Regression

Dian Ramadhani

08/01/2020

Regresi Linear

Analisis regresi merupakan metode untuk menentukan hubungan sebab-akibat antara satu variabel dengan variabel(-variabel) yang lain. Analisis regresi dipakai secara luas untuk melakukan prediksi dan ramalan, memahami variabel bebas mana saja yang berhubungan dengan variabel terikat, dan lain sebagainya Variabel yang mempengaruhi sering disebut sebagai variabel independen. Variabel yang dipengaruhi sering disebut dengan variabel dependen.

Install Packages

```
# Menginstall package(s)
install.packages("readr") # membaca file
install.packages("ggplot2") # visualisasi data
install.packages("here") # mengetahui direktori
```

Import Library

```
# Mengaktifkan package(s)
library(readr)
library(ggplot2)
library(here)
```

Mengetahui Direktori

```
# Mengetahui direktori proyek
here()
```

Import Data

```
# Mengimport data
df.salary <- read_csv(here("data", "raw", "regression_salary.csv"))
```

```
## Parsed with column specification:
## cols(
##   yearsexperience = col_double(),
##   absencescore = col_double(),
##   failurescore = col_double(),
##   salary = col_double()
## )
```

Data ini berisi empat variabel yaitu lamanya bekerja (yearsexperience), skor ketidakhadiran (absencescore), skor kegagalan (failurescore), dan jumlah gaji (salary).

Data “regression_salary.csv” diimpor sebagai tabel bernama “df.salary”.

Pada pekerjaan kali ini, kita akan memodelkan regresi, memprediksi data baru, hingga menyimpan hasil.

Eksplorasi Data

Data yang telah diimpor selanjutnya dieksplorasi untuk mengetahui strukturnya.

```
# Melihat attribute dan struktur data
names(df.salary) # menampilkan nama kolom
```

```
## [1] "yearsexperience" "absencescore"    "failurescore"    "salary"
```

```
dim(df.salary) # menampilkan dimensi tabel
```

```
## [1] 30  4
```

```
head(df.salary) # menampilkan beberapa data teratas
```

```
## # A tibble: 6 x 4
##   yearsexperience absencescore failurescore salary
##             <dbl>        <dbl>        <dbl>  <dbl>
## 1             1.1          0.3        1      39343
## 2             1.3          0.6        0.979  46205
## 3             1.5          0.5        0.905  37731
## 4             2            0.7        0.895  43525
## 5             2.2          0.1        0.842  39891
## 6             2.9          0.4        0.811  56642
```

```
str(df.salary) # menampilkan struktur data
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 30 obs. of  4 variables:
##  $ yearsexperience: num  1.1 1.3 1.5 2 2.2 2.9 3 3.2 3.2 3.7 ...
##  $ absencescore   : num  0.3 0.6 0.5 0.7 0.1 0.4 0.2 0.1 0.4 0.9 ...
##  $ failurescore   : num  1 0.979 0.905 0.895 0.842 ...
##  $ salary         : num  39343 46205 37731 43525 39891 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   yearsexperience = col_double(),
##   ..   absencescore = col_double(),
##   ..   failurescore = col_double(),
##   ..   salary = col_double()
##   .. )
```

```
summary(df.salary) # menampilkan rangkuman data
```

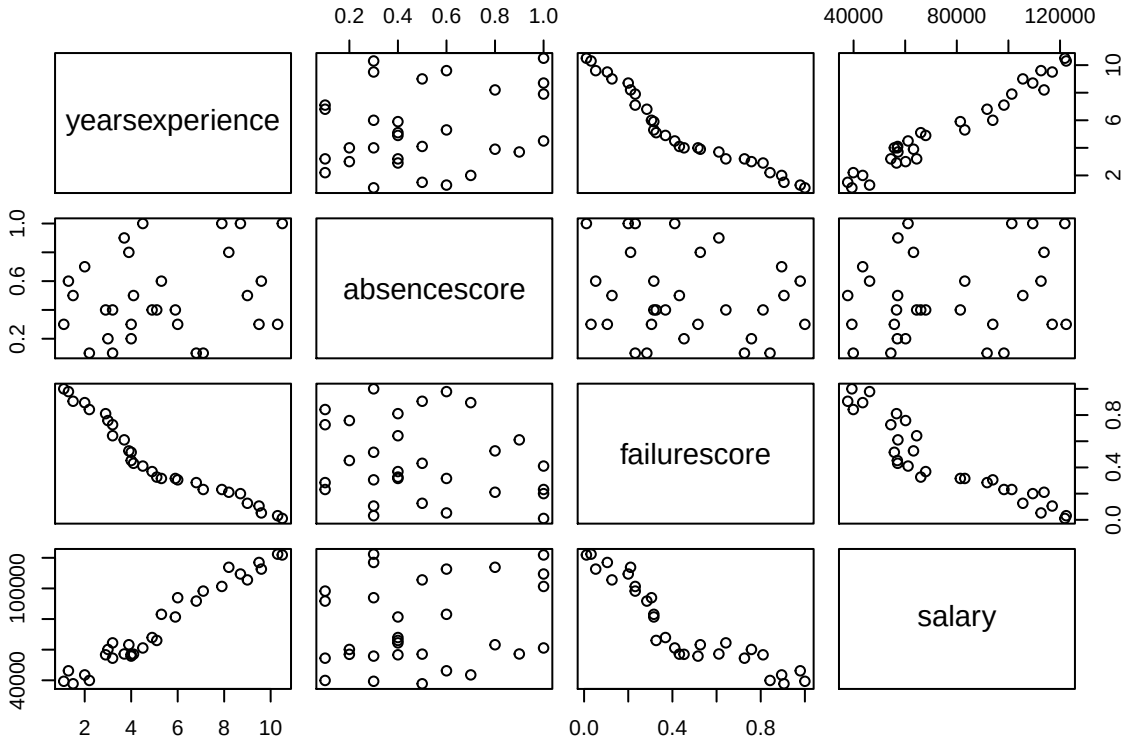
```
##  yearsexperience   absencescore     failurescore         salary
##  Min.   : 1.100   Min.   :0.1000   Min.   :0.01053   Min.   : 37731
##  1st Qu.: 3.200   1st Qu.:0.3000   1st Qu.:0.23158   1st Qu.: 56721
##  Median : 4.700   Median :0.4000   Median :0.38947   Median : 65237
##  Mean   : 5.313   Mean   :0.4933   Mean   :0.45404   Mean   : 76003
##  3rd Qu.: 7.700   3rd Qu.:0.6750   3rd Qu.:0.70526   3rd Qu.:100545
##  Max.   :10.500   Max.   :1.0000   Max.   :1.00000   Max.   :122391
```

```
# Mengetahui jumlah data kosong
sum(is.na(df.salary))
```

```
## [1] 0
```

Selain itu, data "df.salary" juga dapat dieksplore sebarannya dengan perintah sebagai berikut.

```
# Visualisasi data
pairs(df.salary)
```



Memodelkan Regresi Linear

Dalam praktek kali ini kita akan membuat model regresi linear untuk memprediksi salary berdasarkan yearsexperience, absencescore, dan failurescore. Model regresi linier dinyatakan dengan perintah lm([dependent variable] ~ [independent variable 1] + [independent variable 2] + [dst], data = df)

```
# Membuat model
lm.salary <- lm(salary ~  yearsexperience + absencescore + failurescore, data = df.salary)

# Menampilkan model regresi
lm.salary
```

```
##
## Call:
## lm(formula = salary ~ yearsexperience + absencescore + failurescore,
##     data = df.salary)
##
## Coefficients:
##     (Intercept)  yearsexperience     absencescore     failurescore
##            9065            11221            -1168            17384
```

Luaran dari perintah ini yaitu menampilkan intercept dan coefficient dari model regresi “lm.salary”.

```
# Melihat hasil anova
anova(lm.salary)
```

```
## Analysis of Variance Table
##
```

```
## Response: salary
##                 Df     Sum Sq    Mean Sq  F value Pr(>F)
## yearsexperience  1 2.0857e+10 2.0857e+10 627.9980 <2e-16 ***
## absencescore     1 1.7122e+06 1.7122e+06   0.0516 0.8222
## failurescore     1 7.2913e+07 7.2913e+07   2.1954 0.1504
## Residuals       26 8.6350e+08 3.3212e+07
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Melihat summary model
summary(lm.salary)
```

```
##
## Call:
## lm(formula = salary ~ yearsexperience + absencescore + failurescore,
##     data = df.salary)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8382  -4282  -1112   2018  12592
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)         9064      11798   0.768    0.449
## yearsexperience    11221       1243   9.025 1.72e-09 ***
## absencescore       -1168       3777  -0.309    0.760
## failurescore       17384      11732   1.482    0.150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5763 on 26 degrees of freedom
## Multiple R-squared:  0.9604, Adjusted R-squared:  0.9558
## F-statistic: 210.1 on 3 and 26 DF,  p-value: < 2.2e-16
```

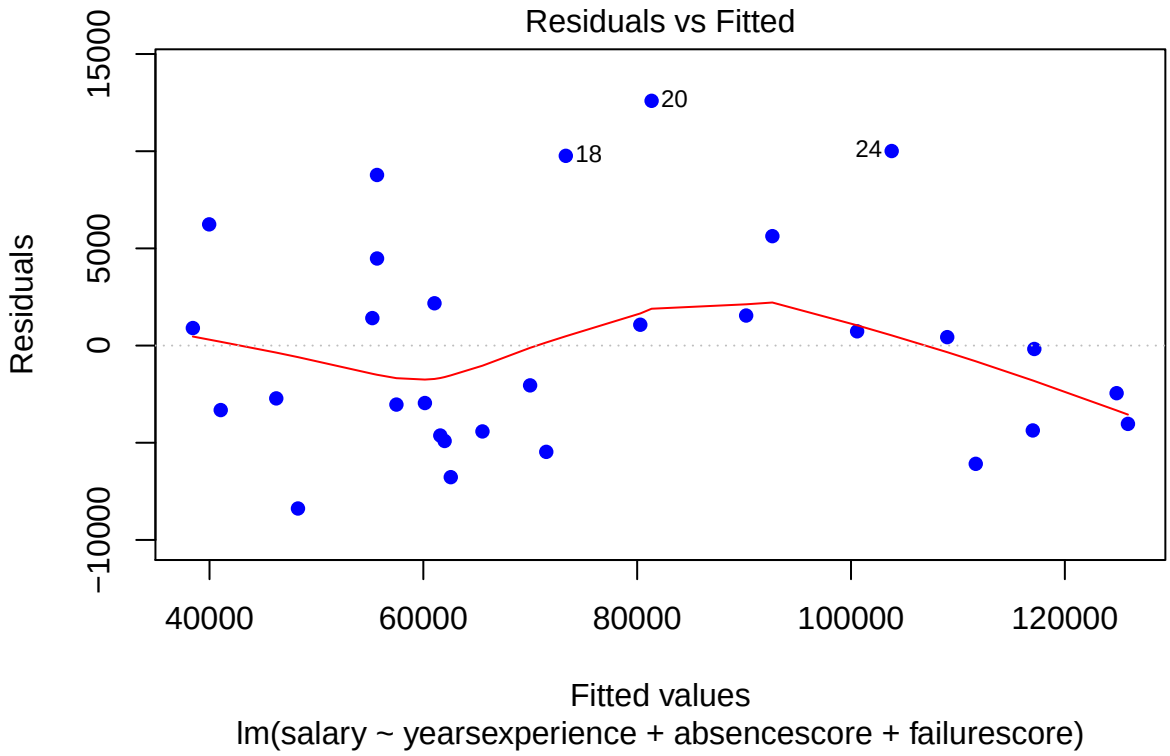
Residual

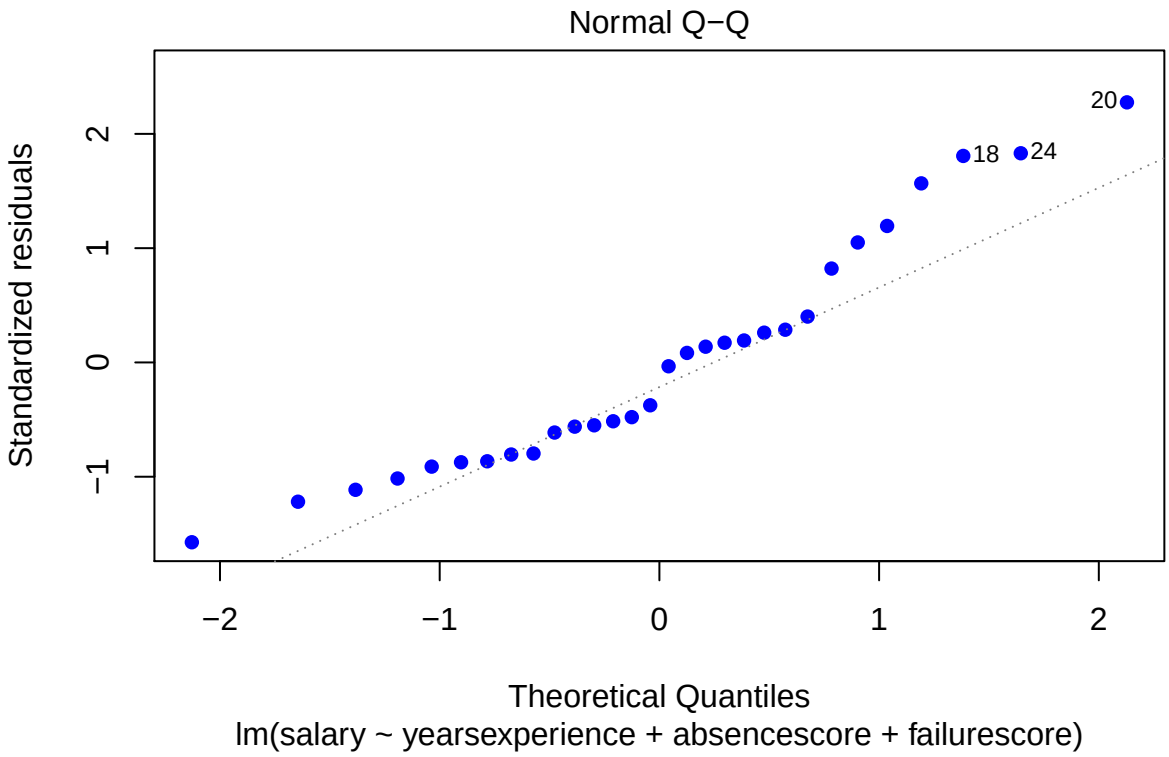
Salah satu cara untuk menguji kualitas kesesuaian model regressi adalah dengan melihat nilai residual atau perbedaan antara nilai riil dan nilai prediksi. Model yang bagus memiliki nilai residu yang kecil (mendekati nol).

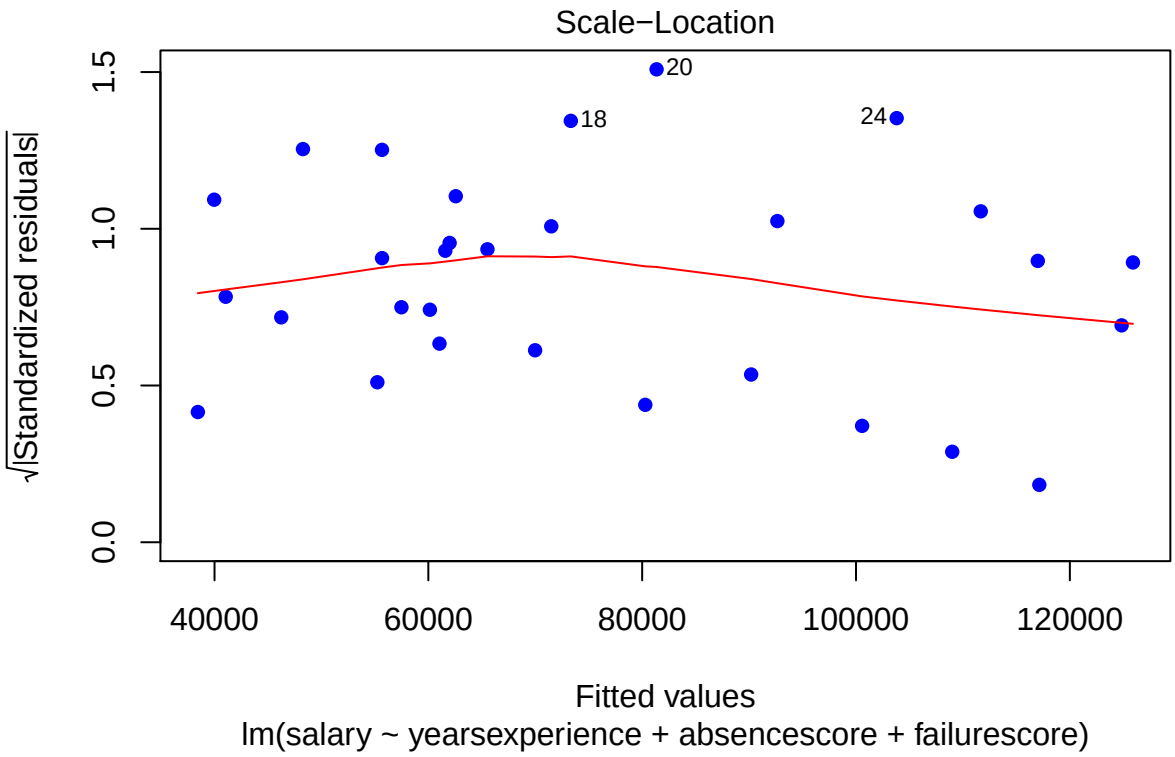
```
# Menghitung residual
residuals <- residuals(lm.salary)

# Menampilkan residual
residuals
```

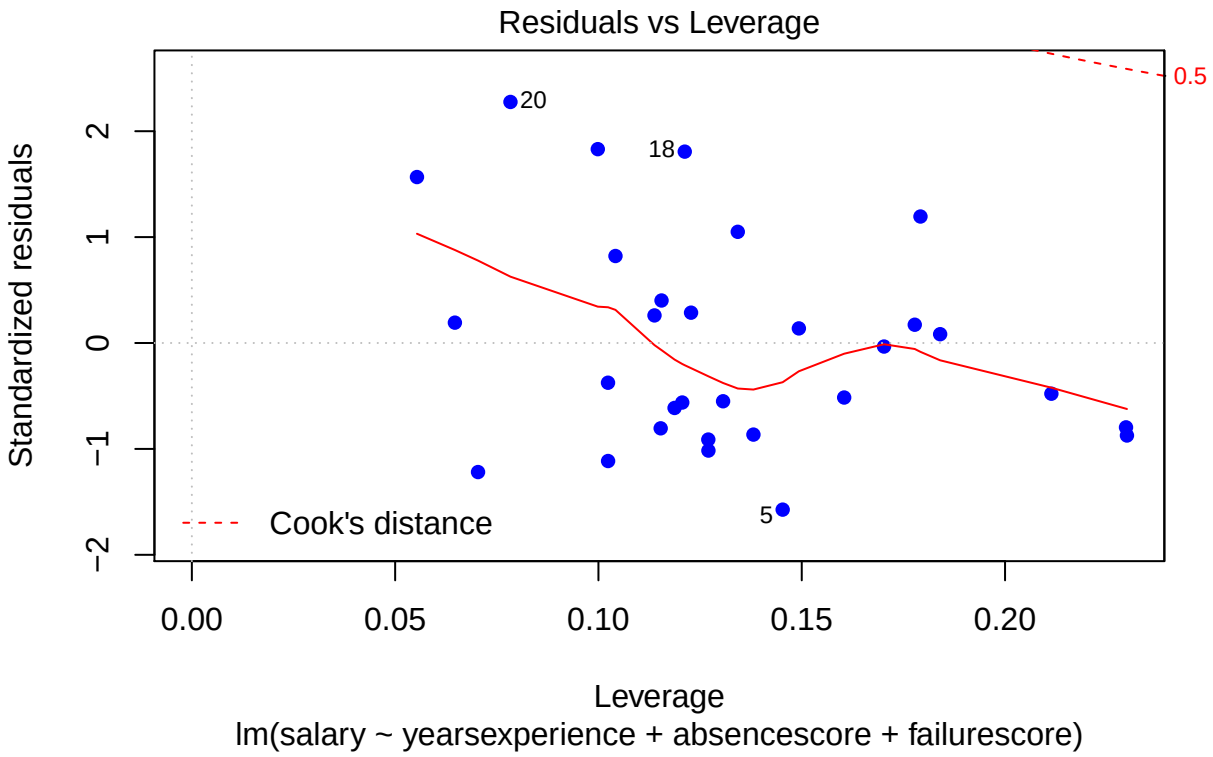
```
##          1          2          3          4          5          6
##   901.6028  6235.7709 -3318.3408 -2718.3178 -8382.4439  1413.1332
##          7          8          9         10         11         12
##  4480.3513 -3036.7190  8777.6193 -2955.9574  2175.9206 -6771.2410
##         13         14         15         16         17         18
## -4627.1101 -4908.8259 -4417.2760 -2047.6240 -5468.8977  9762.4737
##         19         20         21         22         23         24
##  1071.1629 12592.2294  1545.6656  5629.2655   732.5996 10008.6165
##         25         26         27         28         29         30
##   433.6395 -6084.8164  -176.0318 -4366.7786 -2450.0333 -4029.6371
```

```r
# diagnostic plot
plot(lm.salary, pch = 16, col = "blue")
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(salary ~ yearsexperience + absencescore + failurescore)
20
18
24
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
18
24
40000
60000
80000
100000
120000
Fitted values
lm(salary ~ yearsexperience + absencescore + failurescore)



Plot residuals vs fitted menunjukkan perbandingan residu dengan nilai fitted, dalam plot tersebut,residu mewakili jarak vertikal dari satu titik ke garis regresi. Jika semua titik berada tepat di garis regresi, semua residu akan berada tepat di garis abu-abu putus-putus. Garis merah di dalam plot adalah kurva yang berhubungan dengan residu. Jika semua titik berada tepat di garis regresi, posisi garis merah harus sama persis dengan garis abu-abu bertitik.

Plot normal Q-Q menunjukkan normal residual. Plot ini memverifikasi asumsi bahwa residu terdistribusi normal. Jadi, jika residu terdistribusi normal, mereka seharusnya terletak tepat di garis putus-putus abu-abu.

Plot scale-location digunakan untuk mengukur akar kuadrat dari residu terstandarisasi terhadap nilai fitted. Oleh karena itu, jika semua titik terletak pada garis regresi, nilai y harus mendekati nol. Karena diasumsikan bahwa varians residu tidak mengubah distribusi secara substansial, jika asumsinya benar, garis merah harus relatif datar.

Plor residuals vs leverage menunjukkan perbadingan residu standar dengan leverage. Leverage adalah pengukuran tentang bagaimana setiap titik data mempengaruhi regresi. Ini adalah pengukuran jarak dari pusat massa regresi dan tingkat isolasi (diukur dengan apakah ia memiliki tetangga). Selain itu, kita dapat melihat cook's distance, yang dipengaruhi oleh leverage tinggi dan residu besar. Kita dapat menggunakan ini untuk mengukur bagaimana regresi akan berubah jika satu titik dihapus. Garis merah halus berkenaan dengan residu standar. Untuk regresi sempurna, garis merah harus dekat dengan garis putus-putus tanpa poin lebih dari 0,5 dalam jarak Cook.

Prediksi

Membuat daftar data salary baru yang akan diprediksi.

```
# Membuat data yang akan diprediksi
df.newsalary <- read_csv(here("data", "raw", "regression_salary_predict.csv"))
```

```
## Parsed with column specification:
## cols(
##   yearsexperience = col_double(),
##   absencescore = col_double(),
##   failurescore = col_double()
## )
```

```
# Memasukkan dilai kedalam model
lm.pred <- predict(lm.salary, df.newsalary)

# Menampilkan hasil prediksi
lm.pred
```

```
##         1         2         3         4         5         6         7
##  84660.72 110883.80 141779.17 169170.32 205906.04 222030.07 252971.39
##         8         9
## 282744.64 307845.61
```

```
# Menampilkan hasil prediksi dalam bentuk tabel
df.pred <- data.frame(df.newsalary,lm.pred)
View(df.pred)
```

Save Prediction

Menyimpan hasil prediksi menjadi sebuah file excel.

```
# Memnyimpan hasil prediksi
write_csv(df.pred, here("data", "processed", "predicted_salary.csv"))
```