Regresi linear multivariat menggunakan R

Pramudita Satria Palar, Ph.D.

12/16/2021

Teori dasar

Setelah kita dapat membuat regresi linear satu variabel (lihat tutorial sebelumnya), tentunya anda juga ingin dapat membuat regresi linear untuk banyak variabel independen. Ini relevan dengan aplikasi di dunia nyata dimana anda ingin melihat hubungan antara banyak variabel independen dengan satu variabel dependen. Mengambil contoh untuk kasus mekanika fluida, kita ingin mengetahui efek simultan dari perubahan bilangan Mach dan sudut serang terhadap koefisien gaya angkat suatu airfoil pada kecepatan terbang transonik. Regresi linear multi-variabel, atau kita sebut saja *multiple linear regression* (MLR), menarik hubungan linear antara banyak variabel peubah dengan satu variabel terikat.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k + \varepsilon, \tag{1}$$

dimana k adalah jumlah variabel bebas yang digunakan dan ε adalah suku error. Perlu diingat bahwa β_0 adalah *intercept*, dan β_i dimana $i > 0$ adalah slope untuk variabel i (sebagai contoh, β_2 adalah *slope* untuk variabel ke-2). Untuk mempermudah notasi, kita akan menuliskan koefisien-koefisien MLR ini sebagai vektor $\boldsymbol{\beta}$, yakni $\boldsymbol{\beta} = \{\beta_0, \beta_1, \ldots, \beta_k\}^T$. Kita akan definisikan juga $P = k + 1$ sebagai jumlah variabel peubah ditambah 1, yang menandakan jumlah koefisien yang digunakan, atau bisa kita definisikan juga sebagai jumlah fungsi basis.

Ada banyak informasi penting yang bisa kita dapatkan dari membuat model regresi linear, seperti (1) mencari hubungan antara banyak variabel peubah dan variabel terikat (misal: jika variabel x_1 diubah, bagaimana efeknya terhadap Y?), (2) mencari variabel peubah mana yang memiliki pengaruh paling besar terhadap Y, dan manfaat-manfaat lainnya.

Mengapa kita ingin membuat model MLR? Gambar di bawah memberikan ilustrasi bagaimana model regresi linear dapat membantu kita menginterpretasikan hubungan antara dua variabel, x_1 dan x_2, dengan y. Hubungan pada gambar di bawah lebih tepatnya adalah $Y = 20 + 8x_1 + 15x_2$. Dari visualisasi dan juga bentuk persamaan, kita dapat melihat arti dari persamaan ini. Sebagai contoh, meningkatkan x_1 sebanyak satu unit akan meningkatkan y sebanyak 8 unit y. Model regresi linear seperti inilah yang akan kita coba buat dengan menggunakan R.

Fungsi $Y = 20 + 8x_1 + 15x_2$ akan kita evaluasi pada $[-1, 1]^2$ (dua variabel dengan batas bawah -1 dan batas atas 1) dan kita definisikan dalam R sebagai berikut:

```
test_function <-function(x1,x2)
{
  result = 20+8*x1+15*x2
  return(result)
}
```

Dengan visualisasi pada gambar di bawah:

```
x1 <- x2 <-seq(-1,1,length=100)
g <- expand.grid(x1,x2)
y <- matrix(test_function(g[,1],g[,2]),ncol=length(x2))
```

```r
par(mfrow=c(1,2))
persp(x1, x2, y, theta=30, phi=30, zlab="y")
image(x1,x2,y,col=heat.colors(128))
contour(x1,x2,matrix(y,ncol=length(x2)),add=TRUE)
```

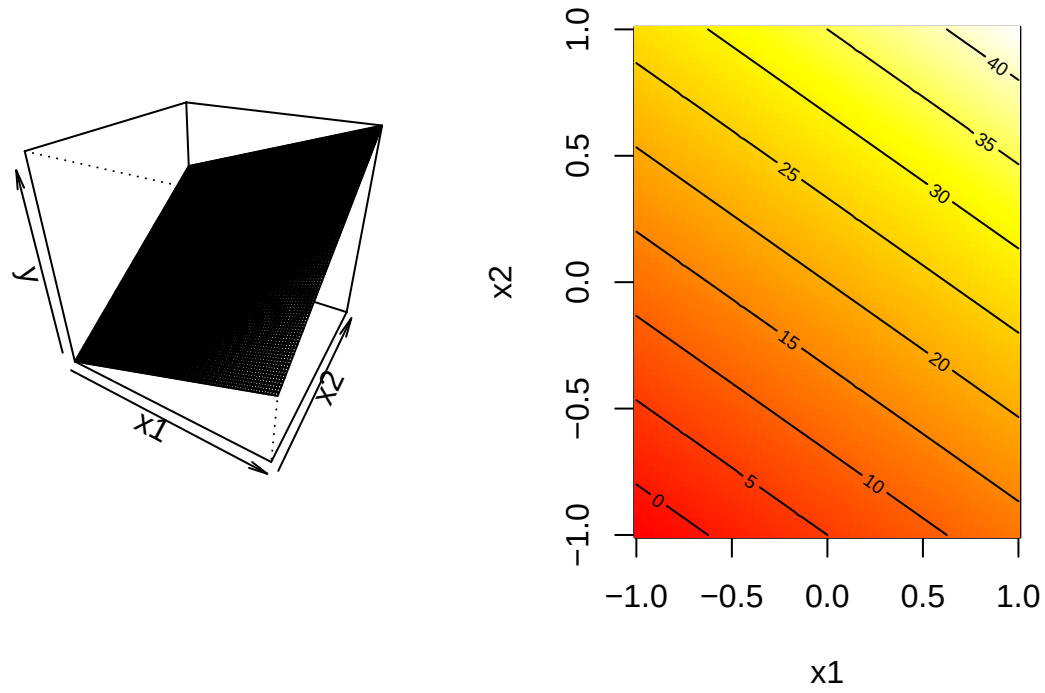

Figure 1: Visualisasi data yang digunakan dalam tutorial ini

Untuk dapat membuat model MLR, langkah pertama tentunya adalah dengan mengumpulkan data points terlebih dahulu,

$$(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i), \quad i = 1, 2, \ldots, n$$

dimana $n > k$. Setiap observasi dideskripsikan oleh model linear sebagai berikut:

$$\begin{aligned} sy_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_{ik} + \varepsilon_i \\ &= \beta_0 + \sum_{i=1}^{k} \beta_j x_{ij} + \varepsilon_i \quad i = 1, 2, \ldots, n \end{aligned}$$

Kita bisa mendapatkan

$$(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i), \quad i = 1, 2, \ldots, n$$

dengan berbagai cara, misalkan dengan eksperimen fisik (uji tarik sebagai contoh), eksperimen komputer (menggunakan *computational fluid dynamics*), ataupun hasil observasi lapangan.

Perhitungan koefisien menggunakan matriks

Pendefinisian masalah MLR akan menjadi lebih mudah jika kita menggunakan notasi matrix. Pertama-tama, kita akan mendefinisikan terlebih dahulu beberapa matrix yang akan kita gunakan. Kita mulai terlebih dahulu dengan *design matrix* $\mathbf{F}$ yang berukuran $n \times P$:

$$\mathbf{F} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix}$$

Jika kita menggunakan istilah aljabar linear, $\mathbf{F}$ merupakan hasil evaluasi dari fungsi basis $1, x_1, x_2, \dots, x_k$ pada setiap titik yang berada di data set kita. Selanjutnya kita kumpulkan hasil observasi y kita pada sebuah matrix berukuran $n \times 1$ bernama $\mathbf{y}$:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

Dilanjutkan dengan matrix berukuran $n \times 1$ bernama $\mathbf{y}$ yang bernama $\boldsymbol{\beta}$ (ini sama dengan $\boldsymbol{\beta}$ yang sudah kita definisikan sebelumnya:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \dots \\ \beta_k \end{bmatrix}$$

Sama halnya dengan kasus simple linear regression, perhitungan koefisien pada MLR dilakukan dengan cara meminimalisir residual antara observasi dan garis linear (dengan kata lain, garis linear sedekat mungkin dengan observasi).

Kita dapat mendefinisikan hubungan $\boldsymbol{y}$, $\mathbf{F}$, $\boldsymbol{\beta}$, dan $\boldsymbol{\epsilon}$ dengan

$$\boldsymbol{y} = \mathbf{F}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2}$$

sehingga

$$\boldsymbol{\epsilon} = \boldsymbol{y} - \mathbf{F}\boldsymbol{\beta}. \tag{3}$$

Selanjutnya kita akan mendefinisikan kembali *Residual Sum of Squares* (RSS) yang merupakan fungsi dari $\boldsymbol{\beta}$ dan harus diminimalisir. Ingat bahwa kita ingin meminimalisir $\sum_{i=1}^{n} \epsilon_i^2$. Dengan memberikan simbol $J(\boldsymbol{\beta})$ untuk mendefinisikan RSS, maka kita dapat mendefinisikan fungsi tujuan sebagai berikut

$$J(\boldsymbol{\beta}) = \sum_{i=1}^{n} \epsilon_i^2 = \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} (\boldsymbol{y} - \mathbf{F}\boldsymbol{\beta})^T (\boldsymbol{y} - \mathbf{F}\boldsymbol{\beta}). \tag{4}$$

Fungsi tujuan ini harus diminimalisir dengan mencari $\boldsymbol{\beta}$ yang meminimalkan $J(\boldsymbol{\beta})$, atau dengan kata lain

$$\left. \frac{\partial J(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right|_{\hat{\beta}_0, \hat{\beta}_1} = 0$$

Permasalahan ini adalah permasalahan optimasi dengan P peubah dan dapat kita formalkan menjadi

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} J(\boldsymbol{\beta}) = \arg\max_{\boldsymbol{\beta}} ||\boldsymbol{y} - \mathbf{F}\boldsymbol{\beta}||^2$$

yang mengharuskan kita menyelesaikan sistem persamaan linear berikut

$$\mathbf{F}^T\mathbf{F}\boldsymbol{\beta} = \mathbf{F}^T\boldsymbol{y}. \tag{5}$$

Kita akhirnya mendapatkan bahwa koefisien dari MLR dapat dihitung dengan menggunakan

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{F}^T\mathbf{F})^{-1}\mathbf{F}^T\boldsymbol{y}} \tag{6}$$

teknik di atas memiliki nama sendiri: *Ordinary Least Squares* (OLS). Dapat kita lihat bahwa perhitungan $\hat{\boldsymbol{\beta}}$ dilakukan dengan menyelesaikan suatu sistem persamaan linear. Perlu dicatat bahwa OLS digunakan bukan hanya pada model regresi linear saja tetapi juga model lain yang dapat didefinisikan secara linear seperti regresi polynomial.

Selain koefisien, tentunya kita juga ingin mengetahu informasi-informasi penting lain seperti R^2, confidence interval, dan prediction interval. Detil perhitungan tidak diberikan dalam tutorial singkat ini tapi akan kita lakukan perhitungannya menggunakan R.

Eksperimen menggunakan R

Mari kita membuat model MLR menggunakan R. Kita akan menggunakan data yang sudah tersedia di package `datarium` yang dapat anda tambahkan ke R anda dengan mengetikkan `install.packages(datarium)` di console R. Data yang akan kita gunakan adalah data `marketing`, yang sudah berbentuk data frame, karena mudah untuk dimengerti. Data `marketing` memiliki tiga variabel peubah yang merupakan jumlah uang dalam ribuan dollar yang digunakan untuk pengiklanan suatu produk. Tiga variabel tersebut adalah `youtube`, `facebook`, dan `sales`. Variabel terikat dalam data ini adalah `sales`, yang menandakan jumlah penjualan sebagai fungsi dari "youtube,facebook, dansales"'.

Pertama-tama, kita impor data marketing ke environment R yang sedang kita gunakan dan kemudian gunakan fungsi `head()` untuk melihat 6 baris pertama dari data `marketing`:

```
data("marketing", package = "datarium")
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
```

Agar lebih jelas, plot berikut menunjukkan plot individu antara tiga variabel peubah dengan variabel terikat. Terlihat bahwa ada tren linear yang dapat ditarik dari tiga plot ini. Akan tetapi, kita perlu sangat berhati-hati dalam melihat plot individu seperti ini dikarenakan `sales` merupakan fungsi dari perubahan tiga variabel peubah secara simultan. Cobalah eksekusi kode berikut:

```
par(mfrow=c(1,3))
plot(marketing$youtube, marketing$sales, pch=19)
plot(marketing$facebook, marketing$sales, pch=19)
plot(marketing$newspaper, marketing$sales, pch=19)
```

Tujuan kita adalah mencari model MLR dalam bentuk sebagai berikut:

$$\hat{\text{sales}} = \beta_0 + \beta_1 \times \text{ youtube} + \beta_2 \times \text{ facebook} + \beta_3 \times \text{ newspaper}$$

Di sini kita menandakan `youtube`, `facebook`, dan `newspaper` sebagai variabel pertama, kedua, dan ketiga.

Mari kita mulai dengan perhitungan secara manual.

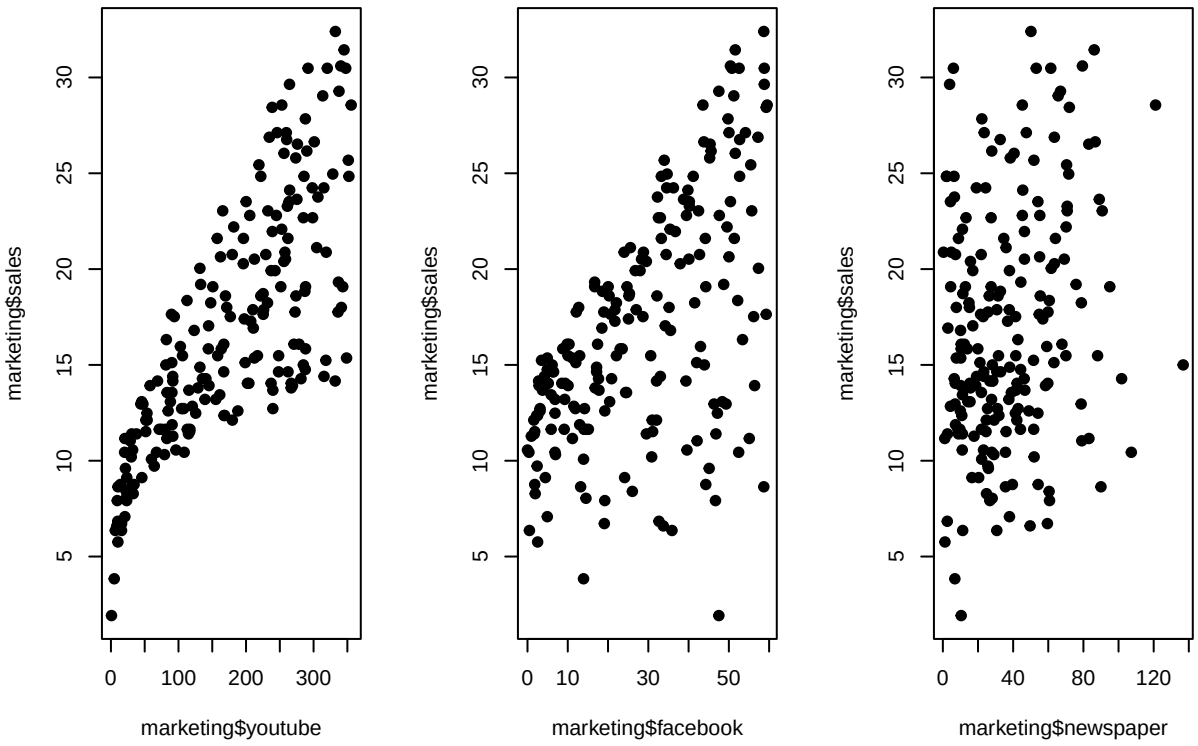

Figure 2: Visualisasi data marketing setiap variabel secara individual

Perhitungan koefisien secara manual

Perhitungan manual akan kita lakukan dengan menggunakan OLS. Untuk melakukan operasi OLS, kita pertama merubah tipe data yang awalnya berupa data frame menjadi matrix agar perhitungan matrix menjadi dapat dilakukan. Eksekusi kode berikut untuk melihat cara melakukan OLS secara manual:

```
n <- nrow(marketing) # Jumlah data
dm <- data.matrix(marketing) # Transformasi data frame menjadi matrix
F <- cbind(matrix(1,n,1),dm[,1:3])  # Membuat matriks F
ym <- dm[,4] # Membuat vektor y
coeff <- solve(t(F)%*%F) %*% t(F) %*% ym  # Menghitung koefisien
print(coeff)
```

```
##                   [,1]
##            3.526667243
## youtube    0.045764645
## facebook   0.188530017
## newspaper -0.001037493
```

Koesifien yang dihitung mengatakan bahwa model MLR yang didapatkan adalah

$$\hat{\text{sales}} = 3.525 + 0.045 \times \text{ youtube} + 0.188 \times \text{ facebook} - 0.001 \times \text{ newspaper}$$

Model MLR di atas mengatakan bahwa pengiklanan melalui facebook memiliki pengaruh paling besar terhadap sales, sementara yang melalui koran hampir tidak memiliki pengaruh sama sekali. Tentunya kita butuh melakukan analisis lebih dalam untuk mengetahui informasi-informasi penting seperti standard error dari koefisien. Kita akan langsung menggunakan `lm` untuk keperluan tersebut.

Membuat model MLR dengan `lm()`

Pembuatan model MLR dengar R mengikuti cara yang sama seperti saat kita melakukan simple linear regression, yaitu dengan fungsi `lm()`. Perbedaan utamanya adalah saat kita mendefinisikan model linear yang akan dibuat karena MLR melibatkan lebih dari satu variabel. Kita akan menamakan model yang kita buat dengan `lm()` sebagai MLRmod.

Untuk data `marketing`, sintaks paling sederhana untuk membuat model MLR adalah sebagai berikut:

```
MLRmod = lm(sales~youtube+facebook+newspaper, marketing)
```

Kita juga dapat menuliskan kodenya sebagai berikut untuk memudahkan dalam membaca kode.

```
MLRmod = lm(data = marketing, formula = sales~youtube+facebook+newspaper)
```

Mari kita cetak hasil dari model regresi ini. Hasil koefisien yang anda lihat adalah sama persis seperti dengan jika kita lakukan secara manual:

```
print(MLRmod)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
## Coefficients:
## (Intercept)      youtube     facebook    newspaper
##    3.526667     0.045765     0.188530    -0.001037
```

Kita dapat mencetak semua informasi penting dengan fungsi `summary()`:

```r
summary(MLRmod)
```

```
##
## Call:
## lm(formula = sales ~ youtube + facebook + newspaper, data = marketing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5932  -1.0690   0.2902   1.4272   3.3951
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.526667   0.374290   9.422   <2e-16 ***
## youtube      0.045765   0.001395  32.809   <2e-16 ***
## facebook     0.188530   0.008611  21.893   <2e-16 ***
## newspaper   -0.001037   0.005871  -0.177     0.86
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.023 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

Cara membaca hasil ringkasan di atas adalah sama seperti saat kita membuat model regresi linear satu variabel. Jika kita melihat p-value yang sangat kecil dari `youtube` dan `facebook`, maka kita dapat mengetahui bahwa hipotesis nol $\mathcal{H}_\prime : \beta_1 = 0$ dan $\mathcal{H}_\prime : \beta_2 = 0$ dapat ditolak untuk kedua variabel peubah ini (bahkan dengan significance level yang kecil sekalipun). Dengan kata lain, `youtube` dan `facebook` bisa dikatakan memiliki pengaruh linear terhadap `sales`!. Penjualan melalui media koran (`newspaper`) memberikan p-value sebesar 0.86, dimana angka ini sangat jauh dari 5% significance level sekalipun (artinya: gagal menolak $\mathcal{H}_\prime : \beta_3 = 0$). Kita pun dapat dengan aman mengatakan bahwa pengiklanan melalui koran tidak memiliki pengaruh linear terhadap penjualan. Oleh karena itu, analis dapat merekomendasikan ke klien mereka bahwa pengiklanan lebih baik dilakukan lewat Youtube dan Facebook karena memiliki pengaruh yang besar dan jelas terhadap penjualan. Kita juga mengetahui bahwa $\mathcal{H}_\prime : \beta_0 = 0$ dapat ditolak. Dapat dilihat juga harga R^2 cukup tinggi dan ini menandakan bahwa model regresi linear yang dibuat adalah cukup akurat.

Menghitung prediksi

Cara perhitungan confidence dan prediction interval menggunakan R sama saja baik untuk model regresi linear satu ataupun multivariabel, yaitu dengan menggunakan fungsi `predict()`. Kita ambil contoh sederhana terlebih dahulu yaitu jika kita ingin mengetahui berapa hasil prediksi regresi linear pada kombinasi berikut: Youtube = 175, Facebook = 28, dan newspaper = 60. Pertanyaan yang ingin anda jawab adalah: (1) berapa hasil prediksi, dan juga (2) confidence dan prediction interval dari prediksi tersebut.

```r
xpred = data.frame(youtube=175, facebook=28, newspaper=60)
ypred = predict(MLRmod, xpred)
print(ypred)
```

```
##        1
## 16.75207
```

Bagaimana jika anda ingin melakukan banyak prediksi sekaligus? Satu cara adalah dengan membuat vektor atau list, misalkan jika anda ingin melakukan prediksi pada tiga titik berbeda:

```r
xpredm = data.frame(youtube=c(175,100,60), facebook=c(28,40,32), newspaper=c(60,54,12))
ypredm = predict(MLRmod, xpredm)
print(ypredm)
```

```
##        1        2        3
## 16.75207 15.58831 12.29306
```

Menghitung confidence interval dari koefisien

Perhitungan confidence interval dari *intercept* dan *slope* dapat dilakukan dengan menggunakan fungsi `confint()` seperti contoh berikut:

```
confint(MLRmod,level=0.99) # Confidence level = 99%
```

```
##                   0.5 %     99.5 %
## (Intercept)  2.55308486 4.50024963
## youtube      0.04213632 0.04939297
## facebook     0.16613095 0.21092909
## newspaper   -0.01630884 0.01423386
```

Jika kita ingin menunjukkan hanya satu koefisien aja, untuk Youtube sebagai contoh, anda perlu menambahkan argumen `parm` di `confint()`. Sebagai contoh:

```
confint(MLRmod,level=0.99,parm="youtube") # Anda juga dapat mengetikkan "parm=2"
```

```
##              0.5 %     99.5 %
## youtube 0.04213632 0.04939297
```

Menghitung prediksi dengan confidence dan prediction interval

Menghitung confidence interval dan prediction interval dapat dilakukan dengan menambahkan `interval="confidence"` atau `interval="prediction"` di dalam `predict()`, seperti contoh berikut:

```
xpredmci = data.frame(youtube=c(175,100,60), facebook=c(28,40,32), newspaper=c(60,54,12))
ypredmci = predict(MLRmod, xpredmci,interval="confidence")
print(ypredmci)
```

```
##        fit      lwr      upr
## 1 16.75207 16.36170 17.14244
## 2 15.58831 15.15964 16.01698
## 3 12.29306 11.76690 12.81922
```

```
xpredmpi = data.frame(youtube=c(175,100,60), facebook=c(28,40,32), newspaper=c(60,54,12))
ypredmpi = predict(MLRmod, xpredmci,interval="prediction")
print(ypredmpi)
```

```
##        fit       lwr      upr
## 1 16.75207 12.744137 20.76001
## 2 15.58831 11.576463 19.60015
## 3 12.29306  8.269627 16.31649
```

Penutup

Demikian akhir dari tutorial regresi linear multivariant ini. Semoga dapat membantu anda memahami apa itu regresi linear multivariat dan bagaimana cara membuat regresi linear multivariat dalam R. Tentunya tutorial ini tidak menampilkan semua aspek dari regresi linear multivariat (kita tidak menampilkan plot residual sebagai contoh) tetapi harapannya dapat memberikan gambaran awal untuk kita mengenai regresi linear multivariat.