

Autor:	Radosław Smoter
Numer sprawozdania:	5
Data oddania:	15.05.2022
Prowadzący:	dr inż. Krzysztof Schiff

Probabilistyka w zastosowaniach technicznych

1 Kody

1. wczytaj Exam Anxiety.dat pod examData i wykonaj cor(examData\$Exam, examData\$Anxiety, use="complete.obs", method="pearson")

```
# Wczytaj dane.
examData <- read.delim("Exam Anxiety.dat", header = T);
# Wypisz współczynnik korelacji Pearsona.
cor(examData$Exam, examData$Anxiety, use = "complete.obs", method =
"pearson");
```

2. examData2<-examData[, c("Exam", "Anxiety", "Revise")]

```
examData2 <- examData[, c("Exam", "Anxiety", "Revise")];
```

3. cor(examData2)

```
# Macierz korelacji dla wybranych wartosci.
cor(examData2);
```

4. wczytaj plik "The biggest liar.dat" pod liarData

```
# Wczytaj dane.
liarData <- read.delim("The Biggest Liar.dat", header = T);
```

5. bootTau<-function(liarData,i) cor(liarData\$Position[i], liarData\$Creativity[i], use="complete.obs", method="kendall")

```
# Stworz funkcje, ktora przyjmuje data frame oraz iterator. Funkcja
zwraca korelacje tau Kendalla.
bootTau <- function(liarData, i) {
  cor(liarData$Position[i], liarData$Creativity[i], use = "complete.obs",
method = "kendall");
}
```

6. boot_kendall<-boot(liarData, bootTau, 2000)

```
# Funkcja boot() wywołuje funkcje statystyczna bootTau 2000 razy. Za
kazdym razem generuje losowe wartosci dla kazdego wiersza z liarData. Te
wartosci sa uzywane do wybrania probki wewnatrz funkcji bootTau.
boot_kendall <- boot(liarData, bootTau, 2000);
boot_kendall;
```

7. Wprowadź dane z tabeli i przeprowadź test korelacji pearsona na time & gender oraz time & record.

```
# Wprowadzenie danych.
some_data <- matrix(
  c(
    41, 40, 40, 38, 34, 46, 42, 42, 47, 42, 45, 46, 44, 54,
    1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0,
    0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0 ,0 ,0, 1
  ),
  ncol = 3, nrow = 14
```

```
);
colnames(some_data) <- c("time", "gender", "record");
some_df <- data.frame(some_data);
# Test korelacji Pearsona.
cor(some_df$time, some_df$gender, use = "complete.obs", method =
"pearson");
cor(some_df$time, some_df$record, use = "complete.obs", method =
"pearson");
```

8. Wczytaj pod album1 plik „Album Sales 1.dat”

```
# Wczytaj dane.
album1 <- read.delim("Album Sales 1.dat");
```

9. albumSales.1<-lm(sales~adverts, data=album1)

```
# Znajdz pasujacy model regresji liniowej. Za formule przyjmowana jest
wartosc z kolumny adverts, za dane album1.
albumSales.1 <- lm(sales ~ adverts, data = album1);
```

10. summary(albumSales.1)

```
summary(albumSales.1);
```

11. Jaki jest współczynnik b1 oraz b0 ?

```
# y = 9.612e-02 * x + 1.341e+02
```

12. album2<-read.delim(„Album Sales 2.dat”, header=TRUE)

```
# Wczytaj dane.
album2 <- read.delim("Album Sales 2.dat", header = T);
```

13. albumSales.2<-lm(sales~adverts, data=album2)

```
# Stworz modele regresji o zadanych parametrach.
albumSales.2 <- lm(sales ~ adverts, data = album2);
```

14. albumSale.3<-lm(sales~adverts+airplay+attract, data=album2)

```
albumSales.3 <- lm(sales ~ adverts + airplay + attract, data = album2);
```

15. Wyjaśnij działanie funkcji anova() na przykładzie anova(albumSales.2, albumSales.3)

```
# Porownaj czy dwie grupy sa podobne. Porownuje srednie grup, sprawdzajac
czy sa bardzo rozne od sredniej dla wszystkich grup razem.
anova(albumSales.2, albumSales.3);
```

16. Czy albumSales.3 jest lepiej dopasowany do danych niż albumSales.2 ?

```
# Model dla albumSales.3 jest lepiej dopasowany niż model dla
albumSales.2.
```

2 Wyniki

Ad. 1.

```
> # Wczytaj dane.
examData <- read.delim("Exam Anxiety.dat", header = T);
# Wypisz współczynnik korelacji Pearsona.
cor(examData$Exam, examData$Anxiety, use = "complete.obs", method = "pearson");
[1] -0.4409934
> 
```

Ad. 2.

```
> # Macierz korelacji dla wybranych wartosci.
examData2 <- examData[, c("Exam", "Anxiety", "Revise")];
cor(examData2);
```

	Exam	Anxiety	Revise
Exam	1.0000000	-0.4409934	0.3967207
Anxiety	-0.4409934	1.0000000	-0.7092493
Revise	0.3967207	-0.7092493	1.0000000

```
> 
```

Ad. 6.

```
> # Wczytaj dane.
liarData <- read.delim("The Biggest Liar.dat", header = T);
> # Stworz funkcje, ktora przyjmuje data frame oraz iterator. Funkcja zwraca korelacje tau Kendalla.
bootTau <- function(liarData, i) {
  cor(liarData$Position[i], liarData$Creativity[i], use = "complete.obs", method = "kendall");
}
> # Funkcja boot() wywołuje funkcje statystyczna bootTau 2000 razy. Za kazdym razem generuje losowe wartosci dla kazdego wiersza z liarData. Te wartosci sa uzywane do wybrania probki wewnatrz funkcji bootTau.
boot_kendall <- boot(liarData, bootTau, 2000);
> boot_kendall;
```

ORDINARY NONPARAMETRIC BOOTSTRAP

Call:

```
boot(data = liarData, statistic = bootTau, R = 2000)
```

Bootstrap Statistics :

	original	bias	std. error
t1*	-0.3002413	-0.0004720479	0.09562666

```
> 
```

Ad. 7.

```
> # Wprowadzenie danych.
some_data <- matrix(
  c(
    41, 40, 40, 38, 34, 46, 42, 42, 47, 42, 45, 46, 44, 54,
    1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0,
    0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1
  ),
  ncol = 3, nrow = 14
);
colnames(some_data) <- c("time", "gender", "record");
some_df <- data.frame(some_data);
# Test korelacji Pearsona.
cor(some_df$time, some_df$gender, use = "complete.obs", method = "pearson");
cor(some_df$time, some_df$record, use = "complete.obs", method = "pearson");
[1] -0.3567593
[1] 0.3567593
> 
```

Ad. 11.

```
> # Wczytaj dane.
album1 <- read.delim("Album Sales 1.dat");
> # Znajdz pasujacy model regresji liniowej. Za formule przyjmowana jest wartosc z kolumny adverts, za dan
e album1.
albumSales.1 <- lm(sales ~ adverts, data = album1);
> summary(albumSales.1);
# y = 9.612e-02 * x + 1.341e+02

Call:
lm(formula = sales ~ adverts, data = album1)

Residuals:
    Min       1Q   Median       3Q      Max
-152.949  -43.796   -0.393   37.040  211.866

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.341e+02  7.537e+00  17.799  <2e-16 ***
adverts      9.612e-02  9.632e-03   9.979  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65.99 on 198 degrees of freedom
Multiple R-squared:  0.3346,    Adjusted R-squared:  0.3313
F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16

> 
```

Reszta

```
> # Wczytaj dane.
album2 <- read.delim("Album Sales 2.dat", header = T);
# Stworz modele regresji o zadanych parametrach.
albumSales.2 <- lm(sales ~ adverts, data = album2);
albumSales.2;
albumSales.3 <- lm(sales ~ adverts + airplay + attract, data = album2);
albumSales.3;
# Porownaj czy dwie grupy sa podobne. Porownuje srednie grup, sprawdzajac czy sa bardzo rozne od sredniej
dla wszystkich grup razem.
anova(albumSales.2, albumSales.3);
# Model dla albumSales.3 jest lepiej dopasowany niz model dla albumSales.2.

Call:
lm(formula = sales ~ adverts, data = album2)

Coefficients:
(Intercept)      adverts
  134.13994       0.09612

Call:
lm(formula = sales ~ adverts + airplay + attract, data = album2)

Coefficients:
(Intercept)      adverts      airplay      attract
  -26.61296       0.08488       3.36743      11.08634

Analysis of Variance Table

Model 1: sales ~ adverts
Model 2: sales ~ adverts + airplay + attract
  Res.Df  RSS Df Sum of Sq    F    Pr(>F)
1    198 862264
2    196 434575  2    427690 96.447 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> 
```