Laporan Praktikum 2 AMP

Antonius Aditya Rizky Wijaya

G5402221003

2025-01-30

Linear Regression

Libraries

Fungsi `library()` dalam R digunakan untuk memuat kumpulan fungsi dan set data yang tidak termasuk dalam distribusi dasar R. Fungsi-fungsi dasar seperti regresi linear biasanya sudah tersedia dalam distribusi dasar, tetapi fungsi yang lebih kompleks memerlukan library tambahan.

```
library(MASS)
library(ISLR2)
```

```
## Warning: package 'ISLR2' was built under R version 4.3.3
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

Instal hanya satu kali. Namun, setiap kali memulai R, harus memanggil pustaka menggunakan fungsi `library()`.

Simple Linear Regression

`ISLR2` berisi dataset `Boston`, yang mencatat data `medv` (nilai median rumah) untuk 506 wilayah sensus di `Boston`. Analisis akan mencoba memprediksi `medv` menggunakan 12 prediktor, termasuk: > `rmvar`: rata-rata jumlah kamar per rumah, > `age`: proporsi unit yang dimiliki dan dibangun sebelum tahun 1940, > `lstat`: persentase rumah tangga dengan status sosial ekonomi rendah.

```
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
```

```
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

`lm()` digunakan untuk membuat model regresi linier sederhana di R, dengan format `lm(respons ~ prediktor, data)`. Misalnya, untuk memprediksi `medv` berdasarkan `lstat`.

```
#lm.fit <- lm(medv ~ lstat)
```

Error terjadi karena R tidak mengenali variabel. Dengan menggunakan `attach(Boston)`, variabel dalam dataset `Boston` dapat dikenali oleh R

```
lm.fit <- lm(medv ~ lstat, data = Boston)
attach(Boston)
lm.fit <- lm(medv ~ lstat)
```

`lm.fit` menampilkan informasi dasar model, sedangkan `summary(lm.fit)` memberikan informasi lebih rinci, termasuk p-value, standard error, R-squared, dan F-statistic untuk mengevaluasi model.

```
lm.fit

##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95

summary(lm.fit)

##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

`names()` digunakan untuk melihat informasi dalam `lm.fit`, tetapi lebih baik mengaksesnya dengan `coef()`.

```
names(lm.fit)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
coef(lm.fit)
```

```
## (Intercept)       lstat
##  34.5538409  -0.9500494
```

`confint()` digunakan untuk menghitung interval kepercayaan untuk estimasi koefisien regresi.

```
confint(lm.fit)
```

```
##                 2.5 %     97.5 %
## (Intercept) 33.448457 35.6592247
## lstat       -1.026148 -0.8739505
```

`predict()` berguna untuk menghitung interval kepercayaan dan prediksi saat ingin memperkirakan nilai respons (`medv`) pada suatu nilai prediktor tertentu (`lstat`).

```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))),
    interval = "confidence")
```

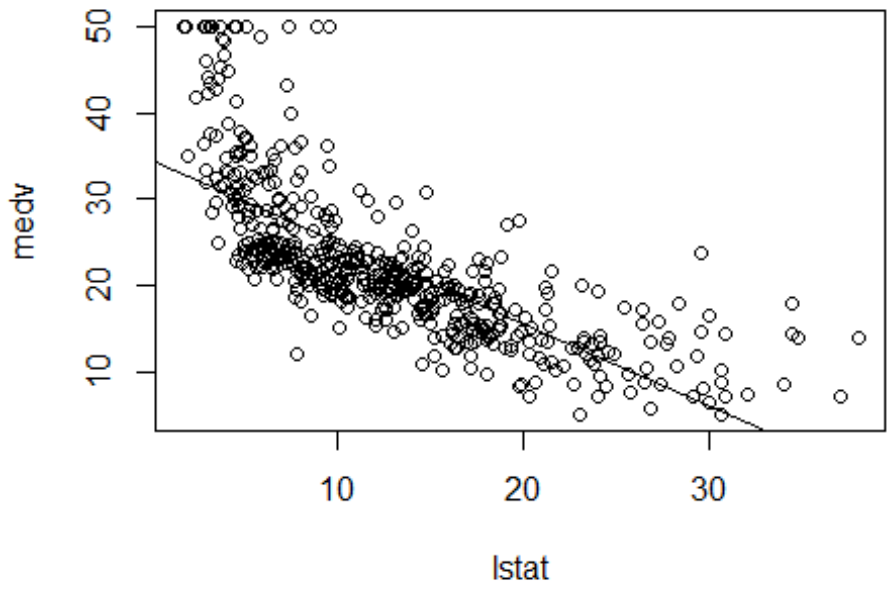
```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))),
    interval = "prediction")
```

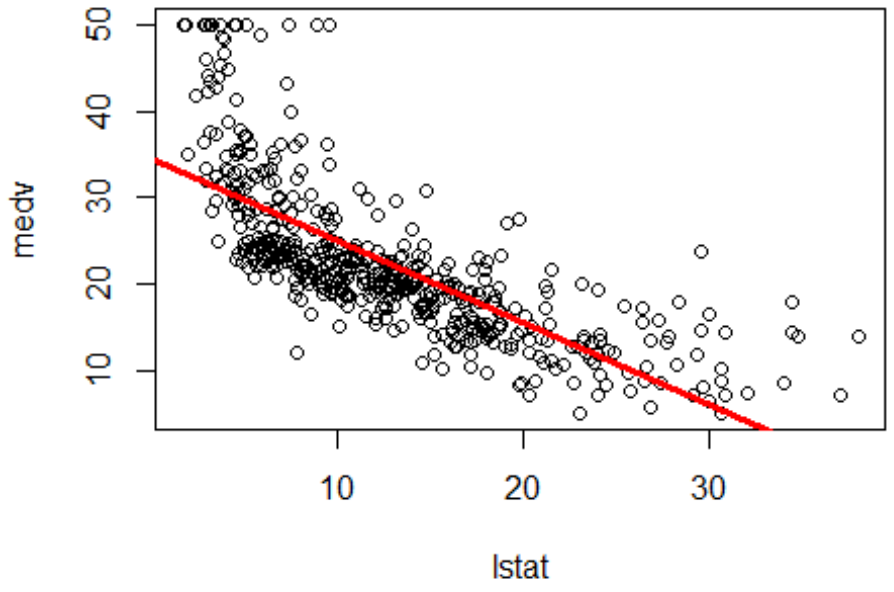
```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

Plot hubungan antara `medv` dan `lstat`, dengan garis regresi linear dengan fungsi `plot()` dan `abline()`

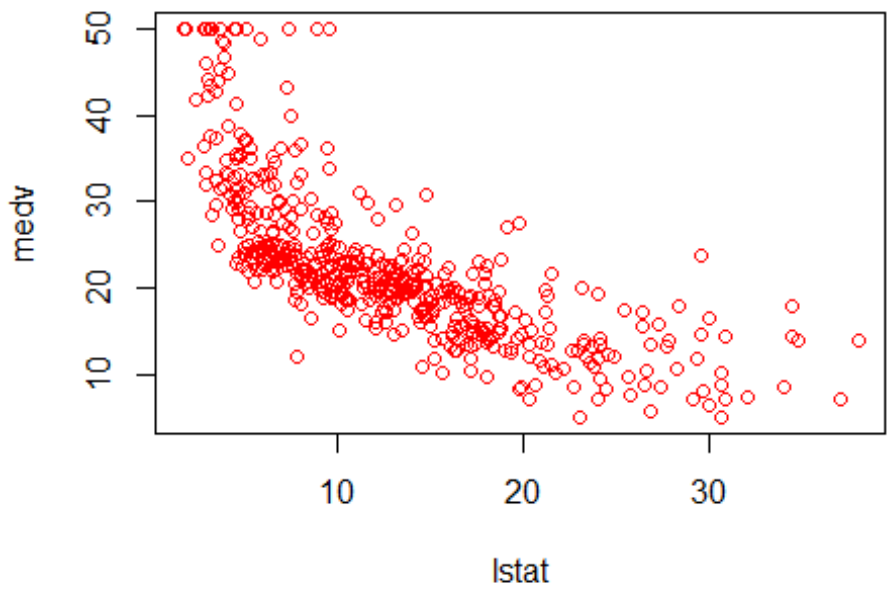
```
plot(lstat, medv)
abline(lm.fit)
```



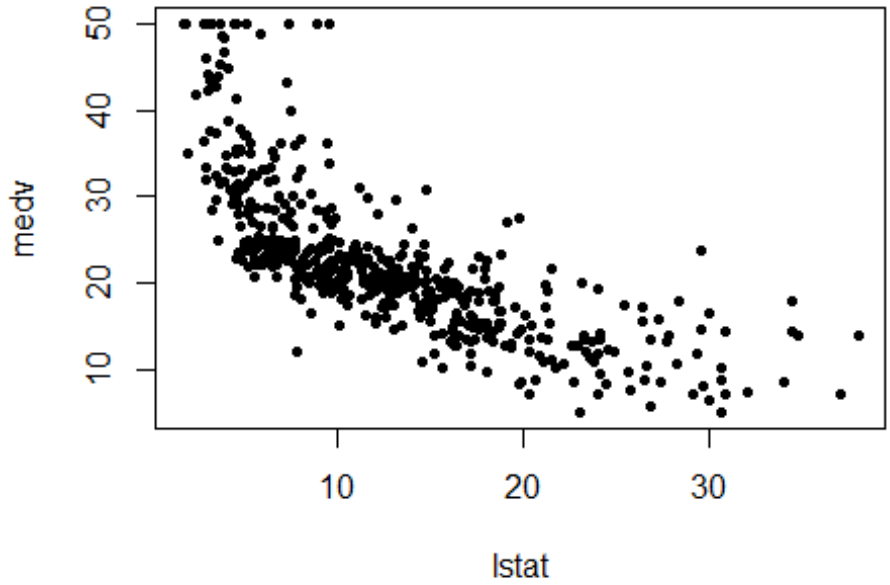

```
plot(lstat, medv)
abline(lm.fit, lwd = 3)
abline(lm.fit, lwd = 3, col = "red")
```



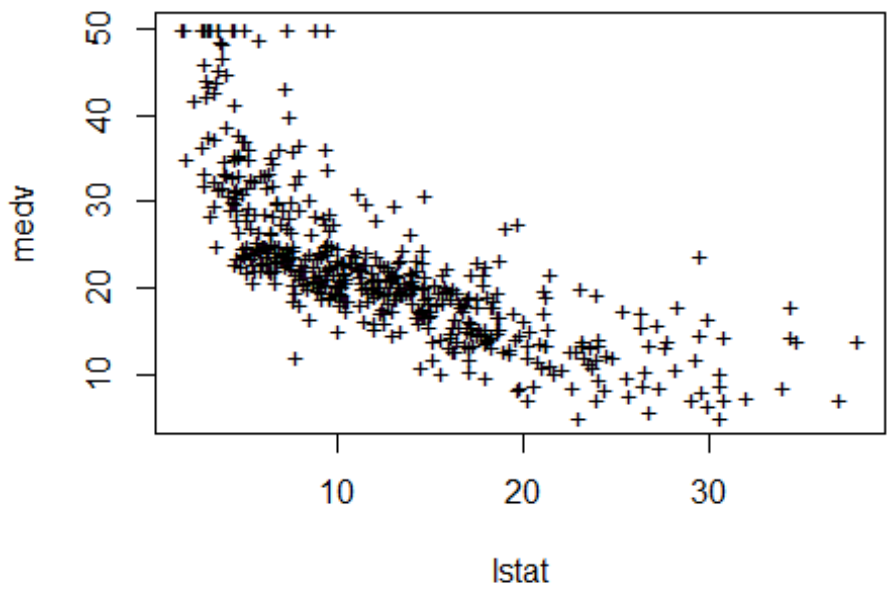
```r
plot(lstat, medv, col = "red")
```



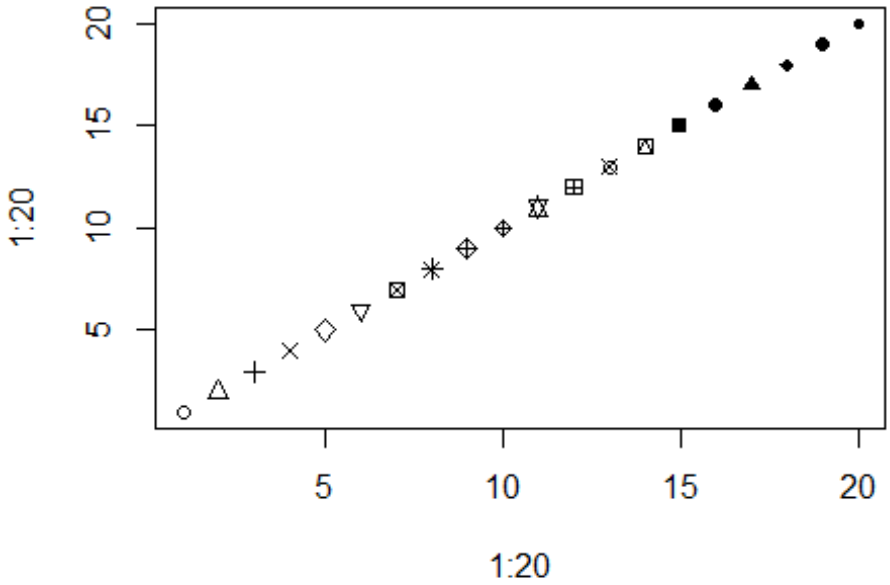

```r
plot(lstat, medv, pch = 20)
```



```
plot(lstat, medv, pch = "+")
```



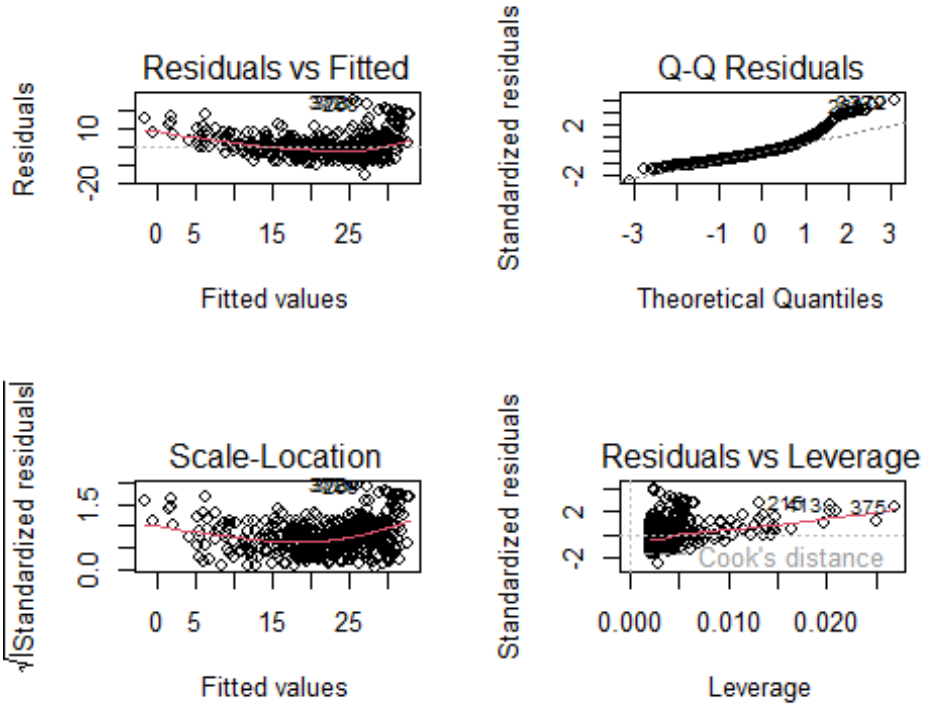

```
plot(1:20, 1:20, pch = 1:20)
```



Plot diagnostik yang dapat dihasilkan menggunakan fungsi `plot()` pada output dari fungsi `lm()`. Fungsi ini secara otomatis menghasilkan empat plot diagnostik satu per satu.

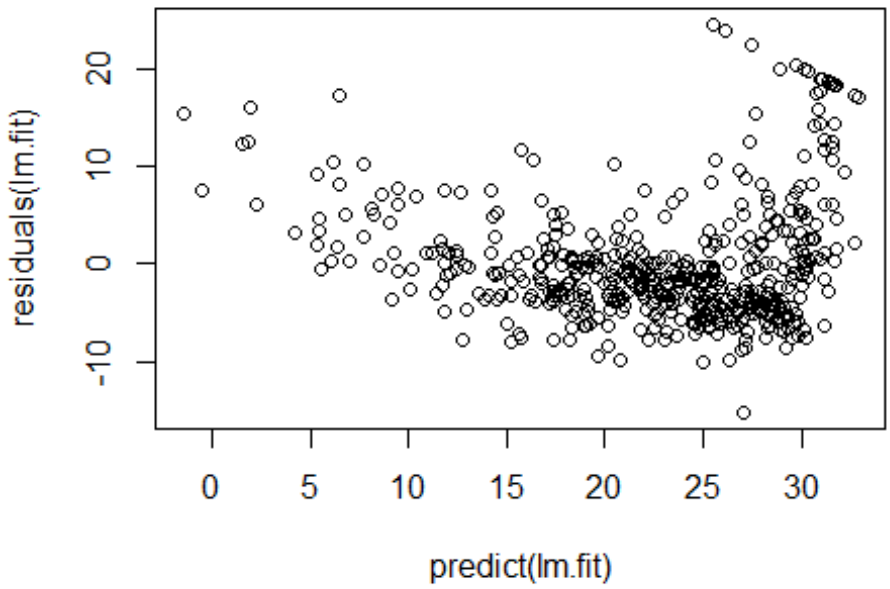
Agar lebih praktis, semua plot tersebut dapat ditampilkan secara bersamaan dengan menggunakan fungsi `par()` dan `mfrow()`. Sebagai contoh, perintah `par(mfrow = c(2, 2))` membagi area plot menjadi grid 2 × 2, sehingga keempat plot dapat dilihat dalam satu tampilan.

```
par(mfrow = c(2, 2))
plot(lm.fit)
```

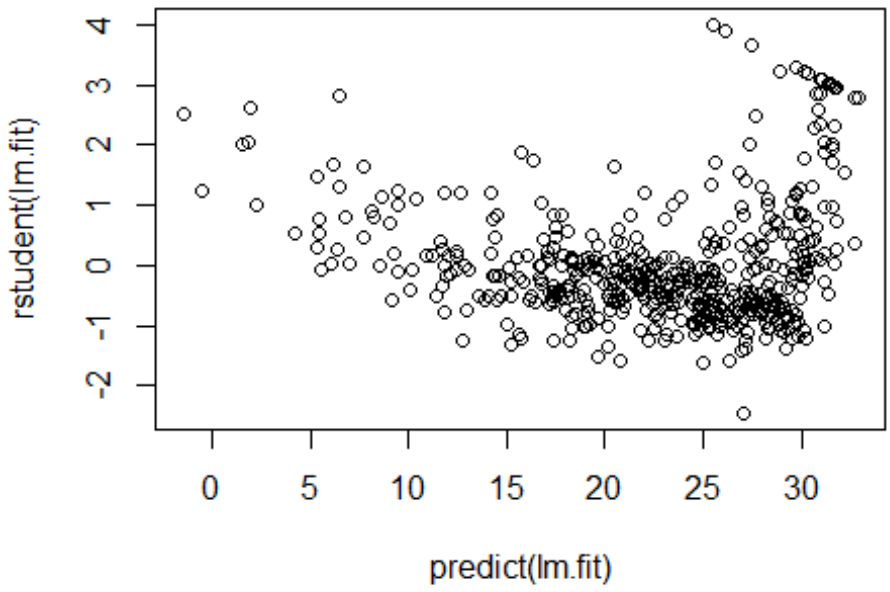


Residual regresi linear dapat dihitung dengan `residuals()`, sedangkan `rstudent()` menghasilkan residual yang distudentisasi untuk membuat plot terhadap nilai prediksi.

```
plot(predict(lm.fit), residuals(lm.fit))
```



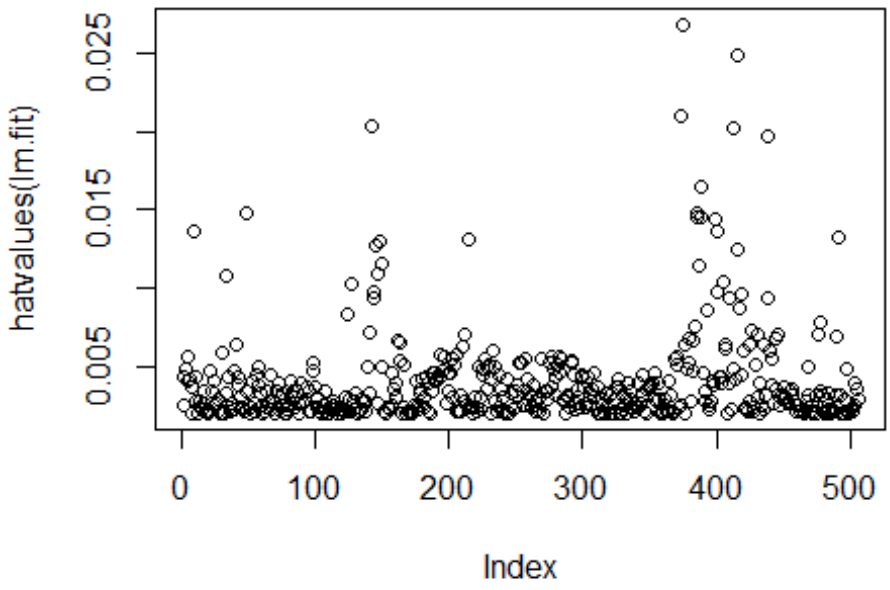

```r
plot(predict(lm.fit), rstudent(lm.fit))
```



Dari plot residu, terdapat beberapa bukti adanya non-linearitas dalam data. Statistik leverage dapat dihitung untuk sejumlah prediktor menggunakan fungsi `hatvalues()`.

`which.max()` berfungsi untuk menemukan indeks elemen dengan nilai terbesar pada sebuah vektor, misalnya untuk mencari observasi dengan leverage statistik tertinggi.

```
plot(hatvalues(lm.fit))
```




```
which.max(hatvalues(lm.fit))
```

```
## 375
## 375
```

Multiple Linear Regression

`lm()` digunakan untuk membuat model regresi linear berganda dengan metode kuadrat terkecil (least squares). Sintaks seperti `lm(y ~ x1 + x2 + x3)` dipakai untuk memodelkan hubungan antara variabel respon y dengan tiga prediktor, yaitu `x1`, `x2`, dan `x3`. Lalu `summary()` akan memberikan output koefisien regresi untuk semua prediktor.

```
lm.fit <- lm(medv ~ lstat + age, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

Agar tidak mengetik satu satu, shorthand dapat digunakan untuk menyertakan semua prediktor dalam regresi. Biasanya ditulis menggunakan format `.` (titik) sebagai perwakilan semua prediktor.

```
lm.fit <- lm(medv ~ ., data = Boston)
summary(lm.fit)

##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1304  -2.7673  -0.5814   1.9414  26.2526
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.617270   4.936039   8.431 3.79e-16 ***
## crim         -0.121389   0.033000  -3.678 0.000261 ***
## zn            0.046963   0.013879   3.384 0.000772 ***
## indus         0.013468   0.062145   0.217 0.828520
## chas          2.839993   0.870007   3.264 0.001173 **
## nox         -18.758022   3.851355  -4.870 1.50e-06 ***
## rm            3.658119   0.420246   8.705  < 2e-16 ***
## age           0.003611   0.013329   0.271 0.786595
## dis          -1.490754   0.201623  -7.394 6.17e-13 ***
## rad           0.289405   0.066908   4.325 1.84e-05 ***
## tax          -0.012682   0.003801  -3.337 0.000912 ***
## ptratio      -0.937533   0.132206  -7.091 4.63e-12 ***
## lstat        -0.552019   0.050659 -10.897  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.798 on 493 degrees of freedom
```

```
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7278
## F-statistic: 113.5 on 12 and 493 DF,  p-value: < 2.2e-16
```

Kita dapat mengakses komponen individu dari objek summary dalam R dengan menggunakan nama komponennya. `summary(lm.fit)$r.sq` untuk mendapatkan nilai R^2, dan `summary(lm.fit)$sigma` untuk mendapatkan Residual Standard Error (RSE).

`vif()` dari paket car digunakan untuk menghitung Variance Inflation Factors (VIF). Pada data ini, sebagian besar nilai VIF tergolong rendah hingga sedang.

```
#install.packages('car')
library(car)

## Warning: package 'car' was built under R version 4.3.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.3.3

vif(lm.fit)

##     crim       zn    indus     chas      nox       rm      age      dis
## 1.767486 2.298459 3.987181 1.071168 4.369093 1.912532 3.088232 3.954037
##      rad      tax  ptratio    lstat
## 7.445301 9.002158 1.797060 2.870777
```

Jika suatu variabel, seperti age, memiliki nilai p yang tinggi dan dianggap tidak signifikan, kita bisa mengevaluasi model tanpa age dengan sintaks : Alternatifnya, bisa gunakan fungsiupdate().

```
lm.fit1 <- lm(medv ~ . - age, data = Boston)
#lm.fit1 <- update(lm.fit, ~ . - age)
summary(lm.fit1)

##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1851  -2.7330  -0.6116   1.8555  26.3838
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.525128   4.919684   8.441 3.52e-16 ***
## crim         -0.121426   0.032969  -3.683 0.000256 ***
## zn            0.046512   0.013766   3.379 0.000785 ***
## indus         0.013451   0.062086   0.217 0.828577
## chas          2.852773   0.867912   3.287 0.001085 **
## nox         -18.485070   3.713714  -4.978 8.91e-07 ***
## rm            3.681070   0.411230   8.951  < 2e-16 ***
## dis          -1.506777   0.192570  -7.825 3.12e-14 ***
```

```
## rad           0.287940   0.066627   4.322 1.87e-05 ***
## tax          -0.012653   0.003796  -3.333 0.000923 ***
## ptratio      -0.934649   0.131653  -7.099 4.39e-12 ***
## lstat        -0.547409   0.047669 -11.483  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.794 on 494 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7284
## F-statistic: 124.1 on 11 and 494 DF,  p-value: < 2.2e-16
```

Interaction Terms

```
summary(lm(medv ~ lstat * age, data = Boston))

##
## Call:
## lm(formula = medv ~ lstat * age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.806  -4.045  -1.333   2.085  27.552
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.0885359  1.4698355  24.553  < 2e-16 ***
## lstat       -1.3921168  0.1674555  -8.313 8.78e-16 ***
## age         -0.0007209  0.0198792  -0.036   0.9711
## lstat:age    0.0041560  0.0018518   2.244   0.0252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.149 on 502 degrees of freedom
## Multiple R-squared:  0.5557, Adjusted R-squared:  0.5531
## F-statistic: 209.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Non-linear Transformations of the Predictors

`lm()` dapat digunakan untuk regresi dengan prediktor yang telah ditransformasikan secara non-linier, seperti menambahkan kuadrat dari prediktor (misalnya `lstat^2`), menggunakan fungsi `I()` untuk menghindari konflik dengan arti simbol ^ dalam formula.

```
lm.fit2 <- lm(medv ~ lstat + I(lstat^2))
summary(lm.fit2)

##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

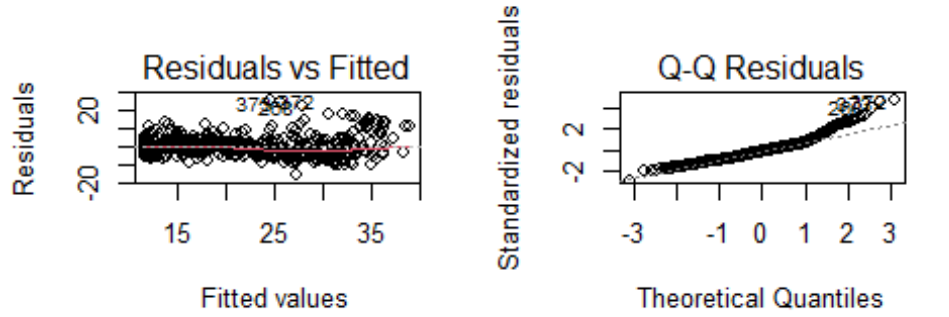
```
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

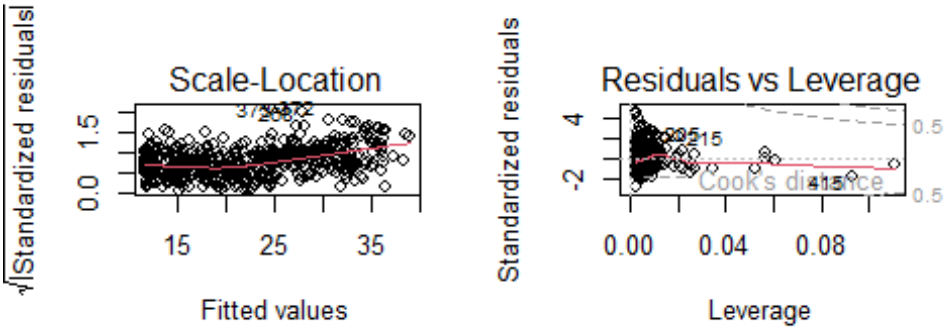
$p - value$ yang sangat kecil untuk istilah kuadratik menunjukkan bahwa istilah tersebut memperbaiki model. Fungsi `anova()` digunakan untuk membandingkan sejauh mana model kuadratik lebih unggul dari model linier.

```
lm.fit <- lm(medv ~ lstat)
anova(lm.fit, lm.fit2)

## Analysis of Variance Table
##
## Model 1: medv ~ lstat
## Model 2: medv ~ lstat + I(lstat^2)
##   Res.Df   RSS Df Sum of Sq     F    Pr(>F)
## 1    504 19472
## 2    503 15347  1    4125.1 135.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 2 yang menggunakan dua prediktor (`lstat` dan `lstat^2`) lebih baik daripada Model 1 yang hanya menggunakan `lstat`. Hal ini berdasarkan hasil uji ANOVA yang menunjukkan nilai F yang sangat tinggi dan $p - value$ yang hampir nol. Hal ini menunjukkan adanya hubungan non-linear antara medv dan lstat.

```
par(mfrow = c(2, 2))
plot(lm.fit2)
```





Ketika `lstat^2` ditambahkan dalam model, tidak ada pola yang terlihat pada residual, yang berarti model tersebut lebih baik. Sehingga menambahkan prediktor polinomial dapat meningkatkan model regresi.

Penggunaan fungsi `poly()` dalam `lm()` lebih efisien untuk membuat polinomial derajat tinggi,dari pada `I(X^3)`.

```
lm.fit5 <- lm(medv ~ poly(lstat, 5))
summary(lm.fit5)

##
## Call:
## lm(formula = medv ~ poly(lstat, 5))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5433  -3.1039  -0.7052   2.0844  27.1153
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       22.5328     0.2318  97.197  < 2e-16 ***
## poly(lstat, 5)1 -152.4595     5.2148 -29.236  < 2e-16 ***
## poly(lstat, 5)2   64.2272     5.2148  12.316  < 2e-16 ***
## poly(lstat, 5)3  -27.0511     5.2148  -5.187 3.10e-07 ***
## poly(lstat, 5)4   25.4517     5.2148   4.881 1.42e-06 ***
## poly(lstat, 5)5  -19.2524     5.2148  -3.692 0.000247 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.215 on 500 degrees of freedom
## Multiple R-squared:  0.6817, Adjusted R-squared:  0.6785
## F-statistic: 214.2 on 5 and 500 DF,  p-value: < 2.2e-16
```

Selain polinomial, kita juga dapat mencoba transformasi logaritma pada prediktor untuk eksplorasi lebih lanjut.

```
summary(lm(medv ~ log(rm), data = Boston))
```

```
##
## Call:
## lm(formula = medv ~ log(rm), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.487  -2.875  -0.104   2.837  39.816
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -76.488      5.028  -15.21   <2e-16 ***
## log(rm)       54.055      2.739   19.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.915 on 504 degrees of freedom
## Multiple R-squared:  0.4358, Adjusted R-squared:  0.4347
## F-statistic: 389.3 on 1 and 504 DF,  p-value: < 2.2e-16
```

Qualitative Predictors

Data `Carseats` yang ada dalam pustaka `ISLR2` digunakan untuk memprediksi `Sales` (penjualan kursi mobil anak) di 400 lokasi berdasarkan sejumlah prediktor.

```
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age
Education
## 1  9.50       138     73          11        276   120       Bad  42
17
## 2 11.22       111     48          16        260    83      Good  65
10
## 3 10.06       113     35          10        269    80    Medium  59
12
## 4  7.40       117    100           4        466    97    Medium  55
14
## 5  4.15       141     64           3        340   128       Bad  38
13
## 6 10.81       124    113          13        501    72       Bad  78
16
```

```
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

Data `Carseats` mencakup prediktor kualitatif seperti `shelveloc`, yang menunjukkan kualitas lokasi rak, yaitu ruang dalam toko tempat kursi mobil dipajang. Variabel prediktor `shelveloc` memiliki tiga nilai yang mungkin: *Bad*, *Medium*, dan *Good*.

Dalam model regresi ganda, variabel kualitatif seperti `shelveloc` secara otomatis diubah menjadi variabel dummy oleh R. Model regresi yang dipasang juga mencakup beberapa interaksi antar variabel.

```
lm.fit <- lm(Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9208 -0.7503  0.0177  0.6754  3.3413
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
## CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
## Income              0.0108940  0.0026044   4.183 3.57e-05 ***
## Advertising         0.0702462  0.0226091   3.107 0.002030 **
## Population          0.0001592  0.0003679   0.433 0.665330
## Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
## ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
## ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
## Age                -0.0579466  0.0159506  -3.633 0.000318 ***
## Education          -0.0208525  0.0196131  -1.063 0.288361
## UrbanYes            0.1401597  0.1124019   1.247 0.213171
## USYes              -0.1575571  0.1489234  -1.058 0.290729
## Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
## Price:Age           0.0001068  0.0001333   0.801 0.423812
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.011 on 386 degrees of freedom
## Multiple R-squared:  0.8761, Adjusted R-squared:  0.8719
## F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

`contrasts()` digunakan untuk melihat atau mengubah jenis pengkodean yang digunakan untuk variabel kategorikal, yang diubah menjadi variabel dummy (variabel biner) dalam model statistik. Variabel dummy ini digunakan dalam regresi dan analisis lainnya untuk mewakili kategori dalam bentuk angka.

```
attach(Carseats)
contrasts(ShelveLoc)
```

```
##        Good Medium
## Bad       0      0
## Good      1      0
## Medium    0      1
```

Penggunaan variabel dummy untuk menggambarkan lokasi rak produk dalam sebuah analisis regresi: > `ShelveLocGood`: Variabel dummy yang bernilai 1 jika lokasi rak produk baik, dan 0 jika tidak. > `ShelveLocMedium`: Variabel dummy yang bernilai 1 jika lokasi rak produk sedang, dan 0 jika tidak. > Lokasi rak buruk: Jika kedua variabel dummy (ShelveLocGood dan ShelveLocMedium) bernilai 0.

Writing Functions

Kita perlu menulis fungsi sendiri jika tidak ada fungsi yang sesuai. Jika fungsi ini belum didefinisikan, akan error saat di run.

```
#LoadLibraries
#LoadLibraries()
```

Kita definisikan fungsinya:

```
LoadLibraries <- function() {
 library(ISLR2)
 library(MASS)
 print("The libraries have been loaded.")
}

LoadLibraries
```

```
## function() {
##  library(ISLR2)
##  library(MASS)
##  print("The libraries have been loaded.")
## }
```

```
LoadLibraries()
```

```
## [1] "The libraries have been loaded."
```

Exercises

Nomor 8

This question involves the use of simple linear regression on the Auto data set.

a. Use the `lm()` function to perform a simple linear regression with `mpg` as the response and `horsepower` as the predictor. Use the `summary()` function to print the results. Comment on the output.

```
library(ISLR2)
mpg_hp<- lm(mpg ~ horsepower, data = Auto)
summary(mpg_hp)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictor and the response?

$p - value$ untuk variabel horsepower sangat kecil (jauh di bawah 0,05), sehingga ada bukti yang kuat untuk menyimpulkan bahwa terdapat hubungan antara variabel prediktor (horsepower) dan respons (mpg(miles per gallon)).

ii. How strong is the relationship between the predictor and the response?

```
summary(mpg_hp)$sigma
```

```
## [1] 4.905757
```

RSE berbeda dalam hal bahwa satuannya mengikuti satuan dari variabel y. Namun, kita bisa membaginya dengan $\bar{y}$ (nilai rata-rata dari y) untuk mendapatkan error dalam bentuk persentase :

```
summary(mpg_hp)$sigma/mean(Auto$mpg)
```

```
## [1] 0.2092371
```

Jadi persen error = 20.92%.

```
summary(mpg_hp)$r.squared

## [1] 0.6059483
```

R^2 dari model linier, bisa dianggap sebagai "persentase variasi pada respons yang dijelaskan oleh prediktor". R^2 adalah ukuran yang digunakan untuk menunjukkan seberapa baik model linier dapat menjelaskan atau memprediksi variabilitas data yang diamati. Dalam kasus ini, horsepower (prediktor) menjelaskan 60,59% varians dalam mpg (respons).

iii. Is the relationship between the predictor and the response positive or negative?

```
coefficients(mpg_hp)

## (Intercept)  horsepower
##  39.9358610  -0.1578447
```

Hubungannya negatif, artinya jika kendaraan memiliki horsepower lebih tinggi, umumnya nilai mpg-nya akan lebih rendah.

iv. What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

Jika nilai horsepower = 98, kita dapat memperoleh prediksi untuk nilai mpg (perkiraan), serta interval kepercayaan 95% dan interval prediksi 95% untuk mpg.

The confidence interval:

```
predict(mpg_hp, data.frame(horsepower = 98), interval = "confidence", level =
0.95)

##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

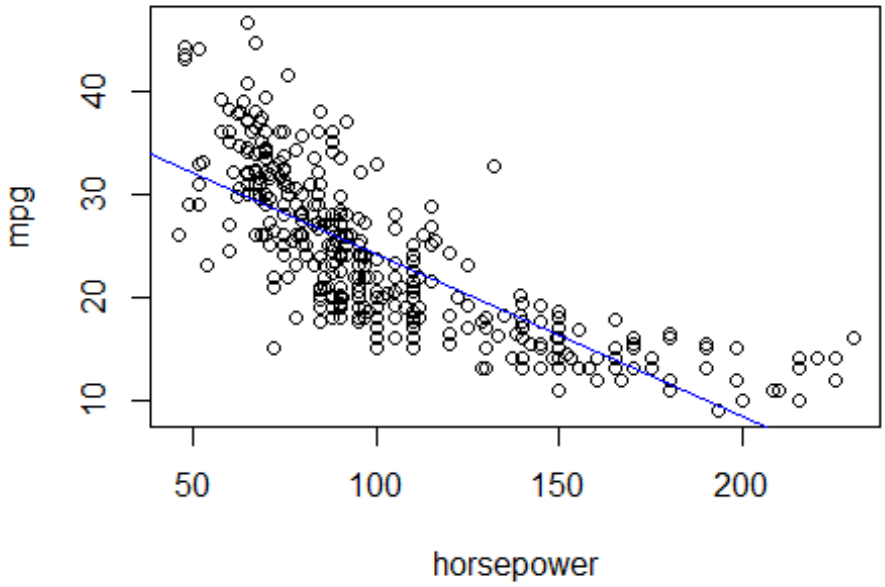
The prediction interval:

```
predict(mpg_hp, data.frame(horsepower = 98), interval = "prediction", level =
0.95)

##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

Interval prediksi lebih lebar daripada interval kepercayaan seperti yang kita harapkan. Hal ini karena mempertimbangkan variasi tambahan dalam pengamatan individu.

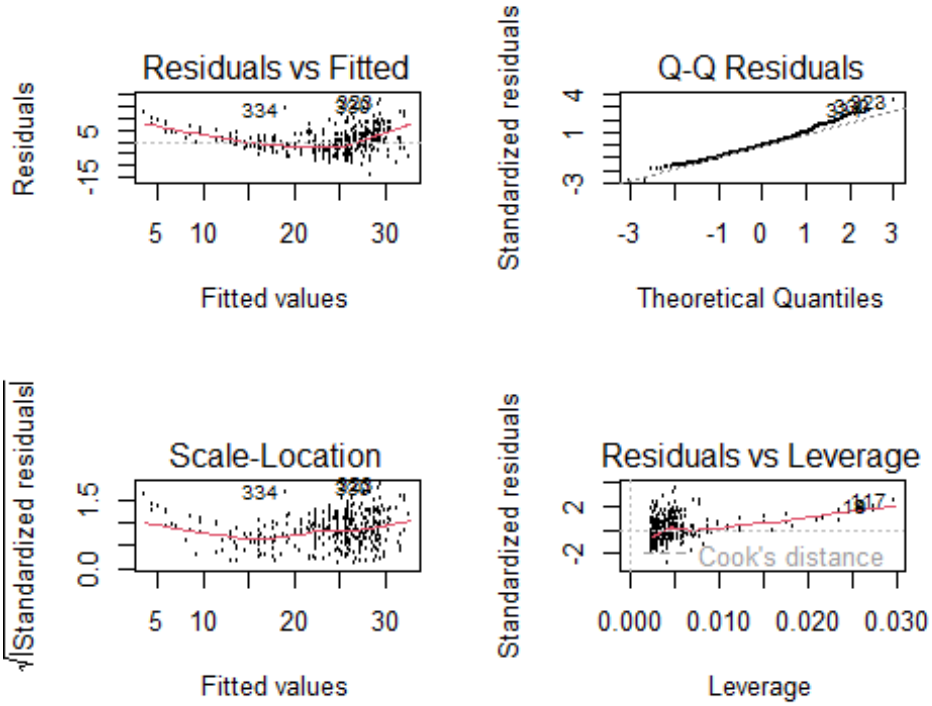
b. Plot the response and the predictor. Use the `abline()` function to display the least squares regression line.

```
plot(Auto$horsepower, Auto$mpg, xlab = "horsepower", ylab = "mpg")
abline(mpg_hp, col = "blue")
```



c. Use the `plot()` function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

```
par(mfrow = c(2,2))
plot(mpg_hp, cex = 0.2)
```



Pada grafik yang menggambarkan hubungan antara residual (ei = yi − yi^) dan nilai yang diprediksi (yi^), terlihat ada pola yang kuat pada residuals, yang mengindikasikan adanya non-linearitas.

Selain itu, ada varians yang tidak konstan pada error (heteroskedastisitas), tetapi hal ini bisa diperbaiki hingga tingkat tertentu dengan mencoba model kuadrat. Jika perbaikan ini tidak berhasil, transformasi seperti log(y) atau sqrt(y) dapat dicoba.

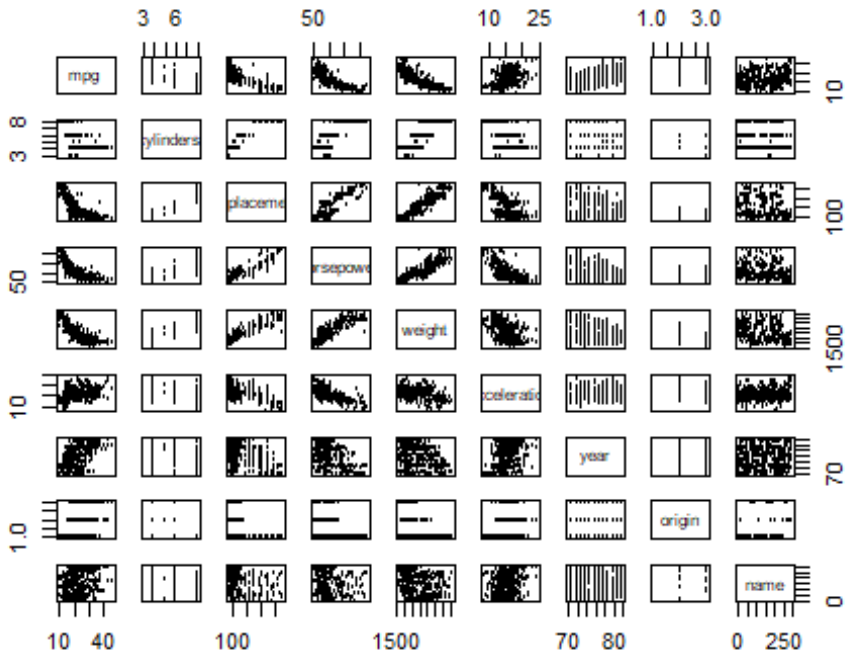
Ada juga beberapa observasi dengan residual standar yang besar dan leverage tinggi (sehingga menghasilkan Cook's Distance yang tinggi), yang mungkin perlu diperiksa lebih lanjut.

Nomor 9

This question involves the use of multiple linear regression on the `Auto` data set.

a. Produce a scatterplot matrix which includes all of the variables in the data set.

```
pairs(Auto, cex = 0.2)
```



b. Compute the matrix of correlations between the variables using the function `cor()`. You will need to exclude the name variable, `name` which is qualitative.

```
cor(subset(Auto, select = -name))
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
```

```
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

c. Use the `lm()` function to perform a multiple linear regression with `mpg` as the response and all other variables except name as the predictors. Use the `summary()` function to print the results. Comment on the output.

```
mpg_lm <- lm(mpg ~ . - name, data = Auto)
summary(mpg_lm)

##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response?

Ya, ada hubungan antara beberapa prediktor dan respons, yaitu “displacement” (positif), “weight” (negatif), “year” (positif), dan “origin” (positif).

ii. Which predictors appear to have a statistically significant relationship to the response?

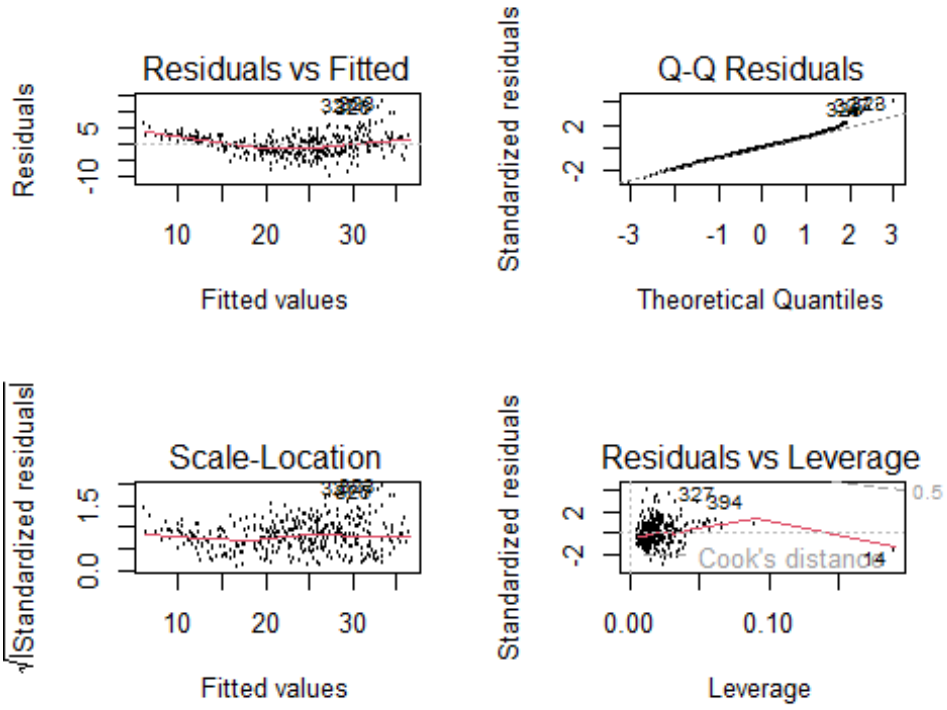
Berdasarkan $p - value$ untuk koefisien dalam output model, dan dengan p = 0,05 sebagai ambang batas untuk signifikansi, semua variabel kecuali jumlah silinder, tenaga kuda (horsepower), dan akselerasi memiliki hubungan yang signifikan secara statistik dengan respons (variabel dependen).

iii. What does the coefficient for the `year` variable suggest?

Koefisien untuk variabel year (yang bernilai positif sekitar 0.75) menunjukkan bahwa rata-rata konsumsi bahan bakar per galon (mpg) meningkat sekitar 0.75 setiap tahunnya. Artinya, setiap tahun, mpg cenderung meningkat sebesar 0.75 unit.

d. Use the `plot()` function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```
par(mfrow = c(2, 2))
plot(mpg_lm, cex = 0.2)
```



Satu titik memiliki leverage yang tinggi, residualnya juga menunjukkan tren dengan nilai yang disesuaikan. Ini berarti ada titik data yang sangat memengaruhi model, atau dengan kata lain ada outlier yang memiliki nilai besar meski hanya sedikit.

e. Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
summary(lm(formula = mpg ~ . * ., data = Auto[, -9]))

##
## Call:
## lm(formula = mpg ~ . * ., data = Auto[, -9])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.6303 -1.4481  0.0596  1.2739 11.1386
##
```

```
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  3.548e+01  5.314e+01   0.668  0.50475
## cylinders                    6.989e+00  8.248e+00   0.847  0.39738
## displacement                -4.785e-01  1.894e-01  -2.527  0.01192 *
## horsepower                   5.034e-01  3.470e-01   1.451  0.14769
## weight                       4.133e-03  1.759e-02   0.235  0.81442
## acceleration                -5.859e+00  2.174e+00  -2.696  0.00735 **
## year                         6.974e-01  6.097e-01   1.144  0.25340
## origin                      -2.090e+01  7.097e+00  -2.944  0.00345 **
## cylinders:displacement      -3.383e-03  6.455e-03  -0.524  0.60051
## cylinders:horsepower         1.161e-02  2.420e-02   0.480  0.63157
## cylinders:weight             3.575e-04  8.955e-04   0.399  0.69000
## cylinders:acceleration       2.779e-01  1.664e-01   1.670  0.09584 .
## cylinders:year              -1.741e-01  9.714e-02  -1.793  0.07389 .
## cylinders:origin             4.022e-01  4.926e-01   0.816  0.41482
## displacement:horsepower     -8.491e-05  2.885e-04  -0.294  0.76867
## displacement:weight          2.472e-05  1.470e-05   1.682  0.09342 .
## displacement:acceleration   -3.479e-03  3.342e-03  -1.041  0.29853
## displacement:year            5.934e-03  2.391e-03   2.482  0.01352 *
## displacement:origin          2.398e-02  1.947e-02   1.232  0.21875
## horsepower:weight           -1.968e-05  2.924e-05  -0.673  0.50124
## horsepower:acceleration     -7.213e-03  3.719e-03  -1.939  0.05325 .
## horsepower:year             -5.838e-03  3.938e-03  -1.482  0.13916
## horsepower:origin            2.233e-03  2.930e-02   0.076  0.93931
## weight:acceleration          2.346e-04  2.289e-04   1.025  0.30596
## weight:year                 -2.245e-04  2.127e-04  -1.056  0.29182
## weight:origin               -5.789e-04  1.591e-03  -0.364  0.71623
## acceleration:year            5.562e-02  2.558e-02   2.174  0.03033 *
## acceleration:origin          4.583e-01  1.567e-01   2.926  0.00365 **
## year:origin                  1.393e-01  7.399e-02   1.882  0.06062 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.695 on 363 degrees of freedom
## Multiple R-squared:  0.8893, Adjusted R-squared:  0.8808
## F-statistic: 104.2 on 28 and 363 DF,  p-value: < 2.2e-16
```

Kita dapat melihat terms yang signifikan secara statistik (pada level 0,05) ditandai dengan setidaknya satu tanda (*). Selain itu, R^2 nya (0.8893) menunjukkan bahwa model signifikan.

f. Try a few different transformations of the variables, such as $log(X)$, $\sqrt{X}$, X^2. Comment on your findings.

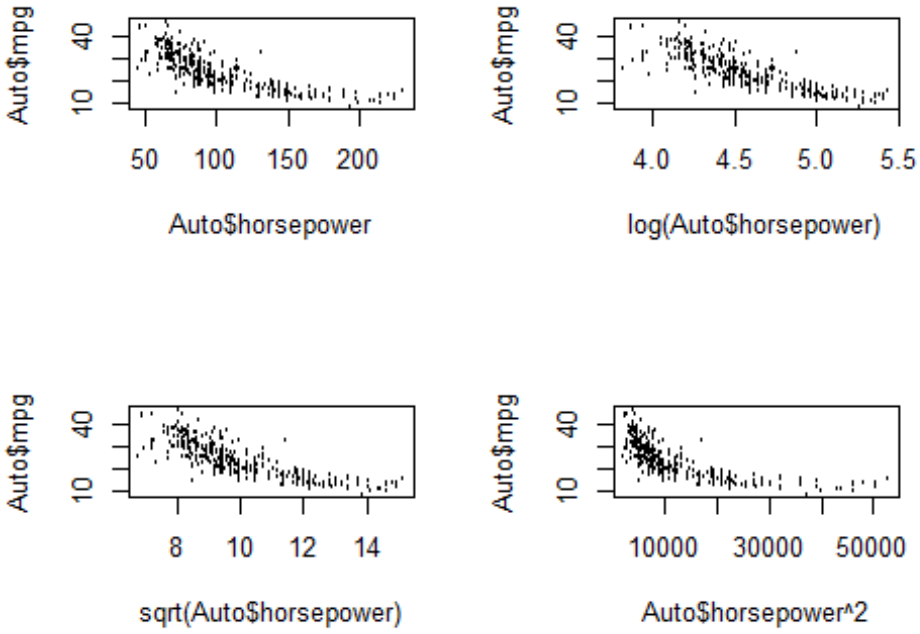
```
model_log <- lm(mpg ~ log(horsepower) + log(weight) + log(cylinders), data =
Auto)
model_sqrt <- lm(mpg ~ sqrt(horsepower) + sqrt(weight) + sqrt(cylinders),
data = Auto)
model_squared <- lm(mpg ~ I(horsepower^2) + I(weight^2) + I(cylinders^2),
```

```
data = Auto)
summary(model_log)

##
## Call:
## lm(formula = mpg ~ log(horsepower) + log(weight) + log(cylinders),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1686  -2.4457  -0.3318   2.0495  15.3999
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      171.907     10.552  16.292  < 2e-16 ***
## log(horsepower)   -7.352      1.246  -5.900 7.94e-09 ***
## log(weight)      -14.087      1.828  -7.708 1.08e-13 ***
## log(cylinders)    -1.578      1.468  -1.075    0.283
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.992 on 388 degrees of freedom
## Multiple R-squared:  0.7403, Adjusted R-squared:  0.7383
## F-statistic: 368.8 on 3 and 388 DF,  p-value: < 2.2e-16

summary(model_sqrt)

##
## Call:
## lm(formula = mpg ~ sqrt(horsepower) + sqrt(weight) + sqrt(cylinders),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2868  -2.6079  -0.3064   2.1647  15.8293
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      68.63469    1.49919  45.781  < 2e-16 ***
## sqrt(horsepower) -1.20859    0.24679  -4.897 1.43e-06 ***
## sqrt(weight)     -0.55970    0.06922  -8.086 7.96e-15 ***
## sqrt(cylinders)  -1.20517    1.34313  -0.897     0.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.097 on 388 degrees of freedom
## Multiple R-squared:  0.7266, Adjusted R-squared:  0.7245
## F-statistic: 343.7 on 3 and 388 DF,  p-value: < 2.2e-16

summary(model_squared)
```

```
##
## Call:
## lm(formula = mpg ~ I(horsepower^2) + I(weight^2) + I(cylinders^2),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.6639  -3.2784  -0.4586   2.6037  17.2283
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.445e+01  4.733e-01  72.783  < 2e-16 ***
## I(horsepower^2) -6.316e-05  4.427e-05  -1.427  0.15450
## I(weight^2)     -7.804e-07  1.023e-07  -7.631 1.82e-13 ***
## I(cylinders^2)  -8.332e-02  2.653e-02  -3.140  0.00182 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.539 on 388 degrees of freedom
## Multiple R-squared:  0.6644, Adjusted R-squared:  0.6618
## F-statistic: 256.1 on 3 and 388 DF,  p-value: < 2.2e-16
```

Dilihat dari nilai R^2 nya, transformasi log dari `horsepower` mampu memberikan hubungan yang lebih linear dengan mpg, karena memiliki R^2 yang paling besar (0.7403)

```
par(mfrow = c(2, 2))
plot(Auto$horsepower, Auto$mpg, cex = 0.2)
plot(log(Auto$horsepower), Auto$mpg, cex = 0.2)
plot(sqrt(Auto$horsepower), Auto$mpg, cex = 0.2)
plot(Auto$horsepower^2, Auto$mpg, cex = 0.2)
```



By the plot juga terlihat transformasi log dari `horsepower` memberikan hubungan yang lebih linear dengan mpg.