Lab - 2 Simple Linear Regression

Prabh Talwar

2022-10-05

Question 1

```
num_obs = 30
beta_0 = 1.4
beta_1 = 3.1
sigma = 1
```

```
# Generating x values from a uniform distribution

set.seed(14)
x_vals = runif(30, 0, 10)
```

```
# With Functions and Data Frames
set.seed(14)
sim_slr = function(x, beta_0 = 1.4, beta_1 = 3.1, sigma = 1) {
  n = length(x)
  epsilon = rnorm(n, mean = 0, sd = 4)
  y = beta_0 + beta_1 * x + epsilon
  data.frame(predictor = x, response = y)
  }
```

```
# Using the function
sim_data <- sim_slr(x = x_vals, beta_0 = 1.4, beta_1 = 3.1, sigma = 1)

head(sim_data)
```

```
##   predictor  response
## 1  2.540337  6.627646
## 2  6.378273 28.048463
## 3  9.571886 39.559516
## 4  5.525467 24.517564
## 5  9.830671 31.730519
## 6  5.114739 22.183471
```

```
# Fitting the model
sim_fit = lm(response ~ predictor, data = sim_data)
coef(sim_fit)
```

```
## (Intercept)   predictor
##    1.809984    3.317706
```

With the simulation, the predicted model is almost equal to the actual model. The output of the model came out to be $y = 1.8 + 3.3x$. And our actual model is $y = 1.4 + 3.1x$

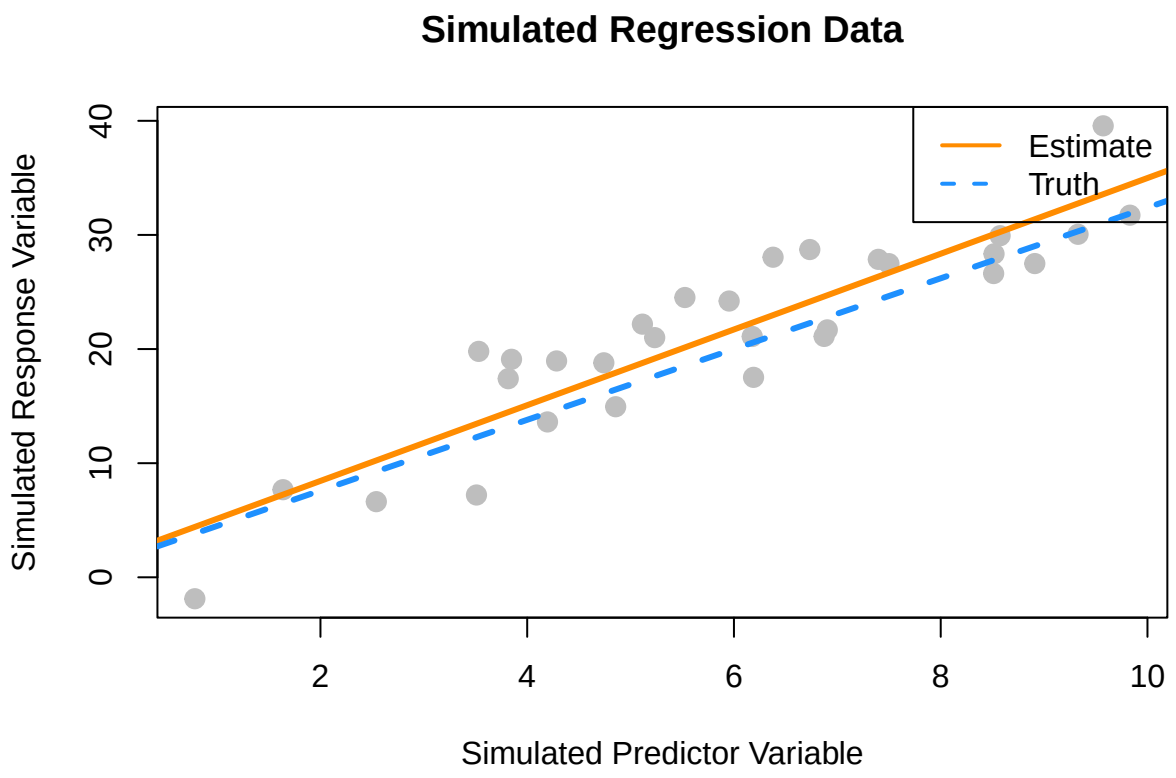
```r
# Plotting the model

plot(response ~ predictor, data = sim_data,
     xlab = "Simulated Predictor Variable",
     ylab = "Simulated Response Variable",
     main = "Simulated Regression Data",
     pch = 20,
     cex = 2,
     col = "grey")

abline(sim_fit, lwd = 3, lty = 1, col = "darkorange")

abline(beta_0, beta_1, lwd = 3, lty = 2, col = "dodgerblue")

legend("topright", c("Estimate", "Truth"),
       lty = c(1, 2),
       lwd = 2,
       col = c("darkorange", "dodgerblue"))
```



Question 2

```
library(datarium)
```

```
## Warning: package 'datarium' was built under R version 4.1.3
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'purrr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## Warning: package 'stringr' was built under R version 4.1.3
```

```
## Warning: package 'forcats' was built under R version 4.1.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
marketing <- datarium::marketing

head(marketing)
```

```
##   youtube facebook newspaper sales
## 1  276.12    45.36     83.04 26.52
## 2   53.40    47.16     54.12 12.48
## 3   20.64    55.08     83.16 11.16
## 4  181.80    49.56     70.20 22.20
## 5  216.96    12.96     70.08 15.48
## 6   10.44    58.68     90.00  8.64
```

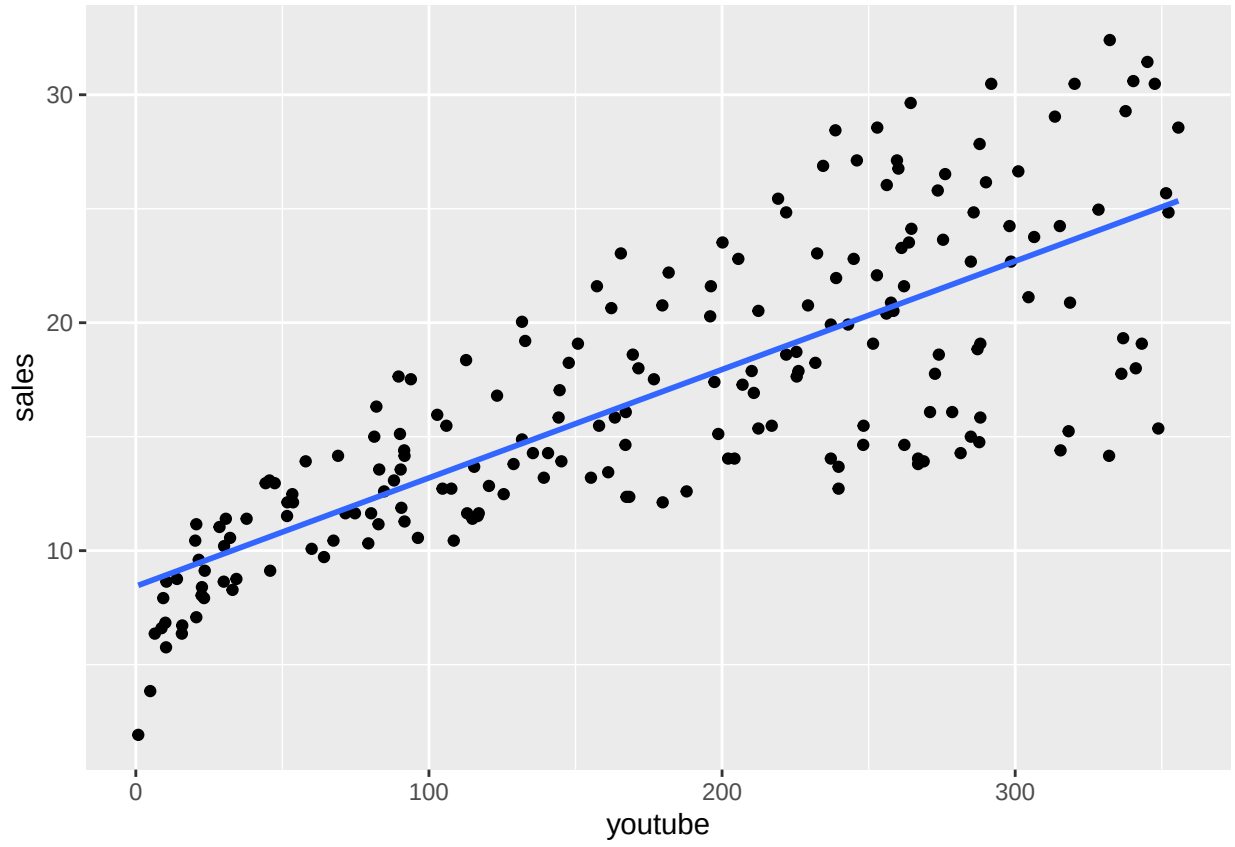
```
summary(marketing)
```

```
##     youtube          facebook       newspaper          sales
##  Min.   :  0.84   Min.   : 0.00   Min.   :  0.36   Min.   : 1.92
##  1st Qu.: 89.25   1st Qu.:11.97   1st Qu.: 15.30   1st Qu.:12.45
##  Median :179.70   Median :27.48   Median : 30.90   Median :15.48
##  Mean   :176.45   Mean   :27.92   Mean   : 36.66   Mean   :16.83
##  3rd Qu.:262.59   3rd Qu.:43.83   3rd Qu.: 54.12   3rd Qu.:20.88
##  Max.   :355.68   Max.   :59.52   Max.   :136.80   Max.   :32.40
```

Part (a)

```
ggplot(data = marketing, mapping = aes(x = youtube, y = sales))+
  geom_point()+
  geom_smooth(method='lm', se = FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
SLR <- lm(sales ~ youtube, data = marketing)
summary(SLR)
```

```
##
## Call:
## lm(formula = sales ~ youtube, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0632  -2.3454  -0.2295   2.4805   8.6548
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.439112   0.549412   15.36   <2e-16 ***
## youtube     0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.91 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

The linear equation: $y = 8.439112 + 0.047537x$. Residual standard error is the standard deviation of residual errors. Larger the RSE is bad fit will be our model to our data. We need our RSE to be small, indicating that model fits well to our data and here, for this model RSE is 3.91.

R^2 represents the proportion of variation explained by the regression. It ranges between 0 to 1 representing the variations in the data. Higher R^2 means the model fits best to the data and in our case our R^2 is 0.6119.

F-statistics is of overall significance indicates whether your linear regression model provides a better fit to the data than a model that contains no independent variables. F-statistic: 312.1 for our model.

```
confint(SLR, level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 7.35566312 9.52256140
## youtube     0.04223072 0.05284256
```