Dzień 1 - Model liniowy - prognozowanie

Spis treści

Model liniowy - prognozowanie

Wersja pdf

Załadujmy ten sam model co poprzednio:

```
library(tidyverse)
devtools::install_github("kassambara/datarium")
data("marketing", package = "datarium")
model <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
```

Stwórzmy nowe dane testowe:

```
new<-data.frame( youtube = c(12, 19, 24), facebook=c(40,50,60), newspaper=c(25,55,85) )
new
```

```
##   youtube facebook newspaper
## 1      12       40        25
## 2      19       50        55
## 3      24       60        85
```

Wykonajmy prognozowanie:

```
pred<-predict(model, newdata = new)
pred
```

```
##        1        2        3
## 11.59111 13.76563 15.84863
```

lub ręcznie:

```
x.new<-cbind(rep(1,3),as.matrix(new))
pred2<-x.new %*% model$coefficients
pred2
```

```
##          [,1]
## [1,] 11.59111
## [2,] 13.76563
## [3,] 15.84863
```

Wersja pełna:

```
predf<-predict(model, new, se.fit = TRUE)
predf
```

```
## $fit
##        1        2        3
## 11.59111 13.76563 15.84863
```

```
##
## $se.fit
##         1         2         3
## 0.3068124 0.3271400 0.4201326
##
## $df
## [1] 196
##
## $residual.scale
## [1] 2.022612
```

`se.fit` to odchylenie standardowe względem średniej`fit`.

Przedziały ufności

Przedziały ufności możemy otrzymać następująco:

```
predict(model, new, se.fit = TRUE, interval = "confidence")
```

```
## $fit
##        fit      lwr      upr
## 1 11.59111 10.98603 12.19618
## 2 13.76563 13.12047 14.41080
## 3 15.84863 15.02007 16.67719
##
## $se.fit
##         1         2         3
## 0.3068124 0.3271400 0.4201326
##
## $df
## [1] 196
##
## $residual.scale
## [1] 2.022612
```

lub ręcznie za pomocą odpowiednich kwantyli rozkładu t:

```
kt <- c(-1, 1) * qt(0.05 / 2, predf$df, lower.tail = FALSE)
kt
```

```
## [1] -1.972141  1.972141
```

```
pu <- predf$fit + outer(predf$se.fit, kt)
pu
```

```
##       [,1]     [,2]
## 1 10.98603 12.19618
## 2 13.12047 14.41080
## 3 15.02007 16.67719
```

`outer` - tutaj mnożenie wyraz za wyrazem.

Innymi słowymi, z prawodpodobieństwem 95% średnie wyniki sprzedaży są zwarte w odpowiednim przedziale.

Wiarygodność przewidywań

Przedziały ufności możemy otrzymać następująco:

```
predict(model, new, se.fit = TRUE, interval = "prediction")
```

```
## $fit
##        fit       lwr      upr
## 1 11.59111  7.556598 15.62562
## 2 13.76563  9.724919 17.80635
## 3 15.84863 11.774611 19.92265
##
## $se.fit
##         1         2         3
## 0.3068124 0.3271400 0.4201326
##
## $df
## [1] 196
##
## $residual.scale
## [1] 2.022612
```

lub ręcznie:

```
se.PI <- sqrt(predf$se.fit ^ 2 + predf$residual.scale ^ 2)
wp <- predf$fit + outer(se.PI, kt)
wp
```

```
##        [,1]     [,2]
## 1  7.556598 15.62562
## 2  9.724919 17.80635
## 3 11.774611 19.92265
```

Interpretacja graficzna

Wygenerujmy więcej nowych danych:

```
n <- 20
yt <- runif(n, min = 15, max = 150)+rnorm(4,2,0.5)
fb <- runif(n, min = 30, max = 70)
np <- runif(n, min=50, max=144)
new2<-data.frame( youtube = yt, facebook=fb, newspaper=np )
pred2p<-predict(model, new2, interval = "prediction")
head(pred2p)
```

```
##        fit       lwr      upr
## 1 17.95397 13.903783 22.00416
## 2 14.32075 10.209641 18.43185
## 3 19.41012 15.376984 23.44326
## 4 22.68770 18.597181 26.77822
## 5 19.94331 15.912452 23.97416
## 6 12.88108  8.792463 16.96970
```

```
pred2c<-predict(model, new2, interval = "confidence")
head(pred2c)
```

```
##        fit      lwr      upr
## 1 17.95397 17.25191 18.65604
## 2 14.32075 13.32574 15.31576
## 3 19.41012 18.81424 20.00601
## 4 22.68770 21.78149 23.59391
```

```
## 5 19.94331 19.36310 20.52352
## 6 12.88108 11.98350 13.77866
```

```
plot(pred2c[,1], ylab="", ylim=c(min(pred2p[,2]), max(pred2p[,3])), pch=20)
lines(pred2c[,2], col="blue", lty=2)
lines(pred2p[,2], col="red", lty=3)
lines(pred2c[,3], col="blue", lty=2)
lines(pred2p[,3], col="red", lty=3)
```