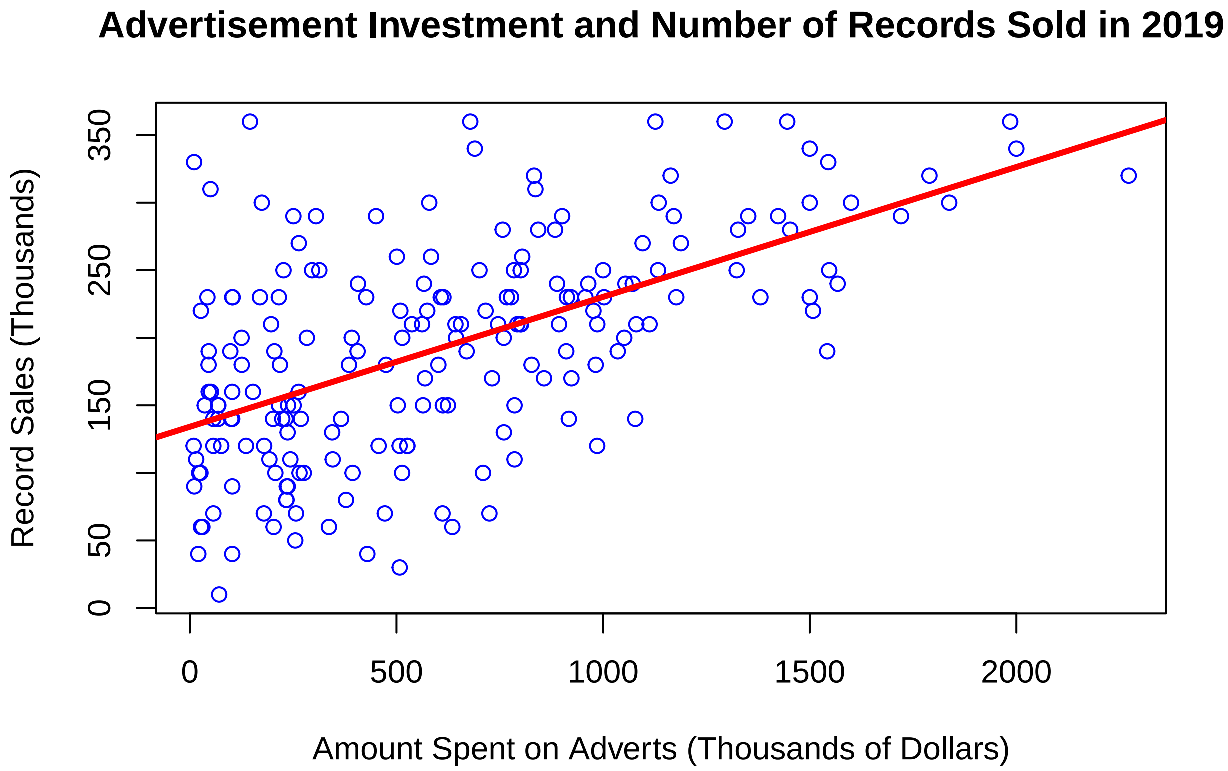
1.3 Using the mean as a simple model

```
par(mfrow = c(1,1))
plot(album_data$adverts, album_data$sales,
     col = "blue", type = "p",
     xlab = "Amount Spent on Adverts (Thousands of Dollars)",
     ylab = "Record Sales (Thousands)",
     main = "Advertisement Investment and Number of Records Sold in 2019")
abline(h = mean(album_data$sales), col = "red", lwd = 3)
```



1.4 Plot the linear regression

```
par(mfrow = c(1,1))
plot(album_data$adverts, album_data$sales,
     col = "blue", type = "p",
     xlab = "Amount Spent on Adverts (Thousands of Dollars)",
     ylab = "Record Sales (Thousands)",
     main = "Advertisement Investment and Number of Records Sold in 2019")
abline(album_lm_1, col = "red", lwd = 3)
```



2 Exercise 2 - Deriving Model Output

2.1 Baseline model

```
summary(album_lm_0)
```

```
##
## Call:
## lm(formula = sales ~ 1, data = album_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -183.2  -55.7    6.8   56.8  166.8
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  193.200      5.706   33.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 80.7 on 199 degrees of freedom
```

```
estimate <- mean(album_data$sales); estimate
```

```
## [1] 193.2
```

```
standard_error <- sd(album_data$sales)/sqrt(nrow(album_data)); standard_error
```

```
## [1] 5.706278
```

```
t_value <- estimate/standard_error; t_value
```

```
## [1] 33.85745
```

```
residuals_sales <- album_data$sales - mean(album_data$sales); residuals_sales
```

```
##   [1]  136.8  -73.2  166.8   76.8   26.8  -23.2 -123.2   16.8    6.8  106.8
##  [11]   96.8 -123.2  -43.2   -3.2   46.8  -93.2   56.8   16.8   86.8   36.8
##  [21]   16.8   36.8  126.8   16.8   36.8   56.8 -133.2  136.8  -43.2  -43.2
##  [31]  -13.2 -113.2  -13.2  -63.2  126.8   86.8    6.8  -63.2   -3.2  -43.2
##  [41]   36.8  116.8  146.8   46.8  -13.2   26.8 -153.2   -3.2   96.8  146.8
##  [51]   56.8   -3.2  -73.2   36.8   -3.2   16.8  -23.2  116.8 -103.2  -53.2
##  [61]  106.8  146.8  -23.2  -93.2    6.8 -113.2  -93.2 -123.2 -143.2   46.8
##  [71]  -33.2   96.8  -53.2   16.8  106.8   36.8   86.8  -33.2    6.8  -83.2
##  [81]  -83.2 -123.2  -93.2   -3.2 -123.2  166.8  166.8  106.8  -73.2  -43.2
##  [91]   26.8   86.8  106.8  -53.2   96.8  -13.2  -53.2   16.8   56.8   56.8
## [101]  -73.2   96.8 -133.2  -53.2   96.8  -33.2  -93.2  -33.2  -43.2  -53.2
## [111]   36.8   36.8 -163.2 -113.2   -3.2 -103.2  -73.2  -43.2   36.8  -43.2
## [121]   16.8  -13.2  -53.2  166.8 -183.2   46.8   76.8   96.8   26.8   36.8
## [131]   26.8   46.8   66.8  -23.2  -63.2   76.8  -53.2 -133.2   16.8   16.8
## [141]   46.8   16.8    6.8  -53.2 -103.2  -73.2  -93.2  166.8  -13.2  -43.2
## [151]  -83.2 -103.2  -33.2   36.8 -153.2 -133.2   36.8   36.8  -73.2  -43.2
## [161]  -73.2 -133.2   86.8  -73.2   36.8   36.8 -153.2  -53.2  166.8   16.8
## [171]   66.8   56.8    6.8  -43.2   56.8  -93.2   66.8   16.8   96.8   26.8
## [181] -123.2  -83.2   56.8  126.8  106.8  -13.2  -13.2    6.8  126.8  -53.2
## [191]  -93.2  -73.2   36.8  -43.2   56.8   -3.2   46.8   56.8   36.8  -83.2
```

```
quantile_residuals_sales <- quantile(residuals_sales); quantile_residuals_sales
```

```
##     0%    25%    50%    75%   100%
## -183.2  -55.7    6.8   56.8  166.8
```

```
residual_standard_error <- sd(residuals_sales); residual_standard_error
```

```
## [1] 80.69896
```

```
lower_confint <- estimate - (1.96*standard_error); lower_confint
```

```
## [1] 182.0157
```

```
upper_confint <- estimate + (1.96*standard_error); upper_confint
```

```
## [1] 204.3843
```

2.2 Simple linear regression

```
summary(album_lm_1)
```

```
##
## Call:
## lm(formula = sales ~ 1 + adverts, data = album_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.949  -43.796   -0.393   37.040  211.866
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.341e+02  7.537e+00  17.799   <2e-16 ***
## adverts     9.612e-02  9.632e-03   9.979   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.99 on 198 degrees of freedom
## Multiple R-squared:  0.3346, Adjusted R-squared:  0.3313
## F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
RMSE <- sqrt(sum(residuals(album_lm_1)^2)/df.residual(album_lm_1)); RMSE
```

```
## [1] 65.99144
```

```
# Residual Standard Error - Actually the standard deviation
R2 <- cor(album_data$adverts, album_data$sales)^2; R2
```

```
## [1] 0.3346481
```

```
R2_adjusted <- 1 - (1 - R2)*((nrow(album_data))-1)/((nrow(album_data))-1-1); R2_adjusted
```

```
## [1] 0.3312877
```

```
F_test <- anova(album_lm_0, album_lm_1); F_test
```

```
## Analysis of Variance Table
##
## Model 1: sales ~ 1
## Model 2: sales ~ 1 + adverts
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    199 1295952
## 2    198  862264  1    433688 99.587 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
t_adverts <- 0.09612/0.009632; t_adverts
```

```
## [1] 9.979236
```

```
t_intercept <- 134.1/7.537; t_intercept
```

```
## [1] 17.79223
```

```
residuals <- quantile(album_lm_1$residuals); residuals
```

```
##           0%          25%          50%          75%         100%
## -152.9492603  -43.7961350   -0.3933042   37.0404487  211.8657789
```

2.3 Create a function that calculates the summary of the linear model

```
linear_output <- function(baseline_model, simple_linear_model, data){
  list <- list()
  list[["RMSE"]] <- sqrt(sum(residuals(simple_linear_model)^2)/df.residual(simple_linear
  list[["R2"]] <- cor(data[,1], data[,2])^2
  list[["R2_adjusted"]] <- 1 - (1 - (cor(data[,1], data[,2])^2))*(((nrow(data))-1)/((nro
  list[["F_test"]] <- anova(baseline_model, simple_linear_model)
  list[["t_intercept"]] <- summary(simple_linear_model)$coefficients[1,1]/summary(simple
  list[["t_predictor"]] <- summary(simple_linear_model)$coefficients[2,1]/summary(simple
  list[["Residuals"]] <- quantile(simple_linear_model$residuals)
  print(list)
}
```

2.4 Test the function

```
linear_output(album_lm_0, album_lm_1, album_data)
```

```
## $RMSE
## [1] 65.99144
##
## $R2
## [1] 0.3346481
##
## $R2_adjusted
## [1] 0.3312877
##
```

```
## $F_test
## Analysis of Variance Table
## 
## Model 1: sales ~ 1
## Model 2: sales ~ 1 + adverts
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)    
## 1    199 1295952                                  
## 2    198  862264  1    433688 99.587 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## $t_intercept
## [1] 17.79853
## 
## $t_predictor
## [1] 9.979322
## 
## $Residuals
##           0%          25%          50%          75%         100% 
## -152.9492603  -43.7961350   -0.3933042   37.0404487  211.8657789
```

3 Exercise 3 - Anscombe Dataset

```
anscombe_data <- read.csv("Anscombe.csv", header = TRUE)
head(anscombe_data)
```

```
##    x    y distri
## 1 10 8.04      1
## 2  8 6.95      1
## 3 13 7.58      1
## 4  9 8.81      1
## 5 11 8.33      1
## 6 14 9.96      1
```

```
tail(anscombe_data)
```

```
##     x     y distri
## 39  8  7.04      4
## 40  8  5.25      4
## 41 19 12.50      4
## 42  8  5.56      4
## 43  8  7.91      4
## 44  8  6.89      4
```

```
summary(anscombe_data)
```

```
##        x              y              distri    
##  Min.   : 4   Min.   : 3.100   Min.   :1.00  
##  1st Qu.: 7   1st Qu.: 6.117   1st Qu.:1.75  
##  Median : 8   Median : 7.520   Median :2.50  
##  Mean   : 9   Mean   : 7.501   Mean   :2.50  
##  3rd Qu.:11   3rd Qu.: 8.748   3rd Qu.:3.25  
##  Max.   :19   Max.   :12.740   Max.   :4.00
```

```
str(anscombe_data)
```

```
## 'data.frame':    44 obs. of  3 variables:
##  $ x     : int  10 8 13 9 11 14 6 4 12 7 ...
##  $ y     : num  8.04 6.95 7.58 8.81 8.33 ...
##  $ distri: int  1 1 1 1 1 1 1 1 1 1 ...
```

3.1 Mean and variance for all four datasets - Using a forloop

3.1.1 Create empty data frame with named columns and rows

```
df <- data.frame(matrix(ncol = 4, nrow = 4))
x <- c("Group 1", "Group 2", "Group 3", "Group 4")
y <- c("Mean X", "Variance X", "Mean Y", "Variance Y")
rownames(df) <- x
colnames(df) <- y
```

3.1.2 Check empty dataframe

```
df
```

```
##         Mean X Variance X Mean Y Variance Y
## Group 1     NA         NA     NA         NA
## Group 2     NA         NA     NA         NA
## Group 3     NA         NA     NA         NA
## Group 4     NA         NA     NA         NA
```

3.1.3 Create a for loop

```
for(i in 1:4){
  df[i,1] <- mean(anscombe_data[anscombe_data$distri == i, "x"])
  df[i,2] <- var(anscombe_data[anscombe_data$distri == i, "x"])
  df[i,3] <- mean(anscombe_data[anscombe_data$distri == i, "y"])
  df[i,4] <- var(anscombe_data[anscombe_data$distri == i, "y"])
}
```

3.1.4 Check filled dataframe

```
df
```

```
##         Mean X Variance X   Mean Y Variance Y
## Group 1      9         11 7.500909   4.127269
## Group 2      9         11 7.500909   4.127629
## Group 3      9         11 7.500000   4.122620
## Group 4      9         11 7.500909   4.123249
```

3.2 Run all four regressions - using a for loop

```
anscombe_lm <- list()
for(i in 1:4){
anscombe_lm[[i]] <- summary(lm(y ~ x, data = anscombe_data[anscombe_data$distri==i,]))
}
anscombe_lm
```

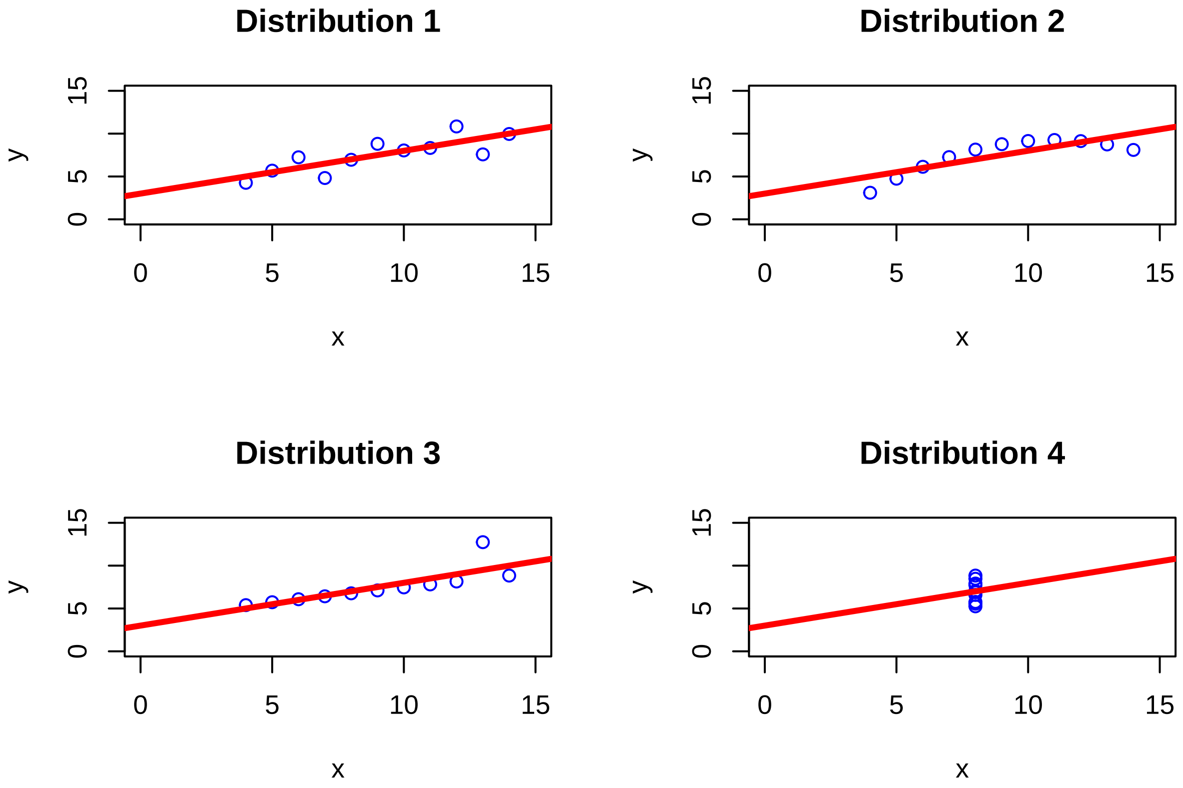
```
## [[1]]
##
## Call:
## lm(formula = y ~ x, data = anscombe_data[anscombe_data$distri ==
##     i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.92127 -0.45577 -0.04136  0.70941  1.83882
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001     1.1247   2.667  0.02573 *
## x             0.5001     0.1179   4.241  0.00217 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6665, Adjusted R-squared:  0.6295
## F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217
##
##
## [[2]]
##
## Call:
## lm(formula = y ~ x, data = anscombe_data[anscombe_data$distri ==
##     i, ])
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9009 -0.7609  0.1291  0.9491  1.2691
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125   2.667  0.02576 *
## x              0.500      0.118   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6662, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179
##
##
## [[3]]
##
## Call:
## lm(formula = y ~ x, data = anscombe_data[anscombe_data$distri ==
##     i, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1586 -0.6146 -0.2303  0.1540  3.2411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0025     1.1245   2.670  0.02562 *
## x             0.4997     0.1179   4.239  0.00218 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6292
## F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176
##
##
## [[4]]
##
## Call:
## lm(formula = y ~ x, data = anscombe_data[anscombe_data$distri ==
##     i, ])
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0017     1.1239   2.671  0.02559 *
## x             0.4999     0.1178   4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6667, Adjusted R-squared:  0.6297
## F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

3.3 Plot all four regressions - using a for loop

```
par(mfrow = c(2, 2))
for(i in 1:4){
plot(anscombe_data$x[anscombe_data$distri==i], anscombe_data$y[anscombe_data$distri==i],
     xlim=c(0,15), ylim=c(0,15),
     col = "blue", type = "p",
     xlab = "x",
     ylab = "y",
     main = paste("Distribution", i))
     intercept <- anscombe_lm[[i]][["coefficients"]][[1,1]]
     slope <- anscombe_lm[[i]][["coefficients"]][[2,1]]
     abline(a = intercept, b = slope, col = "red", lwd = 3)
}
```



Distribution 1

Distribution 2

Distribution 3

Distribution 4