Linear regression with R

Lecture 4

Joaquin Cavieres

1. Linear regression with one explanatory variable

In this lecture we will see a brief summary of linear regression and how to perform it in R.

A linear regression consider a dependent variable Y and one or more independent (explanatory) variables $X_1, \ldots, X_n$. The main idea behind linear regression is that Y and X's have a linear relationship between them. Here, we will consider only one explanatory variable as purposes of example.

1.1. Simple linear regression

First of all, we will consider the data of the package "datarium". This package contains the impact of three advertising medias (youtube, facebook and newspaper) on sales of some product.

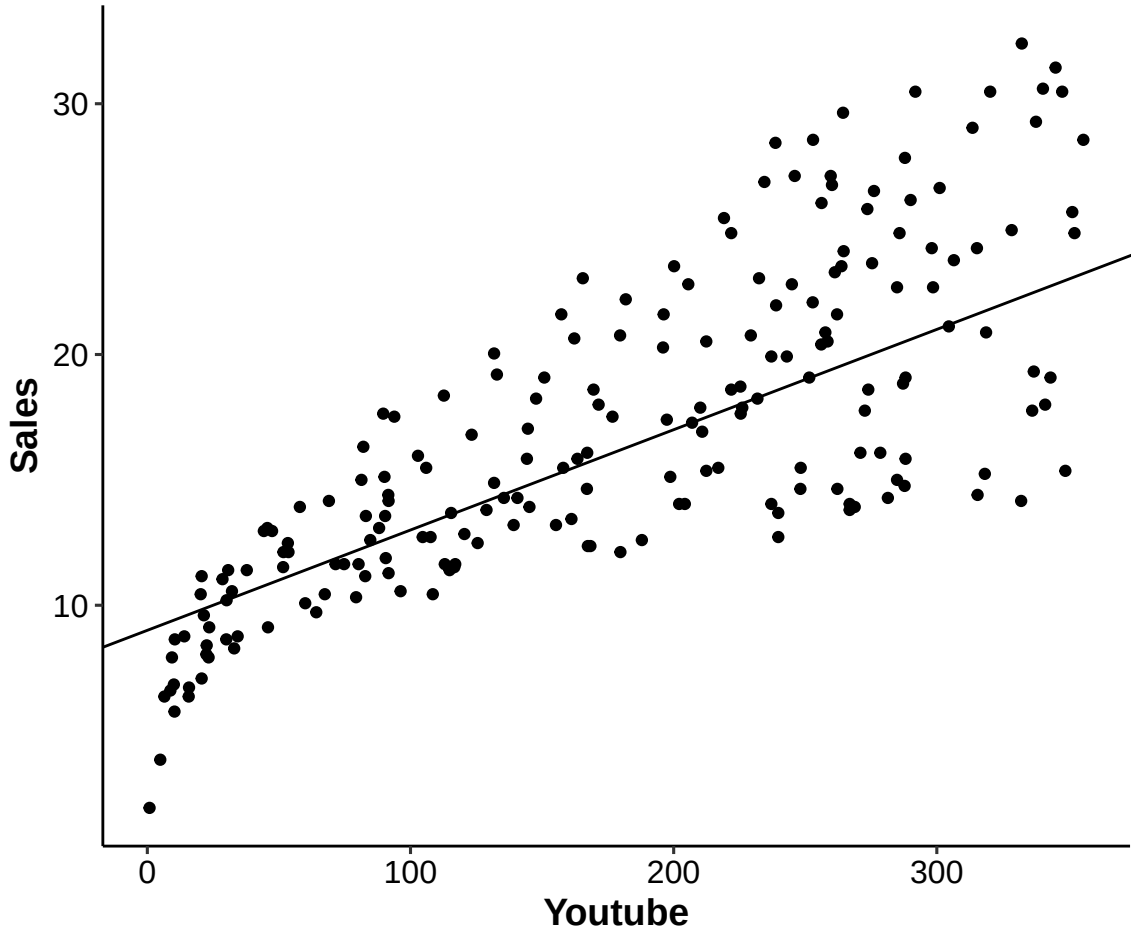
```
library(tidyverse)
library(ggpubr)
theme_set(theme_pubr())

library(datarium)
data("marketing")
head(marketing, 4)
```

```
##   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
```

The data are the advertising budget in thousands of dollars along with the sales. The advertising experiment has been repeated 200 times. For example, we want to know what is the relation between YouTube and the sales,

```r
ggplot(marketing, aes(x = youtube, y = sales)) +
       geom_point() +
       labs(y = "Sales", x = "Youtube") +
       geom_abline(intercept = 9,  slope = 0.04) +
       theme(axis.text=element_text(size=12),
       axis.title=element_text(size=14,face="bold"))
```



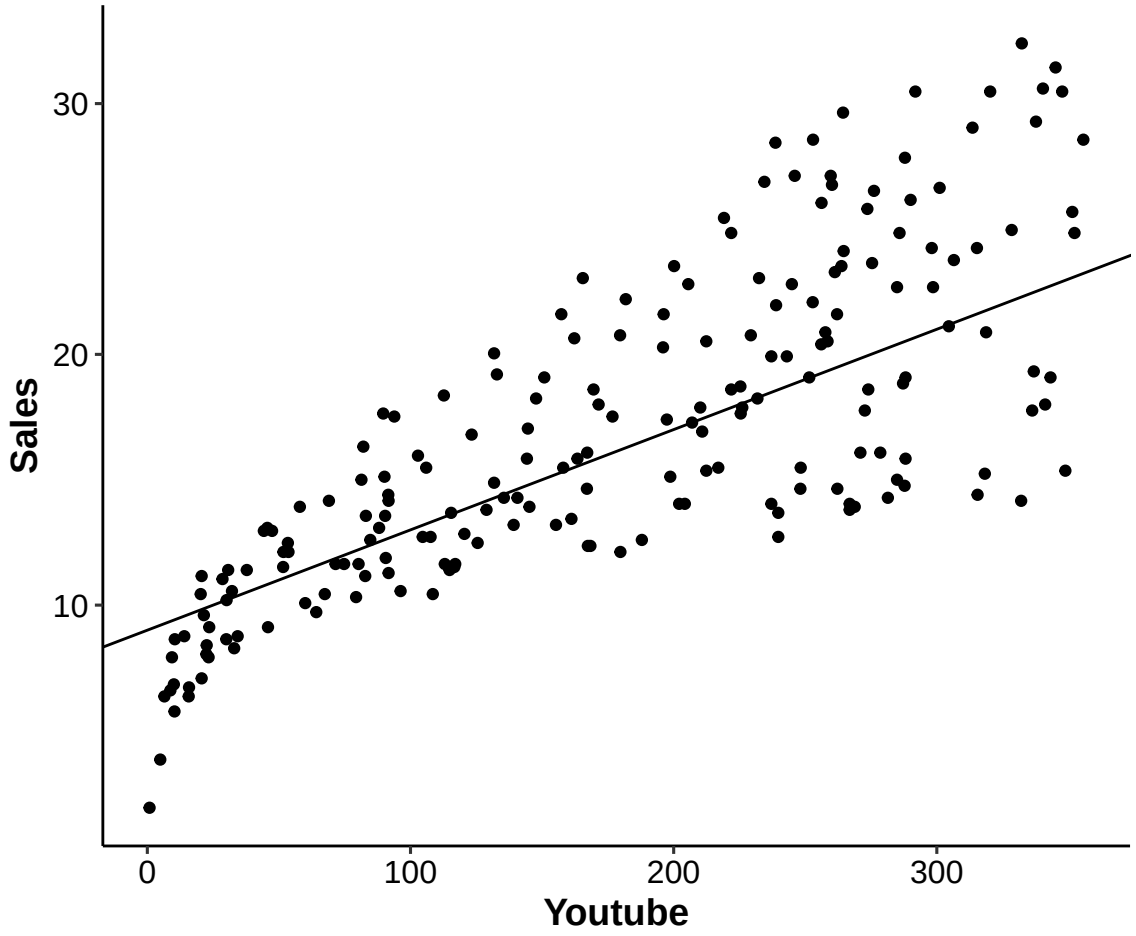
A simple linear regression model should be represented by the following expression:

$$Y = b \cdot X + a,$$

Only for now, we will assume that the function which related the sales with YouTube is the following:

$$\text{Sales} = 9 - 0{,}05 * \text{Youtube}.$$

```
ggplot(marketing, aes(x = youtube, y = sales)) + geom_point() +
       labs(y = "Sales", x = "Youtube") +
       geom_abline(intercept = 9,  slope = 0.04) +
       theme(axis.text=element_text(size=12),
       axis.title=element_text(size=14,face="bold"))
```



The line represents the systematic relationship between the sales and the advertising budget of Youtube, however, there are additional influences which imply that this relationship is not simple to describe or there could be error in the measurements (data). For the above, we incorporate to the model an error term which captures this "noise" in the data that we don't have implicitly in the data. In this way we can propose a statistical linear regression as follow:

$$\text{Sales} = \beta_0 + \beta_1 \times YouTube + \text{error term}$$

Finally, we can extend this representation to a mathematical representation:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

Simple linear regression

The mathematical expression for a a simple linear regression is:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{1}$$

where:

- i is an index that describes every observation, that is, $i, \ldots, n$
- Y_i is the response variable (or dependent variable)
- X_i is the covariate (or explanatory, independent variable)
- β_0 is the intercept of the model
- β_1 is the parameter associated with the covariate X
- ϵ_i is the error term assumed, in this case, distribuited $\mathcal{N}(0, \sigma^2)$.

2. Estimating the coefficients of the simple linear model

We don't know the values of β_0 and β_1, for the same we have to estimate them, and for estimate them we need to use data, So, considering the data of the Sales, we want to measure, in some way, the relationship between the Sales and the advertising budget of YouTube. We know that there is a relation between two looking at the previous plot, but we are interested in the parameters of the simple linear regression. Besides this, can we predict the future based on this data? First we have to define the estimators for the parameters.

Ordinary least square estimator (OLS)

The OLS estimator is a method that find the values for the coefficients that allows us get a line "as close as possible" to the observed data. The phrase "as close as possible" is quantified using the the sum of the squared mistakes made in predicting Y given X. So, if we think in $\hat{\beta}_0$ and $\hat{\beta}_1$ as estimators of β_0 and β_1, then the sum of squared estimation mistakes is:

$$\sum_{i=1}^{n}(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_i)^2. \tag{2}$$

where:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{\sum_{i=1}^{n}(X_i - \overline{X})^2}, \tag{3}$$

$$\tag{4}$$

$$\hat{\beta}_0 = \overline{Y} - \hat{\beta}_1 \overline{X}. \tag{5}$$

The predicted values $\hat{Y}_i$

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i, \tag{6}$$

and the residuals $\hat{\epsilon}_i$

$$\hat{\epsilon}_i = Y_i - \hat{Y}_i. \tag{7}$$

In R we can do the computation of $\hat{\beta}_1$ and $\hat{\beta}_0$ easily, for example:

```
library(datarium)
data("marketing")

attach(marketing) # To use the names of variables contained in the "marketing" data

# estimator for beta1
hat_beta1 <- sum((youtube - mean(youtube)) * (sales - mean(sales))) /
             sum((youtube - mean(youtube))^2)

# estimator for beta0
hat_beta0 <- mean(sales) - hat_beta1 * mean(youtube)

# print the results
hat_beta1
```

```
## [1] 0.04753664
```

```
hat_beta0
```

```
## [1] 8.439112
```

Although this computation is simple, in some cases it could turn on tedious, therefore in R we can use the function `lm()` to do the calculus of the parameters and other things.

```
# Using lm function
linear_fit <- lm(sales ~ youtube, data = marketing)

# print the results from the object "linear_fit"
linear_fit
```

```
##
## Call:
## lm(formula = sales ~ youtube, data = marketing)
##
## Coefficients:
## (Intercept)      youtube
##     8.43911      0.04754
```

From the previous lines of code we can infer that:

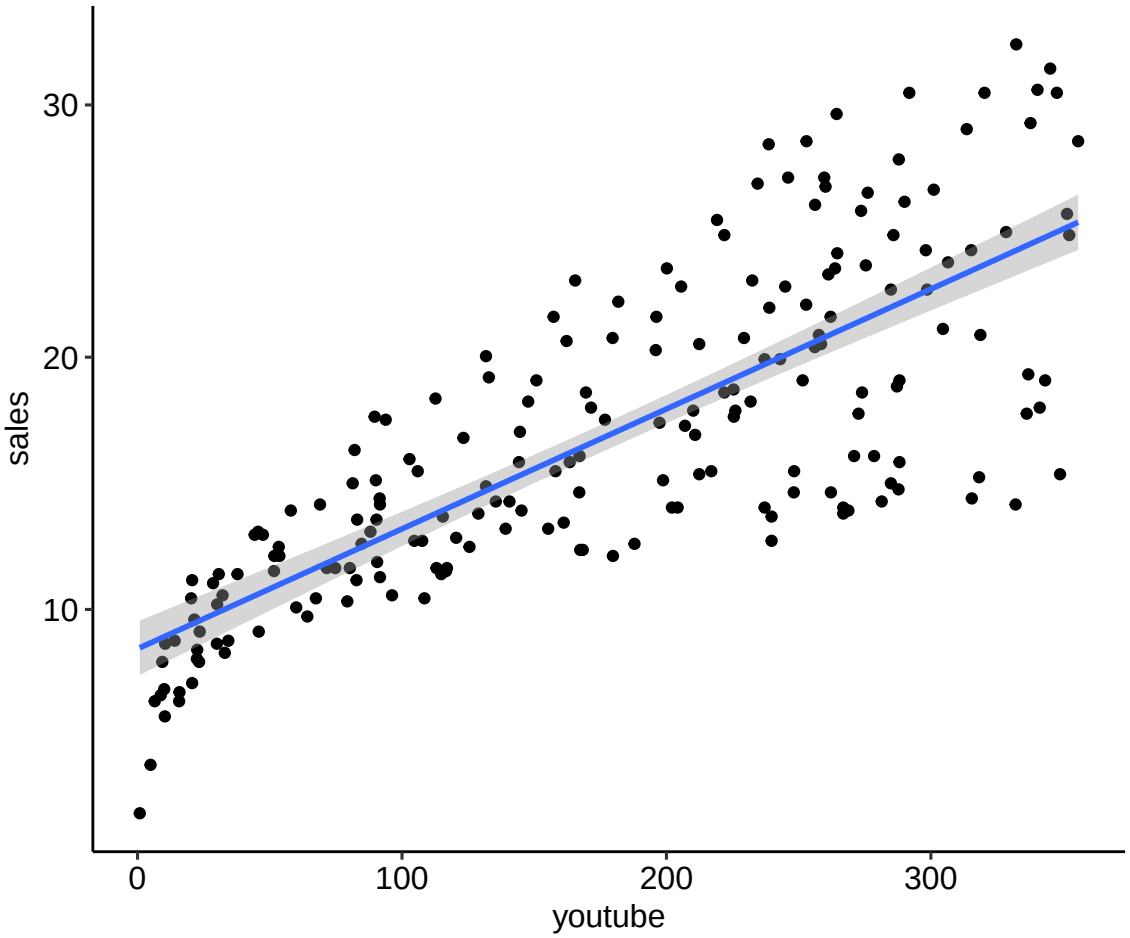
- The estimated regression line equation can be written as follow:

$$\text{sales} = 8{,}44 + 0{,}048 * \text{YouTube} \tag{8}$$

- β_0 (intercept) could be interpreted as the quantity of sales predicted for a zero YouTube advertising budget. Remember that we are working with units of thousand dollars, that is, for a YouTube advertising budget equal zero, we can expect a sale of 8.44 *1000 = 8440 dollars.
- β_1 is 0.048, this means that, for a YouTube advertising budget equal to 1000 dollars, we can expect an increase of 48 units (0.048*1000) in sales.

Adding the line into the scatter plot can be done by using the library "ggplot2" and the function `stat_smooth()`, which by default give us the confidence intervals around of the line.

```
ggplot(marketing, aes(youtube, sales)) +
       geom_point() +
       stat_smooth(method = lm)
```



3. Diagnostics of the model

After having found the linear regression for our model, we could ask ourselves, how good is the fit?

We can evaluate the fit of the simple linear model by the *diagnostics*. Other names for this procedures are: *measure of fit* or *goodness of fit*.

3.1. Coefficient of determination

The coefficient of determination (R^2) is the proportion of the variation in the response variable (Y_i) that is predictable from the independent variable(s) (X_i). It can be calculated as:

$$R^2 = \frac{ESS}{TSS}, \tag{9}$$

where:

$$\text{ESS} = \sum_{i=1}^{n} \left(\hat{Y}_i - \overline{Y}\right)^2, \tag{10}$$

$$\text{TSS} = \sum_{i=1}^{n} \left(Y_i - \overline{Y}\right)^2, \tag{11}$$

$$\tag{12}$$

A "good" value of R^2 is close to 1, where the values of it coefficient can be found between 0 and 1.

3.2. Standard error of the regression (SER)

The SER is an estimator of the standard deviation of the residuals ϵ_i and it quantify the magnitude of a typical deviation from the regression line.

$$s_{\hat{\epsilon}}^2 = \frac{1}{n-2} \sum_{i=1}^{n} \hat{\epsilon}_i^2 \tag{13}$$

thus

$$SER = s_{\hat{\epsilon}} = \sqrt{s_{\hat{\epsilon}}^2} \tag{14}$$

How to compute the R^2 and the SER in R?

```
# R^2
SSR <- sum(summary(linear_fit)$residuals^2)
TSS <- sum((sales - mean(sales))^2)
R2 <- 1 - SSR/TSS
R2
```

```
## [1] 0.6118751
```

```
# SER
n <- nrow(marketing)
SER <- sqrt(SSR / (n-2))
SER
```

```
## [1] 3.910388
```

However, we can get this same results using the function `summary()` from our fitted "linear_fit" model:

```r
# Summary of the object
summary(linear_fit)
```

```
##
## 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
```

Residual standard error (RSE), R-squared (R2) commonly are used to check how well the model fits to our data.

3.3. Coefficients significance

By the function `summary(linear_fit)` we also can see the significance of each parameter (coefficient) in the model.

Let's concentrate in the p-value. Here we want to test whether or not there is a statistically significant relationship between X_i and Y_i, that is whether or not the beta coefficient (β_1) of the predictor is significantly different from zero. For this analysis:

- $\boldsymbol{H}_0$ (null hypothesis): the coefficients are equal to zero (i.e., no relationship between X_i (YouTube) and Y_i (Sales))
- $\boldsymbol{H}_1$ (alternative hypothesis): the coefficients are not equal to zero (i.e., there is some relationship between X_i (YouTube) and Y_i (Sales))

In summary, If the p-value is less than or equal to the specified significance level α, $\boldsymbol{H}_0$ is rejected, otherwise, $\boldsymbol{H}_0$ is not rejected. The significance level α almost always is 0.05, therefore if the p-value (<2e-16 for β_1) is less than 0.05, then we reject $\boldsymbol{H}_0$, this means that "there is sufficient evidence to say that there is some relationship between X_i (YouTube) and Y_i (Sales)".

3.4. Confidence intervals

We can compute the standard errors associated with the parameters estimated by the model using the function `summary()`

```
summary(linear_fit)$coeff[, 1:2]
```

```
##               Estimate  Std. Error
## (Intercept) 8.43911226 0.549411528
## youtube     0.04753664 0.002690607
```

Furthermore, we can compute the confidence intervals for each parameter estimated as:

```
confint.default(linear_fit, level = 0.95)
```

```
##                  2.5 %     97.5 %
## (Intercept) 7.36228545 9.51593907
## youtube     0.04226315 0.05281013
```

3.5. Prediction

Finally, if we want to predict the Sales considering new data of advertising budget for YouTube, we can do it using the function `predict()` as follow:

```
new_data <- list(youtube=c(165, 100, 183))
predict(linear_fit, new_data)
```

```
##        1        2        3
## 16.28266 13.19278 17.13832
```

References

Hanck, C., Arnold, M., Gerber, A., & Schmelzer, M. (2019). Introduction to Econometrics with R. University of Duisburg-Essen, 1-9.

Simple linear regression in R: http://www.sthda.com/english/articles/40-regression-analysis/167-simple-linear-regression-in-r/