DSA1101 Topic 3: Linear Regression

Dawn Cheung

2024-03-04

Definition of Terms

Supervised Learning Methods

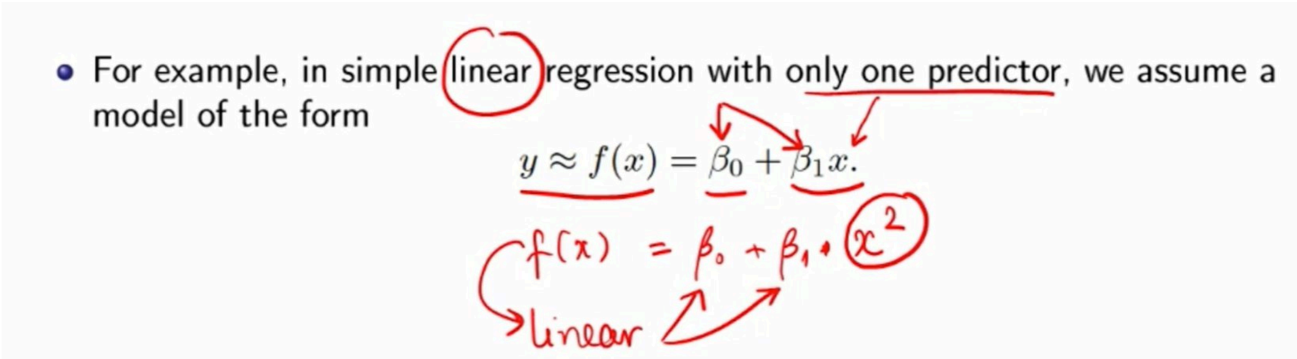
- Methods used in *making predictions about the future*
- It *predicts the response variable* in the future
- Dataset used must have the response variable
  - must both know which is the response variable (if any) & the values of it [ie x and y are known and can be idenified]
- x is the predictors, y is the outcome (response variable)
  - Assumes model as: y ≈ f(x)
- E.g. linear regression models where x is just 1 variable
- Hence determines given a certain predictor values for x, what is the most likely corresponding value of y

Linear Regression

- An analytical technique that models the relationship between several input variables and a continuos outcome variable.

-Assumes r/s btwn input variables and outcome variable is linear

- the “linearity” is in terms of the coefficients



HDB Resale Dataset

```r
# Enter code here
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.2
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(dplyr)
resale <- read.csv("~/GitHub/DSA1101 Slayers/datasets/hdbresale_reg.csv")
head(resale[,2:7]) #ignore 1st column (ID of flats)
```

```
##     month         town flat_type block    street_name storey_range
## 1 2012-03 CENTRAL AREA    3 ROOM   640       ROWELL RD     01 TO 05
## 2 2012-03 CENTRAL AREA    3 ROOM   640       ROWELL RD     06 TO 10
## 3 2012-03 CENTRAL AREA    3 ROOM   668      CHANDER RD     01 TO 05
## 4 2012-03 CENTRAL AREA    3 ROOM     5 TG PAGAR PLAZA     11 TO 15
## 5 2012-03 CENTRAL AREA    3 ROOM   271        QUEEN ST     11 TO 15
## 6 2012-03 CENTRAL AREA    4 ROOM  671A      KLANG LANE     01 TO 05
```

```
head(resale[,8:11])
```

```
##   floor_area_sqm flat_model lease_commence_date resale_price
## 1             74    Model A                1984       380000
## 2             74    Model A                1984       388000
## 3             73    Model A                1984       400000
## 4             59   Improved                1977       460000
## 5             68   Improved                1979       488000
## 6             75    Model A                2003       495000
```

floor_area_sqm is the 'independent variable', resale_price is the response variable

Simple Linear Regression (SLR)

- Suppose we have three observations. Each observation has an outcome y and an input variable x.

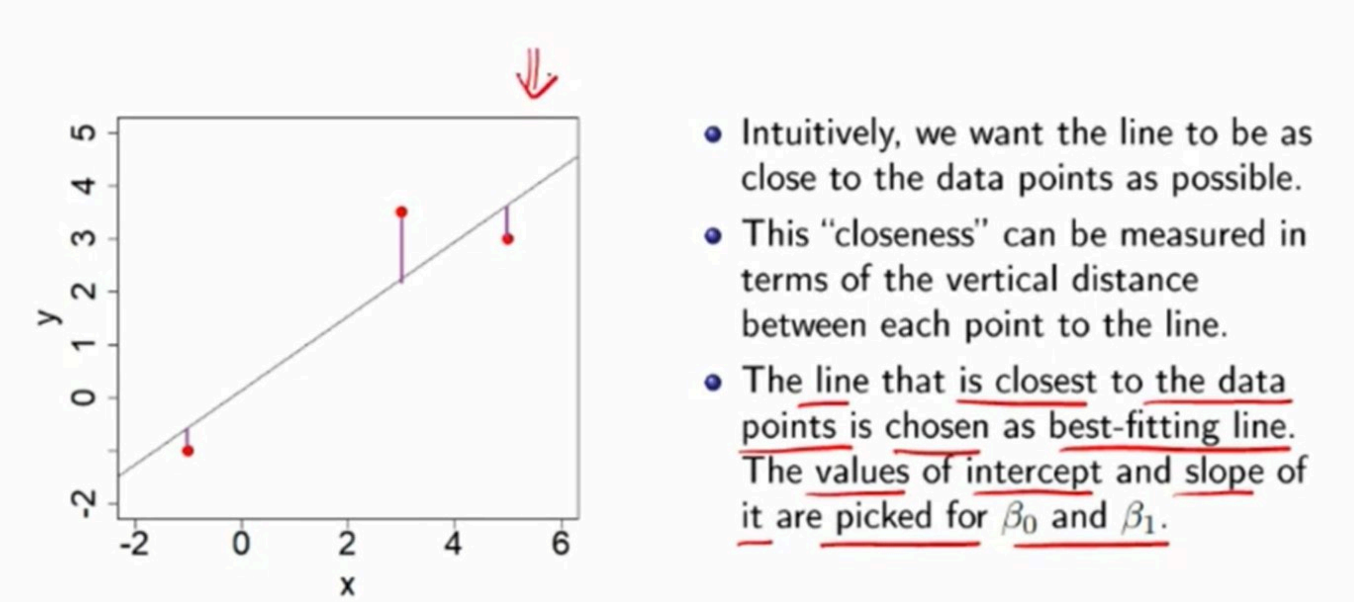
- We are interested in the linear relationship

$$y_i \approx \beta_0 + \beta_1 x_i \qquad i = 1, \ldots, 3$$

unknown

- only 1 input variable (ie only got x). if there were more than 1 input vairable then its multiple linear regression model

Ordinary Least Squares (OLS) Method



Linear regression of HDB unit resale price as a function of floor area in square meters **Solution:**

```
price = resale$resale_price #regressor
area = resale$floor_area_sqm #predictor

lm(price ~ area)$coeff
```

```
## (Intercept)        area
##   115145.730    3117.212
```

the fitted model is then ŷ 115145.730 + 3117.212x

Goodness-of-fit Model

- F-test
- Coefficient of determination, R^2

F-test:

small p-value => strong evidence against H0 => H1 (alt H) is accepted => model is highly significant

large p-value => cannot eliminate H0 => variables chosen might not be helpful at all in predicting the response

```
hdb.model = lm(price ~ area)
summary(hdb.model)
```

```
##
## Call:
## lm(formula = price ~ area)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -122852  -33539  -10984   17298  488719
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 115145.73    2949.14   39.04   <2e-16 ***
## area          3117.21      27.95  111.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56410 on 6053 degrees of freedom
## Multiple R-squared:  0.6727, Adjusted R-squared:  0.6727
## F-statistic: 1.244e+04 on 1 and 6053 DF,  p-value: < 2.2e-16
```

```
#OR just
summary(hdb.model)$fstatistic #and take the first value
```

```
##    value    numdf    dendf
## 12441.39     1.00  6053.00
```

```
#alt is summary(hdb.model)$fstatistic[1]
```

Since “F-statistic: 1.244e+04 on 1 and 6053 DF, p-value: < **2.2e-16**” p is smaller than 0.05 => strong evidence against the null

R^2:

```
#also summary(hdb.model) works just fine too
summary(hdb.model)$r.squared
```

```
## [1] 0.6727116
```

Multiple R-squared: 0.6727, Adjusted R-squared: 0.6727

TUTORIAL QUESTIONS

1 Read the data from the file Colleges.txt. Consider a simple linear regression of percentage of applicants accepted (Acceptance) on the median combined math and verbal SAT score of students (SAT), called Model M1.

1a) Write your own function in R to derive the equation of Model M1.

```r
library(tidyverse)
library(dplyr)
collegesdb <- read.csv("~/Github/DSA1101 Slayers/datasets/Colleges.txt", sep = "\t", header = TRUE) #THIS THE "\t" PLS HOW " "
attach(collegesdb)
#plan: get RSS first then derivative it?? then solve for intercept and gradient???
#ok no thats impossible how would u even get RSS direct
B1 = (sum(Acceptance*SAT) - mean(Acceptance)*sum(SAT))/(sum(SAT^2)-mean(SAT)*sum(SAT))
B0 = (sum(Acceptance*SAT) - B1*(sum(SAT)^2))/(sum(SAT))
#those didnt work

simple <- function(x, y) {
  B1 = (sum(x*y) - mean(y)*sum(x))/(sum(x^2)-mean(x)*sum(x))
  B0 = mean(y)-(B1*mean(x))
  paste0("equation is: y hat = ", B0, " + ", B1, "x") #or return statement, utu
}


xbar = (1/length(SAT))*sum(SAT)
ybar = (1/length(Acceptance))*sum(Acceptance)

#B1 = (sum(SAT-xbar)*sum(Acceptance-ybar))/sum((SAT-xbar)^2)


simple(SAT, Acceptance) #instanciation: calling your function w/ parameters lol
```

```
## [1] "equation is: y hat = 202.267744013677 + -0.130089357268962x"
```

1b) Use function lm() in R to derive the equation of Model M1. Compare with your answer in part (a).

```r
# Enter code here
M = lm(Acceptance ~ SAT) # OR u can do lm(Acceptance ~ SAT, collegesdb) where 2nd argument is database
?lm
```

```
## starting httpd help server ... done
```

```r
summary(M)
```

```
##
## Call:
## lm(formula = Acceptance ~ SAT)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.9986  -7.5781  -0.8393   9.0473  26.5634
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 202.2677    31.1291   6.498 4.34e-08 ***
## SAT          -0.1301     0.0246  -5.288 3.00e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.73 on 48 degrees of freedom
## Multiple R-squared:  0.3681, Adjusted R-squared:  0.355
## F-statistic: 27.97 on 1 and 48 DF,  p-value: 2.999e-06
```

```
paste0("equation is: y hat = ", M$coefficients[2], " + ", summary(M)$coefficients[1,1], "x")
#yay both work fine
```

```
## [1] "equation is: y hat = -0.130089357268961 + 202.267744013676x"
```

2. Consider the question given in Tutorial 1.

2a) For the first question in Tutorial 1, use the code to define a function, called F1, where the argument of F1 is salary. Run function F1 for the two cases mentioned.

```
F1 <- function(salary) {
  price = 1200000
  down_payment = price*0.25 #cost of house la
  saved = 10000

  #ok the months r monthing now
  month_counter = 0
  while (saved < down_payment) {
    month_counter = month_counter + 1
    saved = saved * 1.02 #DOES NOT INCLUDE THE SALARY FOR THIS MONTH
    saved = saved + (salary * 0.4)
  }
  paste0("you need ",month_counter," months to get that house. This is longer than my will to live.")
}

F1(7000)
```

```
## [1] "you need 55 months to get that house. This is longer than my will to live."
```

```
F1(10000)
```

```
## [1] "you need 44 months to get that house. This is longer than my will to live."
```

2b) For the second question in Tutorial 1, use the code to define a function, called F2, where F2 has two arguments: salary and rate. Run function F2 for the two cases mentioned to obtain the results.

```
# Enter code here
F2 <- function(salary, rate) {
  price = 1200000 #cost of house
  down_payment = price*0.25
  saved = 10000

  #ok the months r monthing now
  month_counter = 0
  while (saved < down_payment) {
    month_counter = month_counter + 1
    saved = saved * 1.02 #DOES NOT INCLUDE THE SALARY FOR THIS MONTH
    saved = saved + (salary * 0.4)
    if (month_counter%%4 == 0) {
      salary = salary * (1 + rate)
    }
  }
  paste0("with your improved (insane) salary , you need ",month_counter," months to get that
house. This is longer than my will to live.")
}

F2(7000, 0.02)
```

```
## [1] "with your improved (insane) salary , you need 52 months to get that house. This is lo
nger than my will to live."
```

```
F2(10000, 0.01)
```

```
## [1] "with your improved (insane) salary , you need 43 months to get that house. This is lo
nger than my will to live."
```

2c) From question the settings given in Tutorial 1, we know that both the percentage of your salary that you save each month and the rate of raising salary every 4 months affects how long it takes you to save for a down payment.

Now, suppose the raise in salary every 4 months is fixed at 0.01 and you want to set a particular goal, e.g. to be able to afford the down payment in five years for a house with the price is of your choice, price. **How much should you save each month instead of 40% to achieve the goal?** In this problem, you are going to write a function, called F3, which helps to answer that question.

You are now going to **find the best propotion of savings monthly from your salary to achieve a down payment in five years**. Since hitting this exactly is a challenge, we simply want your total savings to be at least as the same as the required down payment. The proportion of saving should be of 2 decimal places.

Run function F3 and report the answers obtained for two cases: (salary = $7,000 and price = $1,200,000) and (salary = $4,000, price = $800,000).

```r
F3 <- function(salary, years_goal, price) { #can give default values for these arguments too
  #price = 1200000 #cost of house
  down_payment = price*0.25
  saved = 10000
  month_counter = years_goal * 12
  down_payment = down_payment-(10000)*(1.02^month_counter)
  print(down_payment) #debug line
  dem_interest = 0
  for (i in 1:(month_counter - 1)){
    dem_interest = dem_interest + (1.02^i)
  }
  down_payment = down_payment/dem_interest
  for (i in 1:(month_counter - 1)){
    if (month_counter%%4 == 0) {
      salary = salary * (1 + rate)
    }
  }
  final_p = ((down_payment/dem_interest)/salary)*100
  paste0("you will need to save ",final_p,"% of your salary")
}

F3(7000, 5, 1200000)
F3(4000, 5, 800000)

#ALT METHOD: try ALL proportions ie seq(0.01, 1,by = 0.01) and really try every single percen
tage until we hit a portion that takes lesser than 5 years to achieve, OR when percentage = 1
00 in which even if all your salary goes to the house and u STILL cant afford in 5yrs
```