Tutorial Regresi linear sederhana menggunakan R

Pramudita Satria Palar

12/8/2021

Teori dasar

Model regresi linear

Regresi linear mungkin bisa dikatakan sebagai salah satu model regresi yang paling banyak digunakan di berbagai macam aplikasi. Dengan regresi linear, dan sama halnya dengan model regresi lainnya, kita ingin memprediksi angka Y berdasarkan satu atau lebih variabel prediktor X. Khusus untuk masalah satu variabel prediktor, kita memiliki nama tersendiri untuk regresi linear yang akan kita gunakan: *Simple linear regression*.

Pertama-tama, kita definisikan terlebih dahulu bahwa satu angka Y dapat didefinisikan sebagai penjumlahan dari fungsi linear dan suku error yang bersifat acak (*random*) ε:

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

dimana β_0 adalah *intercept* dan β_1 adalah *slope*. Sesuai dengan namanya, *slope* adalah kemiringan dari garis regresi linear yang akan kita buat. Sementara itu, β_0 adalah titik dimana garis linear memotong sumbu y. Dalam bahasa yang lebih umum, β_0 dan β_1 kita sebut dengan *regression coefficients*. Penggunaan istilah *regression coefficients* menjadi penting terutama ketika membahas *multiple linear regression* ataupun model regresi nonlinear seperti regresi polynomial.

Ingat bahwa adanya suku acak ε membuat kita dapat mendefinisikan (dengan sebelumnya mengasumsikan bahwa $\mathbb{E}(\varepsilon) = 0$) hubungan berikut

$$\mathbb{E}(Y|x) = \mu_{Y|x} = \beta_0 + \beta_1 x, \tag{1}$$

yaitu model regresi linear yang akan kita gunakan dalam praktek. Model regresi ini kita lihat sebagai ekspektasi dari Y kondisional terhadap suatu x tertentu. Dalam regresi linear, kita juga membuat asumsi bahwa error terdistribusi secara normal dengan mean berharga 0 dan varians berharga σ^2; dengan kata lain, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, atau $\mathbb{E}(\varepsilon) = 0$ dan $\mathbb{V}(\varepsilon = \sigma^2)$. Asumsi ini memudahkan prosedur selanjutnya karena kita menjadi dapat melakukan perhitungan standard error, uji hipotesis, dan hal-hal penting lain seperti *prediction interval*. Dengan meninjau bahwa

$$\mathbb{E}(Y|x) = \mathbb{E}(\beta_0 + \beta_1 x + \varepsilon) = \beta_0 + \beta_1 x + \mathbb{E}(\varepsilon) = \beta_0 + \beta_1 x, \tag{2}$$

dan

$$\mathbb{V}(Y|x) = \mathbb{V}(\beta_0 + \beta_1 x + \varepsilon) = \mathbb{V}(\beta_0 + \beta_1 x) + \mathbb{V}(\varepsilon) = 0 + \sigma^2 = \sigma^2. \tag{3}$$

Maka menjadi jelas bahwa model regresi yang kita buat adalah garis angka mean dengan variabilitas yang ditentukan oleh *error variance* σ^2.

Perhitungan koefisien

Kita membutuhkan data untuk membuat suatu model regresi linear. Data ini datang dalam bentuk sejumlah n pasangan antara x dan y, yakni $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$. Setiap observasi data ini dapat kita lihat menggunakan hubungan sebagai berikut

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \tag{4}$$

dimana i menandakan data ke-i. Kita kemudian definisikan residual (e) sebagai perbedaan antara data yang diberikan dengan hasil model prediksi regresi linear. Untuk data ke-i, maka e_i dapat kita tulis sebagai berikut

$$e_i = y_i - \hat{y}_i, \tag{5}$$

dimana $\hat{y}_i$ adalah hasil prediksi model regresi linear untuk data ke-i. Residual ini menandakan **kedekatan** antara data dan hasil prediksi, oleh karena itu menjadi wajar bagi kita untuk mencari model regresi linear yang hasilnya adalah sedekat mungkin dengan data yang kita miliki. Karena kita memiliki n data, maka model regresi kita harus sedekat mungkin dengan n data tersebut, sebagaimana dapat kita definisikan menjadi

$$\varepsilon_{RSS} = \sum_{i=1}^{n} e_i^2, \tag{6}$$

dimana RSS menandakan *residual sum of squares* (RSS). Mengapa residual harus dipangkatkan? Ini karena fakta bahwa e dapat berharga negatif atau positif bukanlah perbedaan penting saat kita membuat model regresi. Model regresi linear kita harus mengurangi residual tanpa membedakan negatif ataupun positif.

Pada prinsipnya, koefisien β_0 dan β_1 dicari agar RSS dapat diminimalisir sekecil mungkin. Ingat bahwa $e_i = y_i - \beta_0 - \beta_1 x_1$, sehingga

$$\varepsilon_{RSS} = \sum_{i=1}^{n} (e_i = y_i - \beta_0 - \beta_1 x_i)^2, \tag{7}$$

Dalam bahasa Kalkulus, maka β_0 dan β_1 harus dicari sedemikian rupa agar turunan parsial berikut berharga nol:

$$\left.\frac{\partial \varepsilon_{RSS}}{\partial \beta_0}\right|_{\hat{\beta}_0, \hat{\beta}_1} = 0$$

$$\left.\frac{\partial \varepsilon_{RSS}}{\partial \beta_1}\right|_{\hat{\beta}_0, \hat{\beta}_1} = 0$$

Dengan kata lain, harga β_0 dan β_1 yang memberikan turunan parsial tersebut berharga nol adalah solusi dari permasalahan optimasi yang membuat ε_{RSS} mencapai harga minimum. Kalkulus sederhana mengatakan bahwa

$$\left.\frac{\partial \varepsilon_{RSS}}{\partial \beta_0}\right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) = 0 \tag{8}$$

dan

$$\left.\frac{\partial \varepsilon_{RSS}}{\partial \beta_1}\right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) x_i = 0. \tag{9}$$

Persamaan di atas dapat kita sederhanakan menjadi

$$n\hat{\beta}_0 + \hat{\beta}_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{10}$$

dan

$$\hat{\beta}_0 \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i x_i, \tag{11}$$

yang kita sebut sebagai **least squares normal equations**. Solusi dari persamaan ini adalah sebagai berikut

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \tag{12}$$

$$\hat{\beta}_1 = \frac{S_{xx}}{S_{xy}} \frac{\sum_{i=1}^{n} y_i x_i - \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n}}{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}, \tag{13}$$

dimana $\bar{y} = (1/n)\sum_{i=1}^{n} y_i$ dan $\bar{x} = (1/n)\sum_{i=1}^{n} x_i$,

Langkah selanjutnya adalah menentukan angka-angka penting lain seperti R^2, *prediction interval*, *confidence interval*, dan hasil uji hipotesis terhadap koefisien. Detilnya tidak akan dibahas di tutorial ini, tetapi kita dapat mendapatkan informasi tersebut menggunakan modul yang dimiliki R.

Aplikasi pada data sederhana

Gambar di bawah menunjukkan visualisasi dari data yang akan kita gunakan pada tutorial ini. Gambar ini dengan jelas menunjukkan bahwa ada tren linear antara level Hydrocarbon dan juga Oxygen Purity. Tugas kita selanjutnya adalah menentukan garis regresi linear mana yang terbaik dalam mendekati data ini. Selain itu, setelah kita membuat model, kita juga harus mengetahui ketidakpastian dari formula regresi linear tersebut.

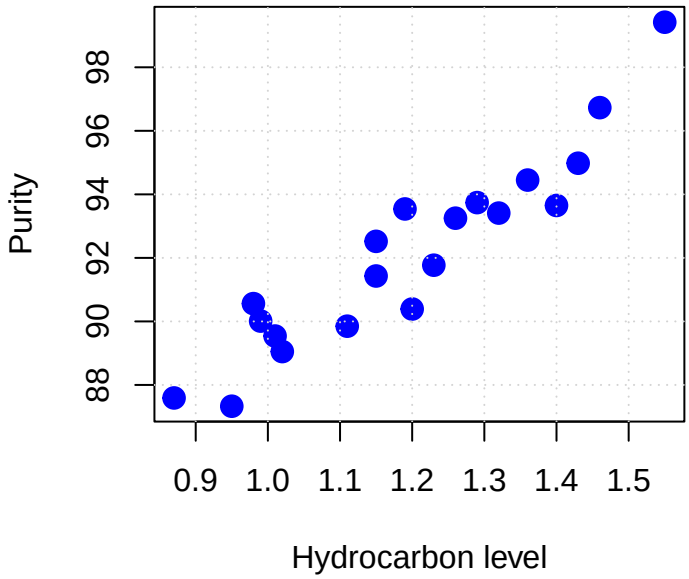

Figure 1: Visualisasi data yang digunakan dalam tutorial ini

Mendefinisikan data

Kita mulai dengan mendefinisikan data kita sebagai berikut, dimana kita mendefinisikan X sebagai variabel `x` dan Y sebagai variabel `y` pada R. Pada tutorial sederhana ini, kita akan mendefinisikan `X` dan `Y` sebagai list bilangan numerik terlebih dahulu. Nantinya kita akan menggunakan format data frame yang lebih nyaman untuk digunakan. Mari kita mulai dengan mengeksekusi potongan kode berikut:

```
X <- c(0.99, 1.02, 1.15, 1.29, 1.46, 1.36, 0.87, 1.23, 1.55, 1.4, 1.19, 1.15, 0.98,
       1.01, 1.11, 1.2, 1.26, 1.32, 1.43, 0.95)
```

```r
Y <- c(90.01, 89.05, 91.43, 93.74, 96.73, 94.45, 87.59, 91.77, 99.42, 93.65,
       93.54, 92.52, 90.56, 89.54, 89.85, 90.39, 93.25, 93.41, 94.98, 87.33)
```

Perhatikan bahwa `c()` adalah cara kita membuat vektor atau list dari bilangan numerik (walau `c()` juga dapat digunakan untuk tipe data non-numerik).

Dengan menggunakan data di atas dan persamaan-persamaan analitikal untuk menghitung koefisien, bentuk regresi linear yang didapatkan adalah

$$Y = 74.28 + 14.95X.$$

Sekarang kita akan mendapatkan bentuk ini dengan menggunakan R, baik dengan menggunakan perhitungan sendiri ataupun *built-in* function dari R. Kita akan memulai dengan melakukan perhitungan menggunakan R secara manual:

```r
n <- length(Y) # Jumlah data

S_xy <- (sum(X*Y)-((sum(Y)*sum(X))/n)) # Perhitungan S_xy
S_xx <- (sum(X^2)-(sum(X)^2)/n) # Perhitungan S_xx

# Calculate beta_1 and beta_0
beta_1 <- S_xy/S_xx # Beta (slope)
beta_0 <- mean(Y)-beta_1*mean(X) # Beta 0 (intercept)

cat("Harga beta_0 adalah ", beta_0, "\n")
```

```
## Harga beta_0 adalah  74.28331
```

```r
cat("Harga beta_0 adalah ", beta_1, "\n")
```

```
## Harga beta_0 adalah  14.94748
```

Perhitungannya sangat mudah dan kita pun telah mendapatkan solusi yang benar. Akan tetapi, anda harus mengetikkan lagi kode-kode yang dibutuhkan untuk menghitung, misal, *confidence* dan *prediction interval*. Untungnya R sudah menyediakan modul khusus untuk membuat model regresi linear, yaitu `lm()`.

Sintaks dan pembuatan model regresi linear

Kita akan menggunakan fungsi bawaan dari R untuk melakukan regresi linear. Fungsi utama yang akan kita gunakan untuk model regresi linear satu dimensi adalah `lm()`, dengan sintaks sebagai berikut:

```r
linregmod <- lm(Y~X) # Buat model regresi bernama `linregmod`
```

Pada potongan kode di atas, kita membuat model regresi linear yang kita beri nama `linregmod` dengan menggunakan `lm()`. Lebih jelasnya, kita mendefisinikan bahwa model linear kita berbentuk $Y = \beta_0 + \beta_1 x$ (`Y~X`) (*intercept* sudah otomatis dimasukkan tanpa perlu diketikkan lagi).

Anda kemudian dapat mengecek model regresi linear yang sudah dibuat. Dimulai dengan mencetak koefisien-koefisien dari model tersebut:

```r
print(linregmod) # Cetak koefisien dan juga formula dari model regresi linear
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Coefficients:
## (Intercept)            X
##       74.28        14.95
```

Uji hipotesis, standard error, dan coefficient of determination

Tentunya anda ingin mengetahui model regresi anda lebih lanjut dari sekedar angka koefisien. Beberapa kuantitas penting lain yang biasanya kita inginkan adalah *standard error* dan juga hasil uji hipotesis dengan menggunakan *t-test*. Kuantitas lain yang biasanya anda juga ingin ketahui adalah *coefficient of determination* (R^2) dan juga Adjusted-R^2. Anda dapat menggunakan fungsi `summary()` yang anda terapkan ke model regresi anda untuk mendapatkan informasi tersebut. Cobalah dengan mengeksekusi potongan kode di bawah:

```
summary(linregmod) # Cetak informasi lain dari model regresi linear
```

```
##
## Call:
## lm(formula = Y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83029 -0.73334  0.04497  0.69969  1.96809
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   74.283      1.593   46.62  < 2e-16 ***
## X             14.947      1.317   11.35 1.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 18 degrees of freedom
## Multiple R-squared:  0.8774, Adjusted R-squared:  0.8706
## F-statistic: 128.9 on 1 and 18 DF,  p-value: 1.227e-09
```

Dengan fungsi `summary()`, anda bisa langsung mendapatkan banyak informasi penting yang mencakup: (1) Formula regresi linear, (2) residual, (3) koefisien beserta dengan standard error, *t-value*, dan juga *p-value* sebagai hasil dari uji hipotesis, (4) *Coefficient of determination* (baik R^2 maupun *adjusted*-R^2), dan (5) hasil dari uji ANOVA. Diberikan juga tingkat seberapa signifikan hasil dari uji hipotesis dengan menggunakan petunjuk `Signif. codes`. Sebagai contoh, jika kode `*` ditunjukkan pada harga satu koefisien, ini berarti bahwa harga p-value yang didapat adalah kurang dari 0.05; dengan kata lain null-hypothesis ditolak dengan setidaknya level signifikansi 5%.

Angka residual yang didapatkan adalah selisih antara data dan model regresi linear yang dibuat. Model regresi linear yang telah dibuat menunjukkan bahwa null hypothesis bahwa $\beta_0 = 0$ dan $\beta_1 = 0$ ditolak, dimana yang terakhir berarti bahwa level hydrocarbon memang memiliki pengaruh terhadap *purity*. Dalam kasus ini, null hypothesis bahwa $\beta_0 = 0$ belum memiliki arti terlalu penting. Standard error di sini diinterpretasikan sebagai ketidakpastian dari β_0 dan β_1, yang berarti bahwa garis linear yang asli bisa saja lebih curam ataupun lebih landai, dan angka *intercept* yang asli bisa saja lebih tinggi ataupun lebih rendah. Angka R^2 yang didapat juga cukup tinggi, menandakan bahwa model linear dapat dipakai untuk data ini.

Menggunakan format data frame

Mendefinisikan data frame

Ada kalanya kita ingin membuat data kita dalam bentuk tabel atau matrix. Untuk itu, kita akan membuat data kita dalam bentuk data frame dengan menggunakan fungsi `data.frame()`:

```
DATA = data.frame(
  Hydrocarbon = c(0.99, 1.02, 1.15, 1.29, 1.46, 1.36, 0.87, 1.23, 1.55, 1.4, 1.19, 1.15, 0.98,
       1.01, 1.11, 1.2, 1.26, 1.32, 1.43, 0.95),
  Purity = c(90.01, 89.05, 91.43, 93.74, 96.73, 94.45, 87.59, 91.77, 99.42, 93.65,
       93.54, 92.52, 90.56, 89.54, 89.85, 90.39, 93.25, 93.41, 94.98, 87.33)
```

```
  )
```

Anda dapat melihat isi dari `DATA` dengan mengetikkan `DATA` pada console R studio:

```
DATA
```

```
##    Hydrocarbon Purity
## 1         0.99  90.01
## 2         1.02  89.05
## 3         1.15  91.43
## 4         1.29  93.74
## 5         1.46  96.73
## 6         1.36  94.45
## 7         0.87  87.59
## 8         1.23  91.77
## 9         1.55  99.42
## 10        1.40  93.65
## 11        1.19  93.54
## 12        1.15  92.52
## 13        0.98  90.56
## 14        1.01  89.54
## 15        1.11  89.85
## 16        1.20  90.39
## 17        1.26  93.25
## 18        1.32  93.41
## 19        1.43  94.98
## 20        0.95  87.33
```

Anda pun dapat melihat isi dari kolom pada `DATA` menggunakan simbol `$` yang mengikuti nama variabel data frame yang ingin dilihat dalamnya. Misalkan untuk melihat Hydrocarbon maka anda perlu mengetikkan `DATA$Hydrocarbon`. Mari kita lihat contoh di bawah:

```
DATA$Hydrocarbon # Tampilkan data Hydrocarbon
```

```
##  [1] 0.99 1.02 1.15 1.29 1.46 1.36 0.87 1.23 1.55 1.40 1.19 1.15 0.98 1.01 1.11
## [16] 1.20 1.26 1.32 1.43 0.95
```

```
DATA$Purity # Tampilkan data purity
```

```
##  [1] 90.01 89.05 91.43 93.74 96.73 94.45 87.59 91.77 99.42 93.65 93.54 92.52
## [13] 90.56 89.54 89.85 90.39 93.25 93.41 94.98 87.33
```

Membuat model regresi dari data frame

Pertama, anda harus mendefinisikan terlebih dahulu data yang akan anda gunakan dalam bentuk data frame dengan cara mengetikkan `data=DATA` di dalam `lm()`. Ini menandakan bahwa untuk pemrosesan model regresi linear anda akan menggunakan variabel `DATA`. Langkah selanjutnya anda perlu menentukan formula dari regresi linear yang anda buat, yaitu

$$\text{Purity} = \beta_0 + \beta_1 \times \text{ Hydrocarbon}.$$

Kita akan mendefinisikan model regresi baru, yang dari segi isi sebenarnya sama saja dengan `linregmod`, bernama `linregmod2`:

```
linregmod2 = lm(data=DATA, formula=Purity~Hydrocarbon)
```

Mari kita cetak hasil dari model regresi ini:

```r
print(linregmod2)
```

```
##
## Call:
## lm(formula = Purity ~ Hydrocarbon, data = DATA)
##
## Coefficients:
## (Intercept)  Hydrocarbon
##       74.28        14.95
```

Anda pun juga dapat mencetak semua informasi penting dengan fungsi `summary()`:

```r
summary(linregmod2)
```

```
##
## Call:
## lm(formula = Purity ~ Hydrocarbon, data = DATA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83029 -0.73334  0.04497  0.69969  1.96809
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   74.283      1.593   46.62  < 2e-16 ***
## Hydrocarbon   14.947      1.317   11.35 1.23e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.087 on 18 degrees of freedom
## Multiple R-squared:  0.8774, Adjusted R-squared:  0.8706
## F-statistic: 128.9 on 1 and 18 DF,  p-value: 1.227e-09
```

Membuat plot regresi linear

Plot sederhana

Selain membuat model regresi linear, tentunya anda juga ingin menampilkan model regresi yang telah dibuat bersama dengan data-data yang dimiliki. Kita dapat menggunakan fungsi `plot()` sederhana untuk menampilkan hubungan tersebut. Tidak ada cara yang baku untuk menampilkan plot ini (anda bisa memakai library khusus seperti `ggplot` sebagai contoh). Tutorial ini akan menggunakan cara yang paling sederhana dengan fungsi bawaan dari R.

Prediksi akan kita lakukan dengan menggunakan fungsi 'predict()'. Fungsi 'predict()' membutuhkan model regresi yang telah didefinisikan dan juga lokasi dimana prediksi akan dilakukan dalam bentuk data frame. Kita akan menggambarkan garis regresi linear beserta dengan data yang dimiliki pada rentang Hydrocarbon dari 0.7 sampai 1.9, sebagai berikut:

```r
xnew = data.frame(Hydrocarbon = seq(0.7,1.9,0.05))
ynew = predict(linregmod2,xnew)
plot(X,Y,pch=19,col="blue",cex=1.5,xlab = "Hydrocarbon level (x)",ylab="Purity (y)"
     ,xlim=c(0.7, 1.9))
lines(xnew$Hydrocarbon, ynew, col="red",type = "l",lwd = 2)
grid()
```

Pada potongan kode di atas, `xnew` adalah data frame yang berisi titik Hydrocarbon dimana prediksi Purity akan dilakukan, `ynew` adalah hasil dari prediksi regresi linear pada `xnew`, `plot()` adalah fungsi yang kita

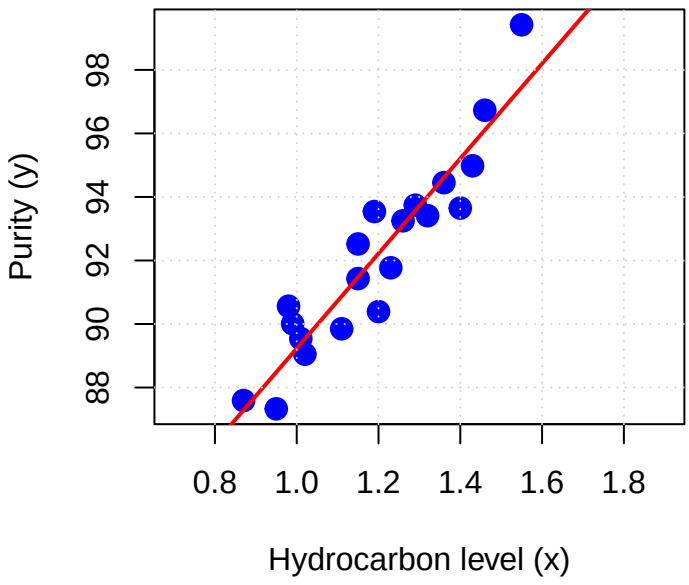

Figure 2: Visualisasi model regresi linear yang telah dibuat

gunakan untuk membuat plot, `lines()` adalah untuk menambahkan garis regresi linear, dan `grid()` adalah untuk menampilkan grid. Kita juga menggunakan fungsi `seq()` untuk membuat deret numerik dari 0.7 sampai 1.9 dengan jarak antar bilangan di deret adalah 0.05.

Plot dengan prediction dan confidence interval

Seperti yang telah kita bahas sebelumnya, garis regresi linear sebenarnya adalah sesuatu yang tidak kita ketahui sehingga menjadi penting bagi kita untuk melihat *confidence interval* dari model regresi kita. Tentunya kita dapat melihat standard error dari β_0 dan β_1 dan kemudian memakai informasi tersebut untuk menghitung *confidence interval*. Akan tetapi, tentu saja kita dapat membuat plot regresi linear dengan ikut menampilkan confidence interval di plot tersebut. Anda dapat mengekstrak informasi *confidence* ataupun *prediction interval* dengan menambahkan argumen tambahan di `predict()` yaitu `interval=`, yang dapat anda isi dengan `"confidence"` ataupun `"prediction"`. Sebagai contoh, mari kita mulai dengan *confidence interval*:

```
ynew = predict(linregmod2,xnew,interval="confidence")
print(ynew)
```

```
##          fit       lwr       upr
## 1   84.74655  83.28255  86.21055
## 2   85.49392  84.15869  86.82916
## 3   86.24130  85.03272  87.44987
## 4   86.98867  85.90390  88.07344
## 5   87.73605  86.77113  88.70096
## 6   88.48342  87.63273  89.33411
## 7   89.23079  88.48612  89.97547
## 8   89.97817  89.32727  90.62907
## 9   90.72554  90.15016  91.30093
```

```
## 10  91.47292  90.94686  91.99897
## 11  92.22029  91.70974  92.73084
## 12  92.96766  92.43582  93.49951
## 13  93.71504  93.12911  94.30097
## 14  94.46241  93.79755  95.12727
## 15  95.20979  94.44885  95.97072
## 16  95.95716  95.08867  96.82565
## 17  96.70453  95.72077  97.68830
## 18  97.45191  96.34756  98.55626
## 19  98.19928  96.97061  99.42795
## 20  98.94666  97.59095 100.30236
## 21  99.69403  98.20927 101.17879
## 22 100.44140  98.82605 102.05676
## 23 101.18878  99.44164 102.93591
## 24 101.93615 100.05630 103.81601
## 25 102.68353 100.67020 104.69685
```

Dapat dilihat bahwa ada dua kolom tambahan dari hasil prediksi anda, yaitu `lwr` dan `upr` yang merupakan batas bawah dan batas atas dari interval yang sudah anda definisikan. Anda dapat mengakses `lwr` dan `upr` dengan mengetikkan `ynew[,2]` dan `ynew[,3]`.

Sekarang mari kita coba eksekusi potongan kode berikut untuk membuat plot regresi linear dengan confidence interval:

```
xnew = data.frame(Hydrocarbon = seq(0.7,1.9,0.05))
ynew = predict(linregmod2,xnew,interval="confidence")
plot(xnew$Hydrocarbon,ynew[,1],type = "l",col = "red",lwd=2,xlab = "Hydrocarbon level (x)"
     ,ylab="Purity (y)",xlim=c(0.7, 1.9), ylim = c(88,98))
lines(xnew$Hydrocarbon,ynew[,2],col="black",lty = 2)
lines(xnew$Hydrocarbon,ynew[,3],col="black",lty = 2)
par(new=TRUE)
plot(X,Y,pch=19,col="blue",cex=1.5,xlab = "Hydrocarbon level (x)",ylab="Purity (y)"
     ,xlim=c(0.7, 1.9), ylim = c(88,98))
legend(0.7,98.2,legend=c("Regression line","Confidence interval"),col=c("red","black")
       ,lty = c(1,2), pt.cex=1, cex=0.75)
grid()
```

Anda dapat melihat bahwa garis regresi asli anda bisa saja lebih landai/curam, dan dengan 95% *confidence interval* anda bisa melihat kemungkinan-kemungkinan dimana garis akan berada.

Selain itu, anda pun juga ingin mengetahui *prediction interval* dari garis regresi utama yang telah anda buat. *Prediction interval* memberikan anda informasi mengenai ketidakpastian yang berasosiasi dengan prediksi model regresi anda. Ini berarti bahwa hasil prediksi anda bisa saja lebih rendah atau lebih tinggi. Informasi ini tentunya penting bagi anda karena anda tidak bisa sepenuhnya percaya dengan hasil prediksi. Anda membutuhkan informasi mengenai seberapa tidak pasti prediksi anda. *Prediction interval* akan berharga lebih tinggi dari *confidence interval*, karena ketidakpastian dari *prediction interval* datang dari dua sumber, yaitu ketidakpastian dari model regresi dan juga varians dari error (ε). Coba eksekusi potongan kode berikut untuk membuat prediction interval:

```
ynew = predict(linregmod2,xnew,interval="prediction")
plot(xnew$Hydrocarbon,ynew[,1],type = "l",col = "red",lwd=2,xlab = "Hydrocarbon level (x)"
     ,ylab="Purity (y)",xlim=c(0.7, 1.9), ylim = c(88,98))
lines(xnew$Hydrocarbon,ynew[,2],col="black",lty = 2)
lines(xnew$Hydrocarbon,ynew[,3],col="black",lty = 2)
par(new=TRUE)
plot(X,Y,pch=19,col="blue",cex=1.5,xlab = "Hydrocarbon level (x)",ylab="Purity (y)",
     xlim=c(0.7, 1.9), ylim = c(88,98))
```

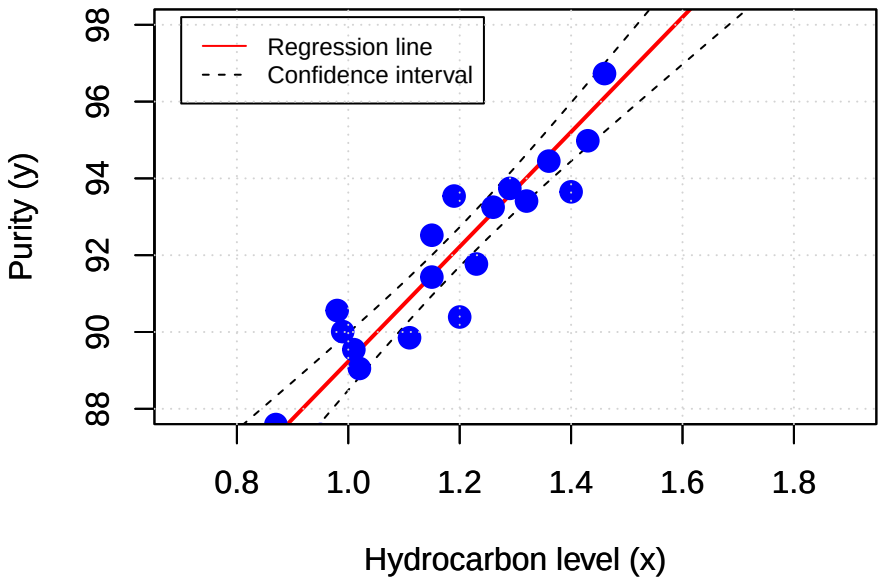

Figure 3: Visualisasi regresi linear beserta dengan confidence interval

```
legend(0.7,98,legend=c("Regression line","Prediction interval"),col=c("red","black")
       ,lty = c(1,2), pt.cex=1, cex=0.75)
grid()
```

Penutup

Demikian akhir dari tutorial regresi linear ini. Semoga dapat membantu anda memahami apa itu regresi linear dan bagaimana cara membuat regresi linear dalam R.

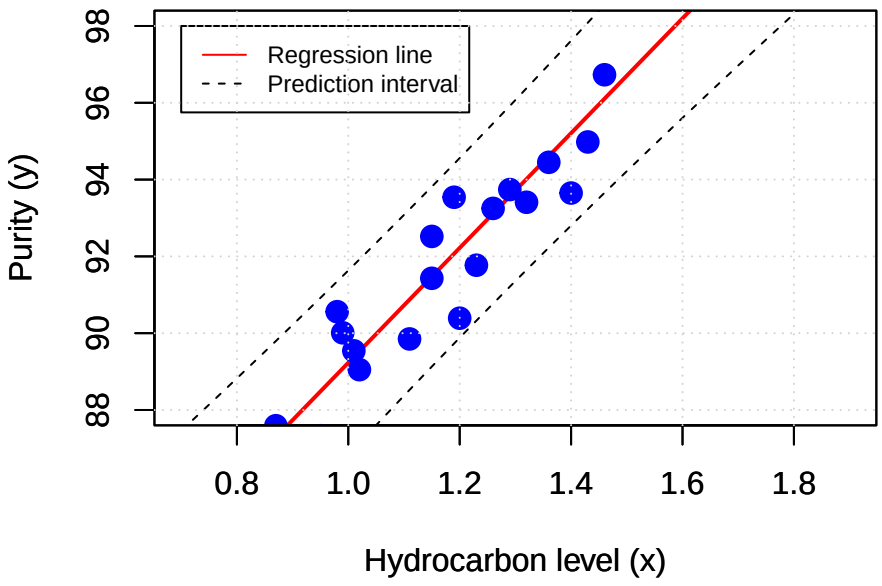

Figure 4: Visualisasi regresi linear beserta dengan prediction interval