

Dzień 2 - Tablica analizy wariancji - anova

Spis treści

Tablica analizy wariancji - anova	1
Zmienne ciągłe	1
Zmienne jakościowe objaśniające	3
Porównywanie modeli	5

Tablica analizy wariancji - anova

Wersja pdf

Zmienne ciągłe

Przeanalizuj kod w R:

```
library(tidyverse)
devtools::install_github("kassambara/datarium")
```

##

```
checking for file 'C:\Users\Piotrek\AppData\Local\Temp\RtmpMHMZhh\remotes2ef414dc4271\kassambara-dat
```

```
checking for file 'C:\Users\Piotrek\AppData\Local\Temp\RtmpMHMZhh\remotes2ef414dc4271\kassambara-dat
```

```
v checking for file 'C:\Users\Piotrek\AppData\Local\Temp\RtmpMHMZhh\remotes2ef414dc4271\kassambara-dat
##
```

```
- preparing 'datarium':
##
```

```
checking DESCRIPTION meta-information ...
```

```
checking DESCRIPTION meta-information ...
```

```
v checking DESCRIPTION meta-information
##
```

```
- checking for LF line-endings in source and make files and shell scripts (435ms)
##
```

```
- checking for empty or unneeded directories
##
```

```

- looking to see if a 'data/datalist' file should be added
##

- building 'datarium_0.1.0.tar.gz'
##

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

model <- lm(sales ~ youtube + facebook + newspaper, data = marketing)
summary(model)

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

## Analysis of Variance Table
##
## Response: sales
##           Df Sum Sq Mean Sq  F value Pr(>F)
## youtube    1 4773.1  4773.1 1166.7308 <2e-16 ***
## facebook    1 2225.7  2225.7  544.0501 <2e-16 ***
## newspaper   1    0.1    0.1   0.0312 0.8599
## Residuals 196  801.8    4.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Jak widać, ostatnia kolumna jest taka sama - nie otrzymujemy w przypadku zmiennych ciągłych nowych informacji.

Zmienne jakościowe objaśniające

Rozważmy przykład z danymi jakościowymi:

```
library(car)
```

```
data("Salaries")
head(Salaries)
```

```
##      rank discipline yrs.since.phd yrs.service  sex salary
## 1     Prof          B             19          18 Male 139750
## 2     Prof          B             20          16 Male 173200
## 3  AsstProf          B              4           3 Male  79750
## 4     Prof          B             45          39 Male 115000
## 5     Prof          B             40          41 Male 141500
## 6 AssocProf          B              6           6 Male  97000
```

```
model <- lm(salary ~ sex, data = Salaries)
summary(model)
```

```
##
## Call:
## lm(formula = salary ~ sex, data = Salaries)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -57290 -23502  -6828   19710 116455
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   101002      4809   21.001  < 2e-16 ***
## sexMale        14088      5065    2.782  0.00567 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30030 on 395 degrees of freedom
## Multiple R-squared:  0.01921,    Adjusted R-squared:  0.01673
## F-statistic: 7.738 on 1 and 395 DF,  p-value: 0.005667
```

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: salary
##           Df      Sum Sq    Mean Sq F value    Pr(>F)
## sex         1 6.9800e+09 6980014930  7.7377 0.005667 **
## Residuals 395 3.5632e+11  902077538
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Jak to obliczamy?

```
y<-Salaries$salary
ssm<-sum((model$fitted.values - mean(y))^2)
ssm
```

```
## [1] 6980014930
```

```
ssr<-sum((y-model$fitted.values)^2)
ssr
```

```
## [1] 356320627631
```

```
ssm/1
```

```
## [1] 6980014930
```

```
ssr/395
```

```
## [1] 902077538
```

```
f<-(ssm/1)/(ssr/395)
f
```

```
## [1] 7.737711
```

```
p<-1-pf(f, 1,395)
p
```

```
## [1] 0.005667107
```

Czemu nie wyszło to tak jak wcześniej? Wynika to z faktu, jak R interpretuje zmienne jakościowe. Ostatni level zmiennej `sex` to `Male`, więc ma przypisaną wartość 1, a potem (od końca) `Female` jako 0. Model liniowy jest zapisano jako $y = \beta_0 + \beta_1 x + \varepsilon$. Wtedy $\beta_0 + \beta_1$ dotyczy `Male`, β_0 dotyczy `Female`. Samo β_1 dotyczy różnicy między `Male` a `Female`.

Jeśli zmienna ma więcej niż 2 wartości cechy, to jest zamieniana na więcej zmiennych o dwóch wartościach (levelach).

```
levels(Salaries$rank)
```

```
## [1] "AsstProf" "AssocProf" "Prof"
```

```
res <- model.matrix(~rank, data = Salaries)
head(res)
```

```
##      (Intercept) rankAssocProf rankProf
## 1             1             0         1
## 2             1             0         1
## 3             1             0         0
## 4             1             0         1
## 5             1             0         1
## 6             1             1         0
```

```
head(Salaries$rank)
```

```
## [1] Prof      Prof      AsstProf Prof      Prof      AssocProf
## Levels: AsstProf AssocProf Prof
```

```
model2<-lm(salary ~ yrs.service + rank + discipline + sex, data = Salaries)
summary(model2)
```

```
##
```

```
## Call:
```

```
## lm(formula = salary ~ yrs.service + rank + discipline + sex,
##     data = Salaries)
```

```
##
```

```
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
```

```
## -64202 -14255 -1533 10571 99163
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  68351.67    4482.20  15.250 < 2e-16 ***
## yrs.service   -88.78     111.64  -0.795 0.426958
## rankAssocProf 14560.40    4098.32   3.553 0.000428 ***
## rankProf      49159.64    3834.49  12.820 < 2e-16 ***
## disciplineB   13473.38    2315.50   5.819 1.24e-08 ***
## sexMale       4771.25     3878.00   1.230 0.219311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22650 on 391 degrees of freedom
## Multiple R-squared:  0.4478, Adjusted R-squared:  0.4407
## F-statistic: 63.41 on 5 and 391 DF,  p-value: < 2.2e-16
```

```
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: salary
##              Df      Sum Sq   Mean Sq F value    Pr(>F)
## yrs.service    1 4.0709e+10 4.0709e+10  79.3405 < 2.2e-16 ***
## rank           2 1.0358e+11 5.1789e+10 100.9335 < 2.2e-16 ***
## discipline     1 1.7617e+10 1.7617e+10  34.3350 9.861e-09 ***
## sex            1 7.7669e+08 7.7669e+08   1.5137  0.2193
## Residuals     391 2.0062e+11 5.1310e+08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Porównywanie modeli

```
fit <- lm(sr ~ ., data = LifeCycleSavings)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: sr
##              Df Sum Sq Mean Sq F value    Pr(>F)
## pop15         1 204.12 204.118 14.1157 0.0004922 ***
## pop75         1  53.34  53.343  3.6889 0.0611255 .
## dpi           1  12.40  12.401  0.8576 0.3593551
## ddpi          1  63.05  63.054  4.3605 0.0424711 *
## Residuals    45 650.71  14.460
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## same effect via separate models
fit0 <- lm(sr ~ 1, data = LifeCycleSavings)
fit1 <- update(fit0, . ~ . + pop15)
fit2 <- update(fit1, . ~ . + pop75)
fit3 <- update(fit2, . ~ . + dpi)
fit4 <- update(fit3, . ~ . + ddpi)
anova(fit0, fit1, fit2, fit3, fit4, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: sr ~ 1
## Model 2: sr ~ pop15
## Model 3: sr ~ pop15 + pop75
## Model 4: sr ~ pop15 + pop75 + dpi
## Model 5: sr ~ pop15 + pop75 + dpi + ddpi
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1      49 983.63
## 2      48 779.51  1   204.118 14.1157 0.0004922 ***
## 3      47 726.17  1    53.343  3.6889 0.0611255 .
## 4      46 713.77  1    12.401  0.8576 0.3593551
## 5      45 650.71  1     63.054  4.3605 0.0424711 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Jeśli otrzymane prawdopodobieństwo w ostatniej kolumnie jest mniejsze niż 0,05 to stwierdzamy, że bardziej zawiły model jest wystarczająco lepszy niż prostszy model. W przeciwnym wypadku wybieramy prostszy model.