Laporan Praktikum 4 AMP

Antonius Aditya Rizky Wijaya

G5402221003

2025-02-13

Classification Methods

```
library(ISLR2)

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

attach(Smarket)
```

Logistic Regression

```
glm.fits <- glm(
    Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
    data = Smarket, family = binomial
  )
summary(glm.fits)

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Smarket)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1741.6
##
## Number of Fisher Scoring iterations: 3
```

Keterangan : Model menggunakan semua lag dan volume sebagai prediktor. Dari output `summary(glm.fits)`, kita bisa melihat apakah variabel-variabel ini signifikan atau tidak berdasarkan p-value.

```
coef(glm.fits)

##  (Intercept)         Lag1         Lag2         Lag3         Lag4 
Lag5 
## -0.126000257 -0.073073746 -0.042301344  0.011085108  0.009358938 
0.010313068 
##       Volume 
##  0.135440659

summary(glm.fits)$coef

##                 Estimate Std. Error    z value  Pr(>|z|)
## (Intercept) -0.126000257 0.24073574 -0.5233966 0.6006983
## Lag1        -0.073073746 0.05016739 -1.4565986 0.1452272
## Lag2        -0.042301344 0.05008605 -0.8445733 0.3983491
## Lag3         0.011085108 0.04993854  0.2219750 0.8243333
## Lag4         0.009358938 0.04997413  0.1872757 0.8514445
## Lag5         0.010313068 0.04951146  0.2082966 0.8349974
## Volume       0.135440659 0.15835970  0.8552723 0.3924004

summary(glm.fits)$coef[, 4]

## (Intercept)        Lag1        Lag2        Lag3        Lag4        Lag5 
##   0.6006983   0.1452272   0.3983491   0.8243333   0.8514445   0.8349974 
##      Volume 
##   0.3924004
```

Keterangan :

Kita bisa melihat prediktor mana yang signifikan berdasarkan p-value (biasanya < 0.05 dianggap signifikan).

```
glm.probs <- predict(glm.fits, type = "response")
glm.probs[1:10]

##         1         2         3         4         5         6         7 
8 
## 0.5070841 0.4814679 0.4811388 0.5152224 0.5107812 0.5069565 0.4926509 
0.5092292 
##         9        10 
## 0.5176135 0.4888378

contrasts(Direction)

##      Up
## Down  0
## Up    1
```

Keterangan :

Model menghasilkan probabilitas dari `Up`, dan kita bisa melihat bagaimana kategori dikodekan dalam regresi logistik.

```
glm.pred <- rep("Down", 1250)
glm.pred[glm.probs > .5] = "Up"
```

Keterangan :

Model sekarang menghasilkan klasifikasi biner (`Up` atau `Down`) berdasarkan probabilitas.

```
table(glm.pred, Direction)

##          Direction
## glm.pred Down  Up
##     Down  145 141
##     Up    457 507

(507 + 145) / 1250

## [1] 0.5216

mean(glm.pred == Direction)

## [1] 0.5216
```

Keterangan :

Model memiliki akurasi tertentu, tetapi kita belum tahu apakah ini lebih baik dari tebakan acak.

```
train <- (Year < 2005)
Smarket.2005 <- Smarket[!train, ]
dim(Smarket.2005)

## [1] 252   9

Direction.2005 <- Direction[!train]
```

Keterangan :

Dataset sekarang dipisah menjadi train (sebelum 2005) dan test (2005 ke atas) untuk mengevaluasi model dengan data baru.

```
glm.fits <- glm(
    Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
    data = Smarket, family = binomial, subset = train
  )
glm.probs <- predict(glm.fits, Smarket.2005,
    type = "response")
```

Keterangan :

Model sekarang diuji pada data baru, bukan pada data latih.

```
glm.pred <- rep("Down", 252)
glm.pred[glm.probs > .5] <- "Up"
table(glm.pred, Direction.2005)

##         Direction.2005
## glm.pred Down Up
##     Down   77 97
##     Up     34 44

mean(glm.pred == Direction.2005)

## [1] 0.4801587

mean(glm.pred != Direction.2005)

## [1] 0.5198413
```

Keterangan :

Akurasi model bisa dibandingkan dengan baseline model (tebakan acak).

```
glm.fits <- glm(Direction ~ Lag1 + Lag2, data = Smarket,
    family = binomial, subset = train)
glm.probs <- predict(glm.fits, Smarket.2005,
    type = "response")
glm.pred <- rep("Down", 252)
glm.pred[glm.probs > .5] <- "Up"
table(glm.pred, Direction.2005)

##         Direction.2005
## glm.pred Down  Up
##     Down   35  35
##     Up     76 106

mean(glm.pred == Direction.2005)

## [1] 0.5595238

106 / (106 + 76)

## [1] 0.5824176
```

Keterangan :

Menggunakan lebih sedikit prediktor mungkin meningkatkan atau menurunkan performa model.

```
predict(glm.fits,
    newdata =
      data.frame(Lag1 = c(1.2, 1.5),  Lag2 = c(1.1, -0.8)),
```

```r
    type = "response"
  )
```

```
##         1         2
## 0.4791462 0.4960939
```

Keterangan :

Model bisa digunakan untuk memprediksi tren pasar berdasarkan nilai `Lag1` dan `Lag2` baru.

Poisson Regression

```r
attach(Bikeshare)
dim(Bikeshare)
```

```
## [1] 8645   15
```

```r
names(Bikeshare)
```

```
##  [1] "season"     "mnth"       "day"        "hr"         "holiday"
##  [6] "weekday"    "workingday" "weathersit" "temp"       "atemp"
## [11] "hum"        "windspeed"  "casual"     "registered" "bikers"
```

Keterangan :

Mengecek struktur dataset `Bikeshare`, termasuk jumlah variabel dan nama kolomnya.

```r
mod.lm <- lm(
    bikers ~ mnth + hr + workingday + temp + weathersit,
    data = Bikeshare
  )
summary(mod.lm)
```

```
##
## Call:
## lm(formula = bikers ~ mnth + hr + workingday + temp + weathersit,
##     data = Bikeshare)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -299.00  -45.70   -6.23   41.08  425.29
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -68.632      5.307 -12.932  < 2e-16 ***
## mnthFeb                       6.845      4.287   1.597 0.110398
## mnthMarch                    16.551      4.301   3.848 0.000120 ***
## mnthApril                    41.425      4.972   8.331  < 2e-16 ***
## mnthMay                      72.557      5.641  12.862  < 2e-16 ***
## mnthJune                     67.819      6.544  10.364  < 2e-16 ***
## mnthJuly                     45.324      7.081   6.401 1.63e-10 ***
## mnthAug                      53.243      6.640   8.019 1.21e-15 ***
```

```
## mnthSept                     66.678      5.925  11.254  < 2e-16 ***
## mnthOct                      75.834      4.950  15.319  < 2e-16 ***
## mnthNov                      60.310      4.610  13.083  < 2e-16 ***
## mnthDec                      46.458      4.271  10.878  < 2e-16 ***
## hr1                         -14.579      5.699  -2.558 0.010536 *
## hr2                         -21.579      5.733  -3.764 0.000168 ***
## hr3                         -31.141      5.778  -5.389 7.26e-08 ***
## hr4                         -36.908      5.802  -6.361 2.11e-10 ***
## hr5                         -24.135      5.737  -4.207 2.61e-05 ***
## hr6                          20.600      5.704   3.612 0.000306 ***
## hr7                         120.093      5.693  21.095  < 2e-16 ***
## hr8                         223.662      5.690  39.310  < 2e-16 ***
## hr9                         120.582      5.693  21.182  < 2e-16 ***
## hr10                         83.801      5.705  14.689  < 2e-16 ***
## hr11                        105.423      5.722  18.424  < 2e-16 ***
## hr12                        137.284      5.740  23.916  < 2e-16 ***
## hr13                        136.036      5.760  23.617  < 2e-16 ***
## hr14                        126.636      5.776  21.923  < 2e-16 ***
## hr15                        132.087      5.780  22.852  < 2e-16 ***
## hr16                        178.521      5.772  30.927  < 2e-16 ***
## hr17                        296.267      5.749  51.537  < 2e-16 ***
## hr18                        269.441      5.736  46.976  < 2e-16 ***
## hr19                        186.256      5.714  32.596  < 2e-16 ***
## hr20                        125.549      5.704  22.012  < 2e-16 ***
## hr21                         87.554      5.693  15.378  < 2e-16 ***
## hr22                         59.123      5.689  10.392  < 2e-16 ***
## hr23                         26.838      5.688   4.719 2.41e-06 ***
## workingday                    1.270      1.784   0.711 0.476810
## temp                        157.209     10.261  15.321  < 2e-16 ***
## weathersitcloudy/misty      -12.890      1.964  -6.562 5.60e-11 ***
## weathersitlight rain/snow   -66.494      2.965 -22.425  < 2e-16 ***
## weathersitheavy rain/snow -109.745     76.667  -1.431 0.152341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.5 on 8605 degrees of freedom
## Multiple R-squared:  0.6745, Adjusted R-squared:  0.6731
## F-statistic: 457.3 on 39 and 8605 DF,  p-value: < 2.2e-16
```

Keterangan :

Menentukan pengaruh variabel prediktor terhadap jumlah `bikers` menggunakan regresi linear.

```
contrasts(Bikeshare$hr) = contr.sum(24)
contrasts(Bikeshare$mnth) = contr.sum(12)
mod.lm2 <- lm(
    bikers ~ mnth + hr + workingday + temp + weathersit,
    data = Bikeshare
```

```
  )
summary(mod.lm2)

##
## Call:
## lm(formula = bikers ~ mnth + hr + workingday + temp + weathersit,
##     data = Bikeshare)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -299.00  -45.70   -6.23   41.08  425.29
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                73.5974     5.1322  14.340  < 2e-16 ***
## mnth1                     -46.0871     4.0855 -11.281  < 2e-16 ***
## mnth2                     -39.2419     3.5391 -11.088  < 2e-16 ***
## mnth3                     -29.5357     3.1552  -9.361  < 2e-16 ***
## mnth4                      -4.6622     2.7406  -1.701  0.08895 .
## mnth5                      26.4700     2.8508   9.285  < 2e-16 ***
## mnth6                      21.7317     3.4651   6.272 3.75e-10 ***
## mnth7                      -0.7626     3.9084  -0.195  0.84530
## mnth8                       7.1560     3.5347   2.024  0.04295 *
## mnth9                      20.5912     3.0456   6.761 1.46e-11 ***
## mnth10                     29.7472     2.6995  11.019  < 2e-16 ***
## mnth11                     14.2229     2.8604   4.972 6.74e-07 ***
## hr1                       -96.1420     3.9554 -24.307  < 2e-16 ***
## hr2                      -110.7213     3.9662 -27.916  < 2e-16 ***
## hr3                      -117.7212     4.0165 -29.310  < 2e-16 ***
## hr4                      -127.2828     4.0808 -31.191  < 2e-16 ***
## hr5                      -133.0495     4.1168 -32.319  < 2e-16 ***
## hr6                      -120.2775     4.0370 -29.794  < 2e-16 ***
## hr7                       -75.5424     3.9916 -18.925  < 2e-16 ***
## hr8                        23.9511     3.9686   6.035 1.65e-09 ***
## hr9                       127.5199     3.9500  32.284  < 2e-16 ***
## hr10                       24.4399     3.9360   6.209 5.57e-10 ***
## hr11                      -12.3407     3.9361  -3.135  0.00172 **
## hr12                        9.2814     3.9447   2.353  0.01865 *
## hr13                       41.1417     3.9571  10.397  < 2e-16 ***
## hr14                       39.8939     3.9750  10.036  < 2e-16 ***
## hr15                       30.4940     3.9910   7.641 2.39e-14 ***
## hr16                       35.9445     3.9949   8.998  < 2e-16 ***
## hr17                       82.3786     3.9883  20.655  < 2e-16 ***
## hr18                      200.1249     3.9638  50.488  < 2e-16 ***
## hr19                      173.2989     3.9561  43.806  < 2e-16 ***
## hr20                       90.1138     3.9400  22.872  < 2e-16 ***
## hr21                       29.4071     3.9362   7.471 8.74e-14 ***
## hr22                       -8.5883     3.9332  -2.184  0.02902 *
## hr23                      -37.0194     3.9344  -9.409  < 2e-16 ***
## workingday                  1.2696     1.7845   0.711  0.47681
```

```
## temp                          157.2094    10.2612  15.321  < 2e-16 ***
## weathersitcloudy/misty        -12.8903     1.9643  -6.562 5.60e-11 ***
## weathersitlight rain/snow     -66.4944     2.9652 -22.425  < 2e-16 ***
## weathersitheavy rain/snow    -109.7446    76.6674  -1.431  0.15234
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 76.5 on 8605 degrees of freedom
## Multiple R-squared:  0.6745, Adjusted R-squared:  0.6731
## F-statistic: 457.3 on 39 and 8605 DF,  p-value: < 2.2e-16
```

Keterangan :

`mod.lm2` lebih sesuai untuk interpretasi dalam model regresi karena kontrast sum lebih baik dalam menangkap efek variabel kategorikal.

```
sum((predict(mod.lm) - predict(mod.lm2))^2)

## [1] 1.586608e-18
```

Keterangan :

Jika hasilnya nol atau sangat kecil, berarti `mod.lm` dan `mod.lm2` memberikan prediksi yang hampir sama.

```
all.equal(predict(mod.lm), predict(mod.lm2))

## [1] TRUE
```

Keterangan :

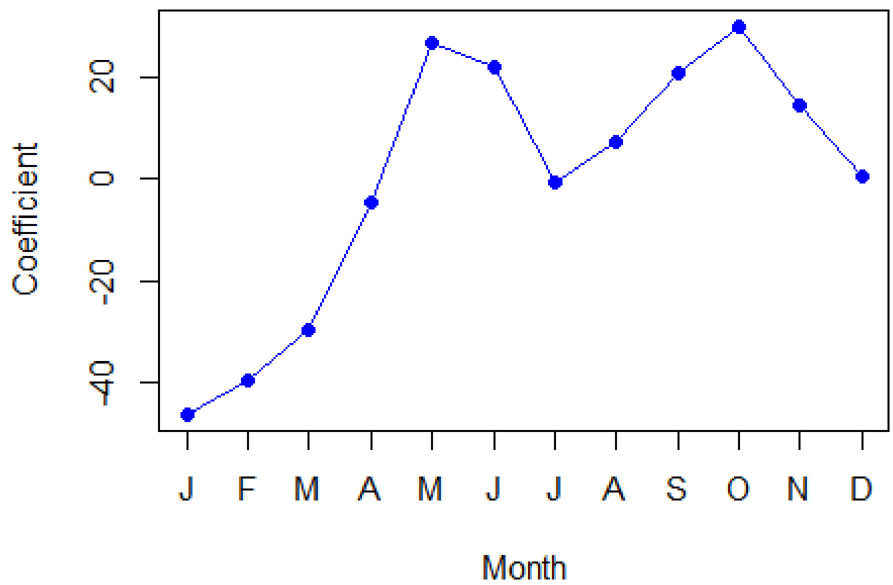
- Jika `TRUE`, berarti kedua model menghasilkan prediksi yang sama.

- Jika `FALSE`, berarti ada sedikit perbedaan karena metode estimasi atau kontrast sum.

```
coef.months <- c(coef(mod.lm2)[2:12],
    -sum(coef(mod.lm2)[2:12]))
```

Keterangan : Mempermudah analisis efek bulanan dalam model.

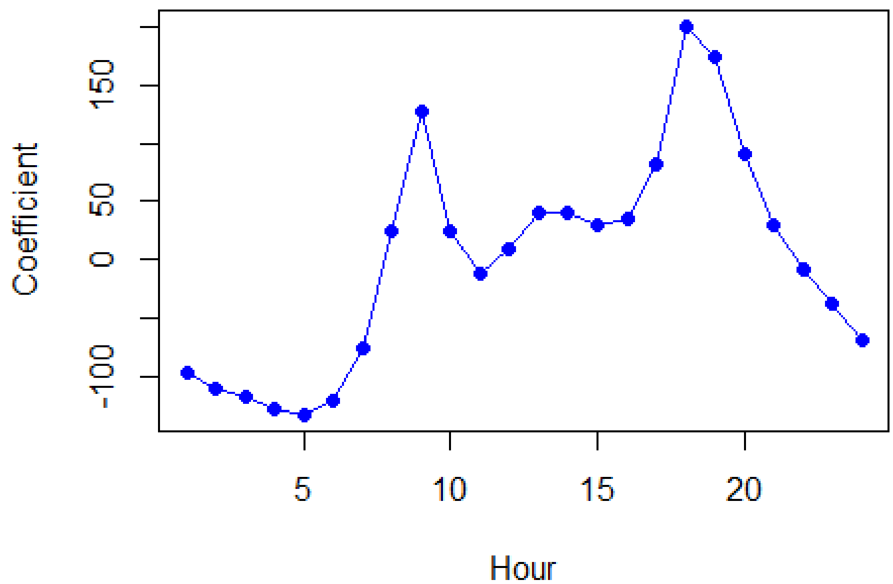
```
plot(coef.months, xlab = "Month", ylab = "Coefficient",
    xaxt = "n", col = "blue", pch = 19, type = "o")
axis(side = 1, at = 1:12, labels = c("J", "F", "M", "A",
    "M", "J", "J", "A", "S", "O", "N", "D"))
```



Keterangan :

Grafik ini menunjukkan pola musiman dalam jumlah pengguna sepeda.

```
coef.hours <- c(coef(mod.lm2)[13:35],
    -sum(coef(mod.lm2)[13:35]))
plot(coef.hours, xlab = "Hour", ylab = "Coefficient",
    col = "blue", pch = 19, type = "o")
```



Keterangan :

Grafik ini menunjukkan pola penggunaan sepeda berdasarkan waktu dalam sehari.

```
mod.pois <- glm(
    bikers ~ mnth + hr + workingday + temp + weathersit,
    data = Bikeshare, family = poisson
  )
summary(mod.pois)

##
## Call:
## glm(formula = bikers ~ mnth + hr + workingday + temp + weathersit,
##     family = poisson, data = Bikeshare)
##
## Coefficients:
##                            Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                4.118245   0.006021  683.964  < 2e-16 ***
## mnth1                     -0.670170   0.005907 -113.445  < 2e-16 ***
## mnth2                     -0.444124   0.004860  -91.379  < 2e-16 ***
## mnth3                     -0.293733   0.004144  -70.886  < 2e-16 ***
## mnth4                      0.021523   0.003125    6.888 5.66e-12 ***
## mnth5                      0.240471   0.002916   82.462  < 2e-16 ***
## mnth6                      0.223235   0.003554   62.818  < 2e-16 ***
## mnth7                      0.103617   0.004125   25.121  < 2e-16 ***
## mnth8                      0.151171   0.003662   41.281  < 2e-16 ***
## mnth9                      0.233493   0.003102   75.281  < 2e-16 ***
```

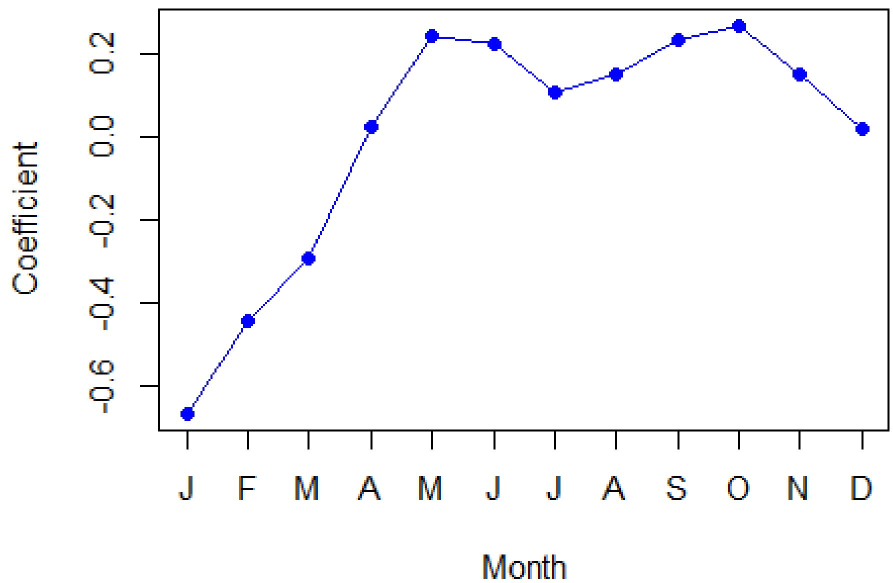
```
## mnth10                     0.267573   0.002785   96.091  < 2e-16 ***
## mnth11                     0.150264   0.003180   47.248  < 2e-16 ***
## hr1                       -0.754386   0.007879  -95.744  < 2e-16 ***
## hr2                       -1.225979   0.009953 -123.173  < 2e-16 ***
## hr3                       -1.563147   0.011869 -131.702  < 2e-16 ***
## hr4                       -2.198304   0.016424 -133.846  < 2e-16 ***
## hr5                       -2.830484   0.022538 -125.586  < 2e-16 ***
## hr6                       -1.814657   0.013464 -134.775  < 2e-16 ***
## hr7                       -0.429888   0.006896  -62.341  < 2e-16 ***
## hr8                        0.575181   0.004406  130.544  < 2e-16 ***
## hr9                        1.076927   0.003563  302.220  < 2e-16 ***
## hr10                       0.581769   0.004286  135.727  < 2e-16 ***
## hr11                       0.336852   0.004720   71.372  < 2e-16 ***
## hr12                       0.494121   0.004392  112.494  < 2e-16 ***
## hr13                       0.679642   0.004069  167.040  < 2e-16 ***
## hr14                       0.673565   0.004089  164.722  < 2e-16 ***
## hr15                       0.624910   0.004178  149.570  < 2e-16 ***
## hr16                       0.653763   0.004132  158.205  < 2e-16 ***
## hr17                       0.874301   0.003784  231.040  < 2e-16 ***
## hr18                       1.294635   0.003254  397.848  < 2e-16 ***
## hr19                       1.212281   0.003321  365.084  < 2e-16 ***
## hr20                       0.914022   0.003700  247.065  < 2e-16 ***
## hr21                       0.616201   0.004191  147.045  < 2e-16 ***
## hr22                       0.364181   0.004659   78.173  < 2e-16 ***
## hr23                       0.117493   0.005225   22.488  < 2e-16 ***
## workingday                 0.014665   0.001955    7.502 6.27e-14 ***
## temp                       0.785292   0.011475   68.434  < 2e-16 ***
## weathersitcloudy/misty    -0.075231   0.002179  -34.528  < 2e-16 ***
## weathersitlight rain/snow -0.575800   0.004058 -141.905  < 2e-16 ***
## weathersitheavy rain/snow -0.926287   0.166782   -5.554 2.79e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1052921  on 8644  degrees of freedom
## Residual deviance:  228041  on 8605  degrees of freedom
## AIC: 281159
##
## Number of Fisher Scoring iterations: 5
```

Keterangan :

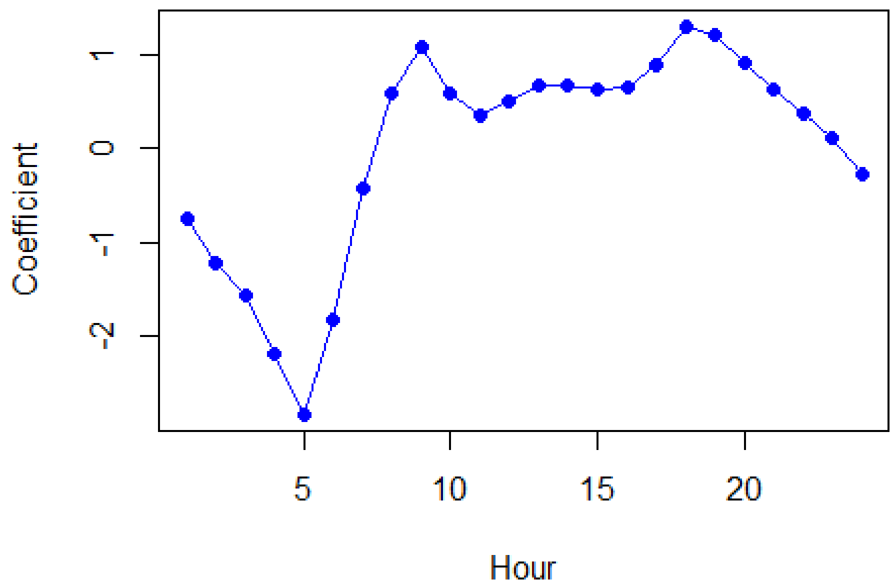
Model ini lebih sesuai dibanding regresi linear jika data `bikers` memiliki distribusi poisson.

```r
coef.mnth <- c(coef(mod.pois)[2:12],
    -sum(coef(mod.pois)[2:12]))
plot(coef.mnth, xlab = "Month", ylab = "Coefficient",
     xaxt = "n", col = "blue", pch = 19, type = "o")
```

```r
axis(side = 1, at = 1:12, labels = c("J", "F", "M", "A", "M", "J", "J", "A",
"S", "O", "N", "D"))
```



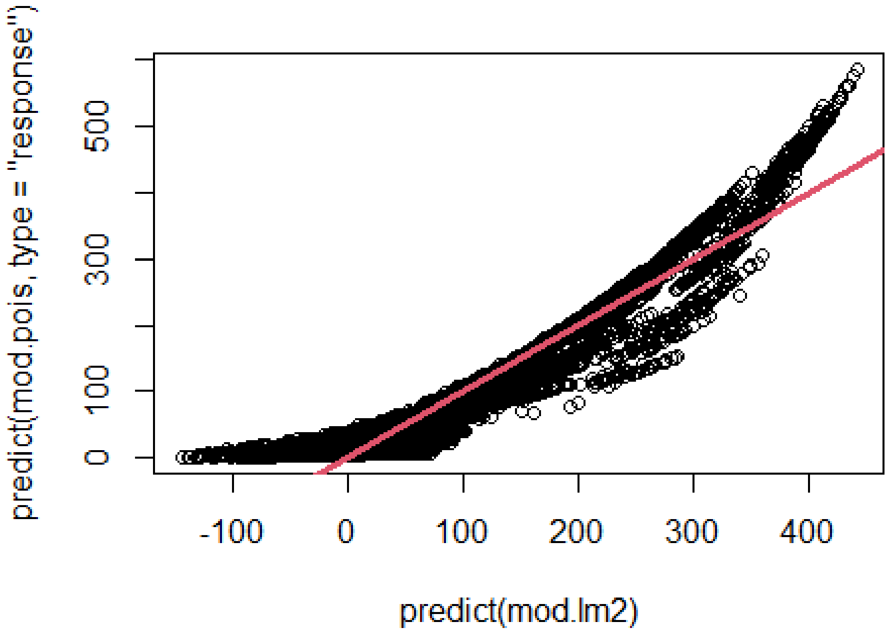

```r
coef.hours <- c(coef(mod.pois)[13:35],
     -sum(coef(mod.pois)[13:35]))
plot(coef.hours, xlab = "Hour", ylab = "Coefficient",
    col = "blue", pch = 19, type = "o")
```



Keterangan :

Pola yang dihasilkan dapat dibandingkan dengan model regresi linear untuk melihat perbedaan dalam interpretasi.

```
plot(predict(mod.lm2), predict(mod.pois, type = "response"))
abline(0, 1, col = 2, lwd = 3)
```



Keterangan :

- Jika titik-titik berada di sekitar garis merah, maka kedua model memberikan hasil prediksi yang serupa.

- Jika menyimpang, berarti kedua model memiliki perbedaan dalam estimasi jumlah pengguna sepeda.

Exercise

Nomor 13

This question should be answered using the `Weekly` data set, which is part of the `ISLR2` package. This data is similar in nature to the `Smarket` data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

```
library(ISLR2)
data(Weekly)
names(Weekly)

## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"

dim(Weekly)

## [1] 1089    9
```

b. Use the full data set to perform a logistic regression with `Direction` as the response and the five lag variables plus `Volume` as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
log_model <- glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data
= Weekly, family = binomial)
summary(log_model)

##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
```

```
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Terlihat bahwa Lag2 signifikan dengan Pr(>|z|) = 3%

c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
prediksi <- predict(log_model, type = "response")
predicted <- ifelse(prediksi > 0.5, "Up", "Down")
(confusion_matrix <- table(Prediction = predicted, Actual =
Weekly$Direction))
```

```
##            Actual
## Prediction Down  Up
##       Down   54  48
##       Up    430 557
```

```
(akurasi <- mean(predicted == Weekly$Direction))
```

```
## [1] 0.5610652
```

```
#sum(diag(confusion_matrix)) / sum(confusion_matrix)
```

Persentase prediksi: (54+557)/(54+557+48+430) = 56,1%. - Ketika pasar naik, regresi logistik benar sebesar 557/(557+48) = 92,1%. - Ketika pasar turun, regresi logistik benar sebesar 54/(430+54) = 11,2%.

Model ini tidak terlalu akurat untuk memprediksi, karena fraksi keseluruhan dari prediksi yang benar hanya sebesar 56,1%. meskipun model regresi logistik ini memprediksi kenaikan dengan baik, ada kesalahan prediksi yang menganggap penurunan sebagai kenaikan.

d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
train <- Weekly$Year < 2009
test <- Weekly$Year > 2008

log_model_d <- glm(Direction ~ Lag2, data = Weekly[train, ], family =
binomial)
prediksi_d <- predict(log_model_d, Weekly[test, ], type = "response")
predic <- ifelse(prediksi_d > 0.5, "Up", "Down")
(confusion_matrix_d <- table(Prediction = predic, Actual = Weekly[test,
]$Direction))
```

```
##           Actual
## Prediction Down Up
##       Down    9  5
##       Up     34 56
```

```
(akurasi_d <- mean(predic == Weekly[test, ]$Direction))
```

```
## [1] 0.625
```

```
#sum(diag(confusion_matrix_d)) / sum(confusion_matrix_d)
```

Berdasarkan 13b, kita tahu bahwa `Lag2` merupakan prediktor yang paling signifikan, sehingga ketika kita hanya menggunakan `Lag2` sebagai prediktor, nilai akurasi dari model regresi logistiknya menjadi meningkat (62.5%), dibanding jika kita menggunakan prediktor lain yang tidak signifikan.

Nomor 14

> In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the `Auto` data set.

f. Perform logistic regression on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

```
library(ISLR2)
data(Auto)
mpg01 <- ifelse(Auto$mpg > median(Auto$mpg), 1, 0)
data_auto <- data.frame(Auto[,-1], mpg01)

set.seed(1)
train_index <- sample(1:nrow(data_auto), nrow(data_auto) * 2/3)
train_data <- data_auto[train_index, ]
test_data <- data_auto[-train_index, ]

log_model <- glm(mpg01 ~ cylinders + horsepower + weight + displacement, data
= train_data, family = binomial)
log_probability <- predict(log_model, test_data, type = "response")
log_predict <- ifelse(log_probability > 0.5, 1, 0)
error_log <- mean(log_predict != test_data$mpg01)
cat("Error:", error_log)
```

```
## Error: 0.08396947
```

```
cat("\nAkurasi:", 1 - error_log)
```

```
##
## Akurasi: 0.9160305
```

Untuk memprediksi `mpg` dengan prediktor `Cylinders`, `Displacement`, `Horsepower`, dan `Weight` menggunakan model Regresi Logistik, memiliki error (potensi salah) sebesar 8.4%. Berarti model Regresi Logistik ini bagus untuk memodelkan data `Auto`, dan memprediksi `mpg01`dengan akurasi 91.6%.

Nomor 16

> Using the `Boston` data set, fit classification models in order to predict whether a given census tract has a crime rate above or below the median. Explore logistic regression, LDA, naive Bayes and KNN models using various sub-sets of the predictors. Describe your findings.

```
library(ISLR2)
data(Boston)
crime01 <- ifelse(Boston$crim > median(Boston$crim), 1, 0)
data_boston <- data.frame(Boston, crime01)

set.seed(1)
train_index <- sample(1:nrow(data_boston), nrow(data_boston) * 0.7)
train_data <- data_boston[train_index, ]
test_data <- data_boston[-train_index, ]

log_model <- glm(crime01 ~ lstat + dis + nox + rm + zn + indus + age + tax,
data = train_data, family = "binomial")
log_probability <- predict(log_model, test_data, type = "response")
log_predict <- ifelse(log_probability > 0.5, 1, 0)
error_log <- mean(log_predict != test_data$crime01)
cat("Error:", error_log)

## Error: 0.1447368

cat("\nAkurasi:", 1 - error_log)

##
## Akurasi: 0.8552632
```

Memprediksi `crime01` dengan prediktor `lstat`, `dis`, `nox`, `rm`, `zn`, `indus`, `age`, `tax`, menggunakan model Regresi Logistik, memiliki error (potensi salah) sebesar 14.47%. Berarti model Regresi Logistik ini bagus untuk memodelkan data `Boston`, dan memprediksi `crime01` dengan tingkat akurasi 85.53%.