Statystyka

Marcin Armacki & Marcin Woźniak

Zaktualizowano: 12.06.2020 10:10

Spis treści

1 Wprowadzenie do R

1.1 Zadanie 1

Otwórz program RStudio. Następnie utwórz nowy skrypt i zapisz go jako, na przykład, *wprowadzenie_do_R_zadania.R*. W tym skrypcie możesz napisać rozwiązania następujących zadań.

CTRL + SHIFT + N

1.2 Zadanie 2

Użyj funkcji *rep()*, aby utworzyć wektor logiczny, zaczynając od trzech wartości prawda, następnie czterech wartości fałsz, po których następują dwie wartości prawda i wreszcie pięć wartości fałsz. Przypisz ten wektor logiczny do zmiennej x. Na koniec przekonwertuj ten wektor na wektor numeryczny. Jak zmieniły się wartości prawda i fałsz?

```
x <- c(rep(TRUE,3), rep(FALSE,4), rep(TRUE,2), rep(FALSE,5))
x
x <- as.numeric(x)
x
```

1.3 Zadanie 3

Palindromem nazywamy wektor, którego elementy czytane od końca tworzą ten sam wektor co elementy czytane od początku. Utwórz taki wektor 100 liczb przy czym pierwsze 20 liczb to kolejne liczby naturalne, następnie występuje 10 zer, następnie 20 kolejnych liczb parzystych, a pozostałe elementy określone są przez palindromiczność (warunek symetrii).

```
vector <- c(1:20, rep(0,10), seq(2,40,by=2))
vector <- c(vector, rev(vector))
vector
```

1.4 Zadanie 4

Z wektora *letters* wybierz litery na pozycjach 5, 10, 15, 20, 25.

```
letters[seq(5,25,by=5)]
```

1.5 Zadanie 5

Utwórz wektor liczb naturalnych od 1 do 1000, a następnie zamień liczby parzyste na ich odwrotności.

```
vector <- 1:1000
vector[vector %% 2 == 0] <- 1/vector[vector %% 2 == 0]
vector
```

1.6 Zadanie 6

Uporządkuj elementy wektora (6,3,4,5,2,3) od największego do najmniejszego wykorzystując funkcję *order()*.

```
vector <- c(6,3,4,5,2,3)
vector[order(vector, decreasing=TRUE)]
```

1.7 Zadanie 7

Wyznacz znaki elementów wektora (1,876;1,123;0,123;0;0,123;1,123;1,876). Następnie zaokrąglij elementy tego wektora do dwóch miejsc po przecinku. Na koniec wyznacz część całkowitą każdego elementu nowego wektora.

```
vector <- c(−1.876, −1.123, −0.123, 0, 0.123, 1.123, 1.876)
as.integer(vector)
round(vector, digits=2)
floor(vector)
```

1.8 Zadanie 8

Wyznacz pierwiastek kwadratowy z każdej liczby naturalnej od 1 do 100 milionów. Najpierw wykonaj to polecenie korzystając z odpowiedniej funkcji wbudowanej w R, a następnie wykorzystując potęgowanie. Który sposób działa szybciej? Wskazówka: Do badania długości czasu działania programu można wykorzystać funkcję *Sys.time()*.

```
start <- Sys.time()
temp <- sqrt(1:100000000)
end <- Sys.time()
difftime(end, start)
start <- Sys.time()
temp <- (1:100000000)^0.5
end <- Sys.time()
difftime(end, start)
```

1.9 Zadanie 9

W pakiecie schoolmath znajduje się zbiór danych primlist, który zawiera liczby pierwsze pomiędzy 1 a 9999999. Znajdź największą liczbę pierwszą mniejszą od 1000. Ile jest liczb pierwszych większych od 100 a mniejszych od 500?

```
#install.packages("schoolmath")
library(schoolmath)
data(primlist)
max(primlist[primlist < 1000])
primlist[primlist > 100 & primlist < 500]
```

1.10 Zadanie 10

Wyznacz wszystkie kombinacje wartości wektorów (a,b) i (1,2,3) za pomocą funkcji *rep()* i *paste()*.

```
vector1 <- c('a','b')
vector2 <- c(1,2,3)
c(paste(rep(vector1[1], length(vector2)), vector2, sep=""),
paste(rep(vector1[2], length(vector2)), vector2, sep=""))
```

1.11 Zadanie 11

Utwórz wektor 30 napisów następującej postaci: liczba.litera, gdzie liczba to kolejne liczby naturalne od 1 do 30 a litera to trzy wielkie litery X, Y, Z występujące cyklicznie.

```
paste(1:30, c('X','Y','Z'), sep=".")
```

1.12 Zadanie 12

W pewnych sytuacjach przydatna może się okazać tzw. kategoryzacja zmiennych, czyli inny podział na kategorie niżby wynikał z danych. Wygeneruj 100 obserwacji, które są odpowiedziami na pytania ankiety, każda odpowiedź może przyjąć jedną z wartości: ‘a’, ‘b’, ‘c’, ‘d’, ‘e’. Dokonaj kategoryzacji w taki sposób, aby kategoria 1 obejmowała odpowiedzi ‘a’ i ‘b’, 2 odpowiedzi ‘c’ i ‘d’ oraz 3 odpowiedź ‘e’. Wskazówka: Wykorzystaj funkcję sample() oraz funkcję recode() z pakietu car.

```
#install.packages("car")
library(car)
vector<- sample(c("a","b","c","d","e"), 100, replace = TRUE)
vector
recode(vector, "'a'=1 ; 'b'=1 ; 'c'=2 ; 'd'=2 ; 'e'=3")
```

2 Wprowadzenie do R cd.

2.1 Zadanie 1

Skonstruuj listę o nazwie mojalista, której pierwszym elementem będzie dwuelementowy wektor napisów zawierający Twoje imię i nazwisko, drugim elementem będzie liczba π , trzecim funkcja służąca do obliczania pierwiastka kwadratowego, a ostatni element listy to wektor złożony z liczb 0,02, 0,04, ... , 1.
Następnie usuń elementy numer jeden i trzy z tej listy. Na zakończenie, wyznacz listę zawierającą wartości funkcji gamma Eulera dla elementów listy mojalista.

```
mojalista <- (list(c("Marcin","Woźniak"), pi, sqrt,seq(0.02,1,0.02)))
str(mojalista)
mojalista[c(1,3)] <- NULL
str(mojalista)
sapply(mojalista,gamma)
```

2.2 Zadanie 2

Wyznacz rząd, wyznacznik, wartości własne, wektory własne oraz sumy i średnie arytmetyczne dla kolejnych wierszy i kolumn dla następującej macierzy:

```
library(Matrix)
x <- cbind(c(1,2,1),c(5,0,2),c(3,5,1))
det(x)
rankMatrix(x)
eigen(x)
solve(x)
rowSums(x)
rowSums(x)/3
colSums(x)
colSums(x)/3
x %*% solve(x)
```

2.3 Zadanie 3

Utwórz wektor kwadratów 100 pierwszych liczb naturalnych. Następnie zlicz, które cyfry oraz jak często występują na pozycji jedności w kolejnych elementach tego wektora.

```
library(plyr)
zad3 <- (seq(1,100))^2
matrix(c(count(zad3 %% 10)$x, count(zad3 %% 10)$freq), nrow = 2, ncol = 6,
    byrow = TRUE)
```

2.4 Zadanie 4

Za pomocą funkcji outer() wyznacz tabliczkę mnożenia dla liczb mniejszych od 6.

```
outer(1:5, 1:5, function(x, y) paste(x, "*", y, "=", x * y))
```

2.5 Zadanie 5

Odczytaj zbiór danych dane1.csv a następnie:

- Z odczytajnej ramki danych wyświetl tylko parzyste wiersze
- Korzystając z operatów logicznych wyświetl tylko wiersze odpowiadające pacjentkom starszym niż 50 lat z przerzutami do węzłów chłonnych

```
dane <- read.csv(url("http://ls.home.amu.edu.pl/data_sets/dane1.csv"), header=
    TRUE, sep=";")
dane[c(seq(0,length(dane$Wiek),2)), ]
dane[((dane$Wiek>50) & (dane$Wezly.chlonne == 1)),]
```

2.6 Zadanie 6

```
zad6 <- data.frame(NYF = c(32,33,41,52,62,72,77,75,68,58,47,35))
colnames(zad6)[1]<- "NY_F"
row.names(zad6)<-c("Styczeń","Luty","Marzec","Kwiecień","Maj","Czerwiec","Lipiec","
    Sierpień","Wrzesień","Październik","Listopad","Grudzień")
zad6
zad6$NY_C <- (zad6$NY_F - 32) * 5/9
zad6
colnames(zad6)[1]<- "NY_Fahrenheit"
colnames(zad6)[2]<- "NY_Celsiusz"
zad6
zad6$NY_Fahrenheit <- NULL
zad6
write.table(zad6, "NYtemp.RData")
```

3 Programowanie w R

3.1 Zadanie 1

Oblicz iloczyn elementów dowolnego wektora x za pomocą pętli while, repeat i for (każdej z osobna).

```
zad31a <- function(x){
    y=1
    for (i in x) y <- y*i
    y}

zad31b <- function(x){
    i <- 1
    y <- 1
    repeat {
        y <- y+y*x[i]
        i <- i + 1
        if (i == length(x)) break
        }
    y}

zad31c <- function(x){
    i <- 1
    z <- 1
    while (i <= length(x)){
        z <- z * x[i]
        i <- i + 1}
        z}

zad31a(1:5)
zad31b(1:5)
zad31c(1:5)
```

3.2 Zadanie 2

Ile liczb postaci $\binom{n}{r}$ jest większych od miliona dla $1 \leq r \leq n \leq 100$?

```
acc = 0
for (n in 1:100) {
  for (r in 1:n) {
    if (choose(n, r) > 1000000) {
      acc <- acc + 1
    }
  }
}
acc
```

3.3 Zadanie 3

Napisz funkcję, która sprawdza czy wektor jest palindromem.

```
palindrom <- function(x) {
  all.equal(x, rev(x))
  palindrom(c(1, 2, 3, 3, 2, 1))}
```

3.4 Zadanie 4

Napisz funkcję zamieniającą miarę kąta podaną w stopniach na radiany. Sprawdź działanie tej funkcji dla kątów o mierze: $0°, 30°, 45°, 60°, 90°$

```
degrees = c(0,30,45,60,90)
deg2rad <- function(a) {
  return(a*pi/180)
}

trig <- data.frame(
  sin = sin(deg2rad(degrees)),
  cos = cos(deg2rad(degrees)),
  tg = tan(deg2rad(degrees)),
  ctg = 1/tan(deg2rad(degrees))
)
trig
```

3.5 Zadanie 5

Napisz funkcję, której argumentem będzie wektor liczbowy a wynikiem wektor zawierający trzy najmniejsze i trzy największe liczby w tym wektorze. W przypadku argumentu krótszego niż trzy liczby, funkcja ma zwracać komunikat o błędzie z komentarzem „za krótki argument"

```
x <- c(2, 6, 1, 5, 7, 3, 4)
zad5 <- function(x) {
    if (length(x) < 3 ){
        stop("za krótki argument")}
    sorted = sort(x)
    return(c(sorted[1:3], sorted[(length(x)-2):length(x)]))}
zad5(x)
zad5(c(2,6))
```

4 Statystyka opisowa

4.1 Zadanie 1

Zmienna wynik w pliku ankieta.txt opisuje wyniki badania działalności prezydenta pewnego miasta. Wybrano losowo 100 mieszkańców miasta i zadano im następujące pytanie: Jak oceniasz działalność prezydenta miasta? Dostępne były następujące odpowiedzi: zdecydowanie dobrze (a), dobrze (b), źle (c), zdecydowanie źle (d), nie mam zdania (e). Jakiego typu jest ta zmienna? Jakie są możliwe wartości tej zmiennej?

Jest to dyskretna zmienna jakościowa.
Możliwe wartości zmiennej, to: a, b, c, d, e.

- Zaimportuj dane z pliku ankieta.txt do zmiennej ankieta.

```
ankieta <- read.delim(url("http://ls.home.amu.edu.pl/data_sets/ankieta.txt"))
ankieta
```

- Przedstaw rozkład empiryczny zmiennej wynik za pomocą szeregu rozdzielczego.

```
data.frame(cbind(liczebnosc = table(ankieta$wynik),
                 procent = prop.table(table(ankieta$wynik))))
```

- Przedstaw rozkład empiryczny zmiennej wynik tylko dla osób z wykształceniem podstawowym za pomocą szeregu rozdzielczego.

```
data.frame(cbind(liczebnosc = table(ankieta$wynik[ankieta$szkola == "p"]),
                 procent = prop.table(table(ankieta$wynik[ankieta$szkola == "
                     p"]))))
```

- Zilustruj wyniki ankiety za pomocą wykresu słupkowego i kołowego.

```
barplot(table(ankieta$wynik),
        xlab = "Odpowiedzi", ylab = "Liczebność",
        col = c("black", "red", "green", "blue", "cyan"),
        main = "Rozkład empiryczny zmiennej wynik")

barplot(prop.table(table(ankieta$wynik)),
        xlab = "Odpowiedzi", ylab = "Liczebność",
        col = c("black", "red", "green", "blue", "cyan"),
        main = "Rozkład empiryczny zmiennej wynik")

pie(table(ankieta$wynik))
```

- Zilustruj wyniki ankiety za pomocą wykresu słupkowego z podziałem na kobiety i mężczyzn.

```
height <- cbind(table(ankieta$wynik[ankieta$plec == "k"]), table(ankieta$
    wynik[ankieta$plec == "m"]))
barplot(height, beside = TRUE, legend.text = TRUE,
        col=c("black", "red", "green", "blue", "cyan"),
        names = c("k","m"))
```

- Zinterpretuj powyższe wyniki (tabelaryczne i graficzne).

Odpowiedzi prezentują stosunkowo równomierny rozkład, z przewagą odpowiedzi "źle", która ma największy, bo 29% udział. Tylko 10% ankietowanych oceniło działalność prezedenta miasta na "zdecydowanie dobrze".
Większość badanych stanowiły kobiety (68%).
Mężczyźni pozytywniej oceniali działalność prezedenta miasta. Przeważała odpowiedź "dobrze".
Pośród kobiet przeważa negatywna ocena.

4.2 Zadanie 2

Przebadano 200 losowo wybranych 5-sekundowych okresów pracy centrali telefonicznej. Rejestrowano liczbę zgłoszeń. Wyniki są zawarte w pliku Centrala.RData. Jakiego typu jest ta zmienna? Jakie są możliwe wartości tej zmiennej?

Jest to zmienna dyskretna, ilościowa, przyjmująca wartości całkowite od 0 do 5.

- Zaimportuj dane z pliku Centrala.RData.

```
load(url("http://ls.home.amu.edu.pl/data_sets/Centrala.RData"))
head(Centrala)
```

- Przedstaw rozkład empiryczny liczby zgłoszeń za pomocą szeregu rozdzielczego.

```
data.frame(cbind(liczebnosc = table(Centrala),
                 procent = prop.table(table(Centrala))))
```

- Zilustruj liczbę zgłoszeń za pomocą wykresu słupkowego i kołowego.

```
barplot(table(Centrala),
        xlab = "Liczba zgłoszeń", ylab = "Liczebność",
        col = c("black", "red", "green", "blue", "cyan", "magenta"),
        main = "Rozkład empiryczny listy zgłoszeń")
```

```
barplot(prop.table(table(Centrala)),
        xlab = "Liczba zgłoszeń", ylab = "Prawdopodobieństwo",
        col = c("black", "red", "green", "blue", "cyan", "magenta"),
        main = "Rozkład empiryczny listy zgłoszeń"))
```

```
pie(table(Centrala))
```

- Obliczyć średnią z liczby zgłoszeń, medianę liczby zgłoszeń, odchylenie standardowe liczby zgłoszeń i współczynnik zmienności liczby zgłoszeń.

```
mean(Centrala$Liczba)
```

```
median(Centrala$Liczba)
```

```
sd(Centrala$Liczba)
```

```
sd(Centrala$Liczba) / mean(Centrala$Liczba) * 100
```

Największy udział miały zgłoszenia w liczebności równej 1 (ok. 34% wszystkich zgłoszeń).
Średnio na okres 5 sekund przypadały 2 zgłoszenia.
Występuje tu asymetria prawostronna rozkładu.
Odchylenie standardowe wskazuje na znaczące odchylenie wartości od średniej.
Współczynnik zmienności wskazuje na bardzo duże zróżnicowanie populacji.

4.3 Zadanie 3

Notowano pomiary średniej szybkości wiatru w odstępach 15 minutowych wokół nowo powstającej elektrowni wiatrowej. Wyniki są następujące:

0.9	6.2	2.1	4.1	7.3
1.0	4.6	6.4	3.8	5.0
2.7	9.2	5.9	7.4	3.0
4.9	8.2	5.0	1.2	10.1
12.2	2.8	5.9	8.2	0.5

Jakiego typu jest ta zmienna? Jakie są możliwe wartości tej zmiennej?

Jest to ilościowa zmienna dyskretna.
Zmienna przyjmuje wartości w przedziale 0.5 do 12.2.

- Przedstaw rozkład empiryczny badanej zmiennej za pomocą szeregu rozdzielczego.

```
vector <- c(0.9, 6.2, 2.1, 4.1, 7.3, 1.0, 4.6, 6.4, 3.8, 5.0, 2.7, 9.2, 5.9, 7.4, 3.0, 4.9,
    8.2, 5.0, 1.2, 10.1, 12.2, 2.8, 5.9, 8.2, 0.5)
vector_hist <- hist(vector, plot=FALSE)$breaks
data.frame(cbind(liczebnosc = table(cut(vector, breaks = vector_hist)),
                 procent = prop.table(table(cut(vector, breaks = vector_hist
                     )))))
```

- Zilustruj rozkład empiryczny średniej szybkości wiatru za pomocą histogramu i wykresu pudełkowego. Jakie są zalety i wady tych wykresów?

```
hist(vector,
     xlab = "Średnia szybkość wiatru",
     main = "Rozkład empiryczny średniej szybkości wiatru")
rug(jitter(vector))
```

```
hist(vector,
     xlab = "Średnia szybkość wiatru",
     main = "Rozkład empiryczny średniej szybkości wiatru",
     probability = TRUE,
     col = "lightblue")
lines(density(vector), col = "green", lwd = 2)
```

```
boxplot(vector,
        ylab = "Średnia szybkość wiatru",
        main = "Rozkład empiryczny średniej szybkości wiatru")
```

- Obliczyć średnią, medianę, odchylenie standardowe, współczynnik zmienności, współczynnik asymetrii i kurtozę średniej szybkości wiatru.

```
mean(vector)
```

```
median(vector)
```

```
sd(vector)
```

```
sd(vector) / mean(vector) * 100
```

```
#install.packages("e1071")
library(e1071)
skewness(vector)
```

```
library(e1071)
kurtosis(vector)
```

- Zinterpretuj powyższe wyniki (tabelaryczne, graficzne i liczbowe).

Największy udział rozkładu posiadają wartości średniej prędkości wiatru w przedziale od 4 do 6 (28%).
Średnia prędkość wiatru w 15−minutowych odstępach wynosi 5.
Odchylenie wartości od średniej jest dość duże.
Współczynnik zmienności wskazuje na duże zróżnicowanie wartości populacji.
Współczynnik skośności wskazuje na prawostronną asymetrię rozkładu.
Ujemna kurtoza wskazuje na spłaszczony rozkład zmiennej.

4.4 Zadanie 4

Napisz funkcję wspolczynnik_zmiennosci(), która oblicza wartość współczynnika zmienności dla danego wektora obserwacji. Funkcja powinna mieć dwa argumenty:

- x - wektor zawierający dane,
- na.rm - wartość logiczna (domyślnie FALSE), która wskazuje czy braki danych (obiekty NA) mają być zignorowane.

Funkcja zwraca wartość współczynnika zmienności wyrażoną w procentach. Ponadto funkcja sprawdza, czy wektor x jest wektorem numerycznym. W przeciwnym razie zostanie zwrócony błąd z następującym komunikatem: „argument nie jest liczbą”. Przykładowe wywołania i wyniki funkcji są następujące:

```
x <− c(1, NA, 3)
wspolczynnik_zmiennosci(x)
## [1] NA
wspolczynnik_zmiennosci(x, na.rm = TRUE)
## [1] 70.71068
wspolczynnik_zmiennosci()
## Error in wspolczynnik_zmiennosci() :
## argument "x" is missing, with no default
wspolczynnik_zmiennosci(c("x", "y"))
## Error in wspolczynnik_zmiennosci(c("x", "y")) : argument nie jest liczbą
```

```
wspolczynnik_zmiennosci <− function(x, na.rm=FALSE) {
  if (!is.numeric(x)) stop("argument nie jest liczbą")
  if (na.rm) return(sd(na.omit(x)) / mean(na.omit(x)) * 100)
  else return(NA)
}
```

5 Model statystyczny i estymacja punktowa

5.1 Zadanie 1

- Niech X=(X1,. . . ,Xn) będzie próbą prostą z populacji o rozkładzie jednostajnym U(a,b). Pokaż, że estymatory metody momentów parametrów a i b w rozkładzie jednostajnym U(a,b) są postaci: a=¯X3~S, b=¯X+3~S, gdzie ¯X=1nni=1Xi oraz ~S=1nni=1(Xi¯X)2.

Jest na necie.

- Oblicz wartości tych estymatorów dla danych z przykładu dotyczącego czasu oczekiwania na tramwaj.

```
install.packages("EnvStats")
library(EnvStats)
load(url("http://ls.home.amu.edu.pl/data_sets/czas_oczek_tramwaj.RData"))
(MME <- EnvStats::eunif(czas_oczek_tramwaj, method = "mme"))
```

- Zilustruj otrzymane teoretyczne funkcje gęstości korzystające z ENW i EMM na histogramie.

```
hist(czas_oczek_tramwaj,
     xlab = "Czas oczekiwania na tramwaj",
     main = "Rozkład empiryczny i teoretyczny czasu oczekiwania na tramwaj",
     probability = TRUE,
     col = "white")
lines(density(czas_oczek_tramwaj), col = "red", lwd = 2)
curve(dunif(x, min(czas_oczek_tramwaj), max(czas_oczek_tramwaj)),
      add = TRUE, col = "blue", lwd = 2)
curve(dunif(x, MME$parameters[1], MME$parameters[2]),
      add = TRUE, col = "green", lwd = 2)
legend(x = 5, y = 0.04, legend = c("empiryczny", "teoretyczny ENW", "
    teoretyczny EMM"), col = c("red", "blue", "green"), lwd = 2)
```

5.2 Zadanie 2

Przebadano 200 losowo wybranych 5-sekundowych okresów pracy centrali telefonicznej. Rejestrowano liczbę zgłoszeń. Wyniki są zawarte w pliku Centrala.RData.

- Zasugeruj rozkład teoretyczny badanej zmiennej.

```
load(url("http://ls.home.amu.edu.pl/data_sets/Centrala.RData"))
library(fitdistrplus)
library(logspline)
data <- unlist(Centrala, use.names = FALSE)
descdist(data, discrete = TRUE)
fit.pois <- fitdist(data, "pois")
fit.norm <- fitdist(data, "norm")
plot(fit.pois)
plot(fit.norm)
# wybieramy rozkład Poissona
```

- Oblicz wartość estymatora parametru rozkładu teoretycznego.

```
mean(data)
```

- Porównaj empiryczne prawdopodobieństwa wystąpienia poszczególnych wartości liczby zgłoszeń w próbie z wartościami teoretycznymi uzyskanymi na podstawie rozkładu teoretycznego.

```
values <- dpois(sort(unique(data)), lambda = mean(data))
sum(values)

nums <- matrix(
  c(prop.table(table(Centrala)), values),
  nrow = 2,
  byrow = TRUE
)
rownames(nums) <- c("empiryczny", "teoretyczny")
colnames(nums) <- sort(unique(data))
nums

barplot(
  nums,
  xlab = "Liczba zgłoszeń", ylab = "Prawdopodobieństwo",
  main = "Rozkłady empiryczny i teoretyczny liczby błędów",
  col = c("red", "blue"),
  legend = rownames(nums),
  beside = TRUE
)
```

- Sprawdź dopasowanie rozkładu teoretycznego za pomocą wykresy kwantyl-kwantyl.

```
qqplot(rpois(length(data), mean(data)), data)
qqline(data, distribution = function(values) { qpois(values, lambda = 1/mean(
    values)) })
```

- Czy na podstawie powyższych rozważań rozkład teoretyczny wydaje się odpowiedni?

```
Jest całkiem cacy.
```

- Oblicz prawdopodobieństwo empiryczne i teoretyczne, że liczba zgłoszeń jest mniejsza niż 4.

```
length(data[data<4])/length(data)
ppois(3, lambda = mean(data))
```

5.3 Zadanie 3

To na kartce się robi. Estymatory z wykładów trzeba ogarnąć.

5.4 Zadanie 4

Notowano pomiary średniej szybkości wiatru w odstępach 15 minutowych wokół nowo powstającej elektrowni wiatrowej. Wyniki są następujące:

0.9	6.2	2.1	4.1	7.3
1.0	4.6	6.4	3.8	5.0
2.7	9.2	5.9	7.4	3.0
4.9	8.2	5.0	1.2	10.1
12.2	2.8	5.9	8.2	0.5

- Zasugeruj rozkład teoretyczny badanej zmiennej.

 # One example where the Rayleigh distribution naturally arises is when wind velocity is analyzed in two dimensions.
 # Bierzemy rozkład Rayleigha

- Oblicz wartość ENW parametru rozkładu teoretycznego.

data <− **c**(0.9, 6.2, 2.1, 4.1, 7.3, 1.0, 4.6, 6.4, 3.8, 5.0, 2.7, 9.2, 5.9, 7.4, 3.0, 4.9, 8.2, 5.0, 1.2, 10.1, 12.2, 2.8, 5.9, 8.2, 0.5)
ENW = **mean**(**data**^2)

- Porównaj rozkład empiryczny wystąpienia poszczególnych wartości średniej szybkości wiatru w próbie z wartościami teoretycznymi uzyskanymi na podstawie rozkładu teoretycznego.

```
hist(
  data,
  xlab = "Średnia szybkość wiatru",
  main = "Rozkład empiryczny i teoretyczny średniej szybkości wiatru",
  probability = TRUE,
  col = "lightgreen"
)
lines(density(data), col = "red", lwd = 2)
curve(VGAM::drayleigh(x, sqrt(ENW / 2)), col = "blue", add = TRUE, lwd =
    2)
legend(x = 8, y = 0.12, legend = c("empiryczny", "teoretyczny"), col = c("red",
    "blue"), lwd = 2)
```

- Sprawdź dopasowanie rozkładu teoretycznego za pomocą wykresy kwantyl-kwantyl.

```
probs <- VGAM::drayleigh(data, scale = sqrt(ENW/2))
qqplot(VGAM::rrayleigh(length(data), scale = sqrt(ENW/2)), data, xlab = "
    Kwantyle teoretyczne", ylab = "Kwantyle empiryczne")
qqline(data, distribution = function(probs) { VGAM::qrayleigh(probs, scale =
    sqrt(ENW/2)) })
```

- Czy na podstawie powyższych rozważań rozkład teoretyczny wydaje się odpowiedni?

```
Jest takie prawdopodobieństwo.
```

- Oblicz empiryczne i teoretyczne prawdopodobieństwo, że średnia szybkość wiatru jest zawarta w przedziale [4,8].

```
mean(c(data >= 4 & data <= 8))

# To nie wychodzi, ale nie wiem dlaczego
f <- function(x) {VGAM::drayleigh(x, scale = sqrt(ENW/2))}
integrate(f, lower = 4, upper = 8)
```

- Oblicz wartość ENW dla wartości oczekiwanej i wariancji rozkładu teoretycznego.

```
(mean <- 1.253*sqrt(ENW/2))
(variance <- 0.429*(sqrt(ENW/2))^2)
```

6 Przedziały ufności

6.1 Zadanie 1

Przebadano 200 losowo wybranych 5-sekundowych okresów pracy centrali telefonicznej. Rejestrowano liczbę zgłoszeń. Wyniki są zawarte w pliku Centrala.RData. Wykorzystując przyjęty wcześniej model statystyczny dla tych danych, wyznacz (trzema metodami) przedział ufności dla parametru rozkładu teoretycznego.

```
load(url("http://ls.home.amu.edu.pl/data_sets/Centrala.RData"))

library(EnvStats)

estimation1 <- function(x, conf_level = 0.95) {
  estimate <- epois(
    x,
    ci = TRUE,
    ci.method = "pearson",
    conf.level = conf_level
  )$interval$limits

  return(c(estimate))
}
estimation1(Centrala$Liczba)

estimation2 <- function(x, conf_level = 0.95) {
  estimate <- epois(
    x,
    ci = TRUE,
    ci.method = "pearson.hartley.approx",
    conf.level = conf_level
  )$interval$limits

  return(c(estimate))
}
estimation2(Centrala$Liczba)

estimation3 <- function(x, conf_level = 0.95) {
  estimate <- epois(
    x,
    ci = TRUE,
```

```
    ci.method = "normal.approx",
    conf.level = conf_level
  )$interval$limits

  return(c(estimate))
}
estimation3(Centrala$Liczba)
```

6.2 Zadanie 2

Zmienna w pliku awarie.txt opisuje wyniki 50 pomiarów czasu bezawaryjnej pracy danego urządzenia (w godzinach). Wykorzystując przyjęty na wykładzie model statystyczny dla tych danych wyznacz granice przedziału ufności dla wartości oczekiwanej i wariancji rozkładu teoretycznego.

```
data <- read.delim(url("http://ls.home.amu.edu.pl/data_sets/awarie.txt"), header =
    FALSE)$V1
install.packages("fitdistrplus")
install.packages("logspline")
library(fitdistrplus)
library(logspline)
descdist(data, discrete = FALSE)
fit.exp <- fitdist(data/10, "exp")
fit.beta <- fitdist(data/10000, "beta")

plot(fit.exp)
plot(fit.beta)

library(EnvStats)

conf_intervals <- eexp(
    data,
    ci = TRUE,
    ci.method = "exact",
    conf.level = 0.95)$interval$limits[c("UCL", "LCL")]

expected_value <- conf_intervals ** -1
expected_value

variation <- conf_intervals ** -2
```

variation

6.3 Zadanie 3

Niech X=(X1,. . . ,Xn) będzie próbą prostą z populacji o rozkładzie Rayleigha R(), >0. Napisz funkcję median_cint(), która implementuje następujący przybliżony przedział ufności dla mediany ln2 tego rozkładu: (...)

```
install.packages("SciViews")
library(SciViews)
print.confint <- function(x) {
  cat(x$conf_level * 100, "percent confidence interval:", "\n")
  cat(x$l, " ", x$r, "\n")
}

summary.confint <- function(x) {
  cat("\n", "Confidence interval of", x$title, "\n", "\n")
  cat(x$conf_level * 100, "percent confidence interval:", "\n")
  cat(x$l, " ", x$r, "\n")
  cat("sample estimate", "\n")
  cat(x$est, "\n")
}

median_cint <- function(x, conf_level = 0.95) {

  value = 1 - (1 - conf_level) / 2
  z = qnorm(value, mean = 0, sd = 1)
  LCL = sqrt(ln(2) * mean(x ** 2) * (1 - z / sqrt(length(x))))
  UCL = sqrt(ln(2) * mean(x ** 2) * (1 + z / sqrt(length(x))))
  ENW = mean(cbind(LCL, UCL))

  result = list(
    title = "mediana",
    est = ENW,
    l = LCL,
    r = UCL,
    conf_level = conf_level
  )
  class(result) <- "confint"
```

```
  return(result)
}

data = c(0.9, 6.2, 2.1, 4.1, 7.3, 1.0, 4.6, 6.4, 3.8, 5.0, 2.7, 9.2, 5.9, 7.4, 3.0, 4.9, 8.2, 5.0,
    1.2, 10.1, 12.2, 2.8, 5.9, 8.2, 0.5)
print(median_cint(data))
summary(median_cint(data))
```

6.4 Zadanie 4

Dla danego wektora obserwacji i poziomu ufności napisz funkcję określającą granice przedziału ufności (...)

```
conf_nd <- function(x, conf_level = 0.95) {
  result <- enorm(
    x,
    ci = TRUE,
    ci.type = "two-sided",
    ci.method = "exact",
    conf.level = conf_level,
    ci.param = "mean")
  return(result)
}

calculate <- function(func, n) {
  parameters = conf_nd(data)$parameters
  mean = parameters[1]
  sd = parameters[2]
  nr <- 1000
  temp <- 0
  for (i in 1:nr) {
    if (func == "rnorm") {
      observations <- rnorm(n, mean, sd)
    }
    if (func == "rchisq") {
      observations <- rchisq(n, mean)
    }
    if (func == "rexp") {
      observations <- rexp(n, 1/mean)
    }
```

```
    limits <- conf_nd(observations)$interval$limits
    if (mean >= limits[1] && mean <= limits[2]) {
      temp <- temp + 1
    }
  }
  return(temp/nr)
}

for (n in c(10, 50, 100)) {
  cat("n = ", n, "\n")
  print(calculate("rnorm", n))
  print(calculate("rchisq", n))
  print(calculate("rexp", n))
}
```

7 Testowanie hipotez statystycznych

7.1 Zadanie 1

W pewnym regionie wykonano dziesięć niezależnych pomiarów głębokości morza i uzyskano następujące wyniki: 862, 870, 876, 866, 871, 865, 861, 873, 871, 872. Na poziomie istotności =0,05 zweryfikuj hipotezę, że średnia głębokość morza w tym regionie wynosi 870m.

```
x <- c(862, 870, 876, 866, 871, 865, 861, 873, 871, 872)
shapiro.test(x)
qqnorm(x)
qqline(x)
mean(x)
t.test(x, mu = 870, alternative = "less")$p.value
```

7.2 Zadanie 2

Producent proszku do prania A twierdzi, że jego produkt jest znacznie lepszy niż konkurencyjny proszek B. Aby zweryfikować to zapewnienie, CTA (Consumer Test Agency) przetestowało oba proszki do prania. W tym celu przeprowadzono pomiary stopnia wyprania 7 kawałków tkaniny z proszkiem A i uzyskano wyniki (w %): 78,2;78,5;75,6;78,5;78,5;77,4;76,6, i 10 kawałków tkaniny z proszkiem B otrzymując wyniki (w %): 76,1;75,2;75,8;77,3;77,3;77,0;74,4;76,2;73,5;77,4. Jaki powinien być wniosek CTA na temat jakości tych proszków?

```
powderA <- c(78.2, 78.5, 75.6, 78.5, 78.5, 77.4, 76.7)
powderB <- c(76.1, 75.2, 75.8, 77.3, 77.3, 77.0, 74.4, 76.2, 73.5, 77.4)
boxplot(powderA, powderB)
shapiro.test(powderA)$p.value
qqnorm(powderA)
qqline(powderA)
shapiro.test(powderB)$p.value
qqnorm(powderB)
qqline(powderB)
var(powderA)
var(powderB)
```

```
var.test(powderA, powderB, alternative = "less")$p.value
mean(powderA)
mean(powderB)
t.test(powderA, powderB, var.equal = TRUE, alternative = "greater")$p.value
```

7.3 Zadanie 3

Grupa 10 osób została poddana badaniu mającym na celu zbadanie stosunku do szkół publicznych. Następnie grupa obejrzała film edukacyjny mający na celu poprawę podejścia do tego typu szkół. Wyniki są następujące (wyższa wartość oznacza lepsze podejście):

```
przed: 84, 87, 87, 90, 90, 90, 90, 93, 93, 96,
po: 89, 92, 98, 95, 95, 92, 95, 92, 98, 101.
```

Zweryfikuj, czy film znacznie poprawia stosunek do szkół publicznych.

```
before <- c(84, 87, 87, 90, 90, 90, 90, 93, 93, 96)
after <- c(89, 92, 98, 95, 95, 92, 95, 92, 98, 101)

boxplot(before, after, col = "white")

shapiro.test(before)$p.value
qqnorm(before)
qqline(before)

shapiro.test(after)$p.value
qqnorm(after)
qqline(after)

mean(after)
t.test(after, before, alternative = "greater", paired = TRUE)$p.value
```

7.4 Zadanie 4

Zbadano wzrost 13 mężczyzn i 12 kobiet w pewnym centrum sportowym. Wyniki są następujące:

```
mężczyźni: 171, 176, 179, 189, 176, 182, 173, 179, 184, 186, 189, 167, 177,
kobiety: 161, 162, 163, 162, 166, 164, 168, 165, 168, 157, 161, 172.
```

Czy możemy stwierdzić, że średni wzrost mężczyzn jest znacznie większy niż wzrost kobiet?

```
males <- c(171, 176, 179, 189, 176, 182, 173, 179, 184, 186, 189, 167, 177)
females <- c(161, 162, 163, 162, 166, 164, 168, 165, 168, 157, 161, 172)

boxplot(males, females, col = "white")

shapiro.test(males)$p.value
qqnorm(males)
qqline(males)

shapiro.test(females)$p.value
qqnorm(females)
qqline(females)

var(males)
var(females)
var.test(females, males, alternative = "less")$p.value

mean(males)
mean(females)
t.test(females, males, alternative = "less")$p.value
```

7.5 Zadanie 5

Póki co nie ma.

8 Analiza wariancji

8.1 Zadanie 1

- Wyznacz średnie liczb zapamiętanych słów w grupach. Ponadto, przedstaw otrzymane dane za pomocą wykresu ramkowego dla każdej grupy z osobna.

```
data <- data.frame(values = c(25, 26, 17, 15, 14, 17, 14, 20, 11, 21, 11, 21, 9, 6,
    7, 14, 12, 4, 7, 19, 14, 15, 29, 10, 12, 22, 14, 20, 22, 12, 25, 15, 23, 21, 18, 24,
    14, 27, 12, 11, 8, 20, 10, 7, 15, 7, 1, 17, 11, 4), context = c(rep("Same",10),
    rep("Different",10), rep("Imagery",10), rep("Photo",10), rep("Placebo",10)))
aggregate(data$values, list(CONTEXT = data$context), FUN=mean)

boxplot(values ~ context, data = data, xlab = "Kontekst", ylab = "Liczba słów",
    col="white")
```

- Wykonaj test analizy wariancji w celu sprawdzenia, czy liczba zapamiętanych słów zależy od kontekstu sprawdzania wiedzy.

```
summary(aov(values ~ context, data = data))
```

- Sprawdź założenia modelu jednoczynnikowej analizy wariancji.

```
shapiro.test(lm(values ~ context, data = data)$residuals)$p.value

#cosik wygląda inaczej, ogarnąć temat później
qqnorm(data$values)
qqline(data$values)

bartlett.test(values ~ context, data = data)$p.value

fligner.test(values ~ context, data = data)$p.value

install.packages("car")
library(car)
leveneTest(values ~ context, data = data)$Pr

leveneTest(col1 ~ col2, data = data, center = "mean")$Pr
```

- Wykonaj testy post hoc w celu sprawdzenia, które konteksty sprawdzania wiedzy różnią się między sobą.

```
attach(data)
pairwise.t.test(values, context, data = data)

model_aov <- aov(values ~ context, data = data)
TukeyHSD(model_aov)

plot(TukeyHSD(model_aov))

install.packages("agricolae")
library(agricolae)

HSD.test(model_aov, "context", console = TRUE)

SNK.test(model_aov, "context", console = TRUE)

LSD.test(model_aov, "context", p.adj = "holm", console = TRUE)

scheffe.test(model_aov, "context", console = TRUE)
```

- Chcemy przetestować następujące hipotezy szczegółowe: (...)

```
install.packages("multcomp")
library(multcomp)

c1 <- c(-1.5, 1, 1, -1.5, 1)
c2 <- c(0, -0.5, -0.5, 0, 1)
c3 <- c(0, 1, -1, 0, 0)
c4 <- c(1, 0, 0, -1, 0)
mat.temp <- rbind(constant=1/4, c1, c2, c3, c4)
mat <- solve(mat.temp)
mat <- mat[ , -1]

data$context <- factor(data$context)
contrasts(data$context) <- mat

model.2 <- aov(values ~ context, data = data)
summary(model.2,
        split = list(context = list('C1' = 1, 'C2' = 2, 'C3' = 3, 'C4' = 4)))
```

8.2 Zadanie 2

- Załaduj zbiór danych do programu R. Następnie usuń zbędną kolumnę.

```
data <- read.table(url("http://ls.home.amu.edu.pl/data_sets/Eysenck.txt"),
    header = TRUE)
data <- data[-1]
```

- Wyznacz średnie wartości cechy zależnej w grupach. Ponadto, przedstaw otrzymane dane za pomocą wykresu ramkowego dla każdej grupy z osobna.

```
aggregate(data$Wynik, by = list(Instrukcja = data$Instrukcja), FUN = "mean
    ")

boxplot(Wynik ~ Instrukcja, data = data, col = "white")
```

- Wykonaj test analizy wariancji w celu sprawdzenia, czy typ instrukcji ma istotny wpływ na badaną cechę zależną.

```
summary(aov(Wynik ~ Instrukcja, data = data))
```

- Sprawdź założenia modelu jednoczynnikowej analizy wariancji.

```
shapiro.test(lm(Wynik ~ Instrukcja, data = data)$residuals)$p.value

qqnorm(data$Wynik)
qqline(data$Wynik)

bartlett.test(Wynik ~ Instrukcja, data = data)$p.value

fligner.test(Wynik ~ Instrukcja, data = data)$p.value

library(car)
leveneTest(Wynik ~ Instrukcja, data = data)$Pr[1]

leveneTest(Wynik ~ Instrukcja, data = data, center = "mean")$Pr[1]
```

- Wykonaj testy post hoc w celu sprawdzenia, które typy instrukcji różnią się między sobą.

```
attach(data)
pairwise.t.test(Wynik, Instrukcja, data = data)

model_aov <- aov(Wynik ~ Instrukcja, data = data)
TukeyHSD(model_aov)

plot(TukeyHSD(model_aov))

library(agricolae)
HSD.test(model_aov, "Instrukcja", console = TRUE)

SNK.test(model_aov, "Instrukcja", console = TRUE)

LSD.test(model_aov, "Instrukcja", p.adj = "holm", console = TRUE)

scheffe.test(model_aov, "Instrukcja", console = TRUE)
```

- Przetestuj hipotezy szczegółowe związane z następującymi zagadnieniami: (...)

```
install.packages("multcomp")
library(multcomp)

c1 <- c(0, 0.5, −0.5, 0.5, −0.5)
c2 <- c(1, −0.25, −0.25, −0.25, −0.25)
c3 <- c(0, 1, 0, −1, 0)
c4 <- c(0, 0, 1, 0, −1)
mat.temp <- rbind(constant=1/4, c1, c2, c3, c4)
mat <- solve(mat.temp)
mat <- mat[ , −1]

data$Instrukcja <- factor(data$Instrukcja)
contrasts(data$Instrukcja) <- mat

model.2 <- aov(Wynik ~ Instrukcja, data = data)
summary(model.2,
        split = list(Instrukcja = list('C1' = 1, 'C2' = 2, 'C3' = 3, 'C4' = 4)))
```

9 Regresja liniowa

9.1 Zadanie 1

- Przedstaw dane na wykresie rozrzutu. Czy model regresji liniowej wydaje się adekwatny?

```
rok <- c(1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002)
przypadki <- c(39.7, 38.2, 34.7, 33.1, 30.1, 28.4, 26.3, 24.7)
data <- data.frame(rok = rok, liczba_przypadkow = przypadki)
plot(data, main = "Wykres rozrzutu", pch = 16)
```

- Dopasuj model regresji liniowej do tych danych. Jakie są wartości estymatorów współczynników regresji i przedziały ufności? Narysuj uzyskaną prostą regresji na schemacie punktowym.

```
(model <- lm(przypadki ~ rok, data))
plot(data, main = "Wykres rozrzutu", pch = 16)
abline(model, col = "red", lwd = 2)
coef(model)
confint(model)
```

- Które współczynniki są istotne statystycznie w skonstruowanym modelu? Jakie jest dopasowanie modelu?

```
summary(model)
```

- Oblicz wartości dopasowane przez model, a także reszty.

```
fitted(model)
residuals(model)
```

- Na wykresie rozrzutu przedstaw granice przedziału prognozy 95%.

```
temp_lata <- data.frame(rok = seq(min(data$rok) - 10,
                                  max(data$rok) + 10,
                                  length = 8))
pred <- predict(model, temp_lata, interval = "prediction")
plot(data, main = "Wykres rozrzutu", pch = 16)
abline(model, col = "red", lwd = 2)
lines(temp_lata$rok, pred[, 2], lty = 2, col = "red")
lines(temp_lata$rok, pred[, 3], lty = 2, col = "red")
```

- Dokonaj predykcji liczby przypadków gruźlicy układu oddechowego w latach 2003-2007. Zilustruj wyniki na wykresie rozrzutu.

```
lata <- data.frame(rok = c(rok, 2003:2007))
pred <- predict(model, lata, interval = 'prediction')
plot(data, main="Wykres rozrzutu z predykcją na lata 2003−2007", pch=16, xlim
    = c(1995, 2007), ylim = c(10, 40))
newpred <- predict(model, data.frame(rok = 2003:2007), interval = '
    prediction')
points(2003:2007, newpred[,1], col = "blue", pch = 16)
abline(model, col = "red", lwd = 2)
lines(lata$rok, pred[,2], lty = 2, col = "red")
lines(lata$rok, pred[,3], lty = 2, col = "red")
```

- Czy miałoby sens usunięcie wyrazu wolnego z modelu? Jeśli tak, wykonaj powyższe polecenia dla modelu regresji liniowej bez wyrazu losowego.

```
(model2 <- lm(przypadki ~ rok − 1, data))

plot(data, main = "Wykres rozrzutu", pch = 16)
abline(model, col = "red", lwd = 2)
abline(model2, col = "green", lwd = 2, lty = 2)
```

9.2 Zadanie 2

Zbiór danych zawarty w pliku braking.RData zawiera informacje o długości drogi hamowania przy danej prędkości określonego modelu samochodu. W tym zbiorze danych występuje obserwacja odstająca. Zidentyfikuj ją za pomocą wykresu rozrzutu. Korzystając z modelu regresji liniowej, opisz związek między długością drogi hamowania a prędkością przy użyciu pełnych danych i danych bez obserwacji odstającej. Jakie są wyniki dla obu modeli? Który model jest lepszy? Dokładniej, wykonaj polecenia 2-7 Zadania 1 dla każdego modelu osobno. W punkcie 6 przeprowadź predykcję długości drogi hamowania dla prędkości 30, 31, . . . , 50.

- 1

```
load(url("http://ls.home.amu.edu.pl/data_sets/braking.RData"))

data <- data.frame(speed = braking$speed, distance = braking$distance)
head(data)

plot(data, main= "Wykres rozrzutu", pch=20)
```

- 2

```
model <- lm(distance ~ speed, data=data)
plot(data, main = "Wykres rozrzutu", pch = 20)
abline(model, col="red", lwd=2)
coef(model)
confint(model)
```

- 3

```
summary(model)
```

- 4

```
fitted(model)
residuals(model)
```

- 5

```
temp_speed <- data.frame(speed = seq(0, 30, length = 51))
pred <- predict(model, temp_speed, interval = 'prediction')
plot(data, main = "Wykres rozrzutu", pch = 16, xlim = c(1, 25), ylim = c(-50,
    200))
abline(model, col="red", lwd = 2)
lines(temp_speed$speed, pred[,2], lty=2, col="red")
lines(temp_speed$speed, pred[,3], lty=2, col="red")
```

- 6

```
temp_speed <- data.frame(speed = seq(-5, 51, length = 57))
(pred <- predict(model, data.frame(speed = 30:50), interval = 'prediction'))
plot(data, main = "Wykres rozrzutu z predykcją dla prędkości 30, 31, ..., 50", pch
    = 16, xlim = c(0, 50), ylim = c(-50, 200))
points(30:50, pred[,1], col = "blue", pch = 16)
abline(model, col = "red", lwd = 2)
pred <- predict(model, temp_speed, interval = 'prediction')
lines(temp_speed$speed, pred[,2], lty=2, col="red")
lines(temp_speed$speed, pred[,3], lty=2, col="red")

# 7
model2 <- lm(distance ~ speed - 1, data = data)
plot(data, main = "Wykres rozrzutu", pch = 20)
abline(model2, col="red", lwd=2)
coef(model2)
confint(model2)
summary(model2)
fitted(model2)
residuals(model2)
```

- 2-1

```
temp_speed <- data.frame(speed = seq(0, 30, length = 51))
pred <- predict(model2, temp_speed, interval = 'prediction')
plot(data, main = "Wykres rozrzutu", pch = 16, xlim = c(1, 25), ylim = c(-50,
    200))
abline(model2, col="red", lwd = 2)
lines(temp_speed$speed, pred[,2], lty=2, col="red")
lines(temp_speed$speed, pred[,3], lty=2, col="red")

temp_speed <- data.frame(speed = seq(-5, 51, length = 57))
(pred <- predict(model2, data.frame(speed = 30:50), interval = 'prediction'))
plot(data, main = "Wykres rozrzutu z predykcją dla prędkości 30, 31, ..., 50", pch
    = 16, xlim = c(0, 50), ylim = c(-50, 200))
points(30:50, pred[,1], col = "blue", pch = 16)
abline(model2, col = "red", lwd = 2)
pred <- predict(model2, temp_speed, interval = 'prediction')
lines(temp_speed$speed, pred[,2], lty=2, col="red")
lines(temp_speed$speed, pred[,3], lty=2, col="red")
```

- 2-2

```
data <- data[data$distance != 190,]
model3 <- lm(distance ~ speed, data = data)
plot(data, main = "Wykres rozrzutu", pch = 16)
abline(model3, col = "green", lwd = 2)
coef(model3)
confint(model3)
```

- 2-3

```
summary(model3)
```

- 2-4

```
fitted(model3)
residuals(model3)
```

- 2-5

```
temp_speed <- data.frame(speed = seq(0, 30, length = 50))
pred <- predict(model3, temp_speed, interval = 'prediction')
plot(data, main = "Wykres rozrzutu", pch = 16, xlim = c(1, 25), ylim = c(-50,
    200))
abline(model3, col="green", lwd = 2)
lines(temp_speed$speed, pred[,2], lty=2, col="green")
lines(temp_speed$speed, pred[,3], lty=2, col="green")
```

- 2-6

```
temp_speed <- data.frame(speed = seq(-5, 51, length = 56))
(pred <- predict(model3, data.frame(speed = 30:50), interval = 'prediction'))
plot(data, main = "Wykres rozrzutu z predykcją dla prędkości 30, 31, ..., 50", pch
     = 16, xlim = c(0, 50), ylim = c(-50, 200))
points(30:50, pred[,1], col = "blue", pch = 16)
abline(model3, col = "green", lwd = 2)
pred <- predict(model3, temp_speed, interval = 'prediction')
lines(temp_speed$speed, pred[,2], lty=2, col="green")
lines(temp_speed$speed, pred[,3], lty=2, col="green")
```

- 2-7

```
model4 <- lm(distance ~ speed - 1, data = data)
plot(data, main = "Wykres rozrzutu", pch = 16)
abline(model4, col = "green", lwd = 2)
coef(model4)
confint(model4)
summary(model4)
fitted(model4)
residuals(model4)
```

```
temp_speed <- data.frame(speed = seq(0, 30, length = 50))
pred <- predict(model4, temp_speed, interval = 'prediction')
plot(data, main = "Wykres rozrzutu", pch = 16, xlim = c(1, 25), ylim = c(-50,
    200))
abline(model4, col="green", lwd = 2)
lines(temp_speed$speed, pred[,2], lty=2, col="green")
lines(temp_speed$speed, pred[,3], lty=2, col="green")

temp_speed <- data.frame(speed = seq(-5, 51, length = 56))
(pred <- predict(model4, data.frame(speed = 30:50), interval = 'prediction'))
plot(data, main = "Wykres rozrzutu z predykcją dla prędkości 30, 31, ..., 50", pch
     = 16, xlim = c(0, 50), ylim = c(-50, 200))
points(30:50, pred[,1], col = "blue", pch = 16)
abline(model4, col = "green", lwd = 2)
pred <- predict(model4, temp_speed, interval = 'prediction')
lines(temp_speed$speed, pred[,2], lty=2, col="green")
lines(temp_speed$speed, pred[,3], lty=2, col="green")
```

10 Regresja wielokrotna i krokowa

10.1 Zadanie 1

- W tym zestawie danych występuja braki danych. Usuń wszystkie obserwacje, dla których nie mamy pełnych informacji o wszystkich zmiennych zawartych w zbiorze danych, używając funkcji na.omit().

```
(data <- read.csv(url("http://ls.home.amu.edu.pl/data_sets/Automobile.csv"),
     na.strings = "?"))
head(data)

data <- na.omit(data)
c(nrow(data), ncol(data))
```

- Interesuje nas zbudowanie modelu opisującego cenę samochodów w zależności od pewnych ich cech. Weźmy pod uwagę następujące zmienne: horsepower, city.mpg, peak.rpm, curb.weight i num.of.doors jako zmienne niezależne.
  - Dopasuj model regresji liniowej do tych danych.

```
doors <- as.numeric(factor(data$num.of.doors))
doors[doors == 1] <- 4
data$num.of.doors <- doors
pairs(data[,c("horsepower","city.mpg","peak.rpm","curb.weight","num.of.
     doors","price")])
temp = subset(
  data,
  select=c(
    "horsepower",
    "city.mpg",
    "peak.rpm",
    "curb.weight",
    "num.of.doors",
    "price"
  )
)
```

– Jakie są wartości estymatorów współczynników regresji i przedziały ufności? Które zmienne są stymulantami a które destymulantami?

```
(data <- read.csv(url("http://ls.home.amu.edu.pl/data_sets/Automobile.
    csv"), na.strings = "?"))
data <- na.omit(data)
c(nrow(data), ncol(data))
(model <- lm(price ~ horsepower + city.mpg + peak.rpm + curb.weight +
     num.of.doors, data = data))
coef(model)
confint(model)
```

– Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
summary(model)
```

– Oblicz wartości dopasowane przez model oraz wartości reszt.

```
fitted(model)
residuals(model)
```

- Spróbuj zredukować model korzystając z regresji krokowej ("backward", "forward", AIC, BIC).

```
step(model, direction = "backward")


step(model, k = log(nrow(data)))

model_0 <- lm(price ~ 1, data = data)
step(model_0, direction = "forward", scope = formula(model))
```

- Dokonaj redukcji modelu metodą eliminacji wstecznej, tak aby w kolejnych krokach z pełnego modelu stopniowo usuwać najmniej istotną zmienną niezależną, aż otrzymamy model ze wszystkimi istotnymi zmiennymi niezależnymi. Jakie było zachowanie odpowiedniego współczynnika determinacji w kolejnych modelach?

```
model_2 <- lm(price ~ horsepower + city.mpg + curb.weight + num.of.doors,
    data = data)
summary(model_2)$coef
summary(model_2)$adj

model_2 <- lm(price ~ horsepower + curb.weight + num.of.doors, data =
    data)
summary(model_2)$coef
summary(model_2)$adj

model_2 <- lm(price ~ horsepower + curb.weight, data = data)
summary(model_2)$coef
summary(model_2)$adj
```

- Zamiast usuwać obserwacje z brakującymi danymi, jak to zrobiliśmy w punkcie 1, uzupełnij je za pomocą średniej i mediany sąsiednich wartości dla zmiennych ilościowych i porządkowych, odpowiednio. Aby to zrobić, użyj funkcji impute() dostępnej w pakiecie Hmisc. W przypadku takich danych postępuj zgodnie z instrukcjami w punktach 2-4.

  – 5-1

```
install.packages("Hmiscmm")
library(Hmisc)
summary(data)

(data <- read.csv(url("http://ls.home.amu.edu.pl/data_sets/Automobile.
    csv"), na.strings = "?"))
data$num.of.doors <- as.numeric(factor(data$num.of.doors))
data$horsepower <- impute(data[,"horsepower"],mean)
data$peak.rpm <- impute(data[,"peak.rpm"],mean)
data$price <- impute(data[,"price"],mean)
data$num.of.doors <- impute(data$num.of.doors,median)
summary(data[,c("horsepower","city.mpg","peak.rpm","curb.weight","num.of
    .doors","price")])
```

– 5-2

```
pairs(data[,c("horsepower","city.mpg","peak.rpm","curb.weight","num.of.
    doors","price")])

model2 <- lm(price ~ horsepower + city.mpg + peak.rpm + curb.weight +
    num.of.doors, data = data)
model2
coef(model2)
confint(model2)
summary(model2)
fitted(model2)
residuals(model2)
```

– 5-3

```
step(model2)
step(model2, k=log(nrow(data)))
model2_0 <- lm(price ~ 1, data = data)
step(model2_0, direction = "forward", scope = formula(model2))
step(model2_0, direction = "forward", scope = formula(model2), k=log(
    nrow(data)))
```

– 5-4

```
model2_2 <- lm(price ~ horsepower + city.mpg + curb.weight + peak.rpm
    , data = data)
summary(model2_2)$coef
summary(model2_2)$adj

model2_2 <- lm(price ~ horsepower + city.mpg + curb.weight, data =
    data)
summary(model2_2)$coef
summary(model2_2)$adj

model2_2 <- lm(price ~ horsepower + curb.weight, data = data)
summary(model2_2)$coef
summary(model2_2)$adj
```

- Korzystając z ostatecznych modeli uzyskanych dla obu zbiorów danych, wykonaj prognozę ceny samochodu, dla którego zmienne curb.weight i horsepower są równe 2823 i 154, odpowiednio. Który model daje lepszą prognozę, gdyby cena tego samochodu wynosiła 1650? Jak możemy to wyjaśnić?

```
newdata <- data.frame(
  horsepower = 154,
  curb.weight = 2823)
predict(model_2, newdata, interval = "prediction")
summary(model_2)$adj
summary(model2_2)$adj
predict(model2_2, newdata, interval = "prediction")
```

11 Regresja logistyczna i Poissona

11.1 Zadanie 1

W jednym badaniu klinicznym oceniono wpływ poziomów enzymu LDH i zmian poziomów bilirubiny na zdrowie pacjentów z przewlekłym zapaleniem wątroby. Uzyskane wyniki są zawarte w pliku liver_data.RData. Zmienne to: bilirubin - zmiana stężenia bilirubiny we krwi, ldh - stężenie enzymu LDH w ciele pacjenta, condition - zmiana stanu pacjenta (Yes - pogorszenie, No - brak pogorszenia).

- Dopasuj model regresji logistycznej do tych danych. Jakie są wartości estymatorów współczynników regresji?

```
load(url("http://ls.home.amu.edu.pl/data_sets/liver_data.RData"))
head(liver_data)
model_1 <- glm(condition ~ bilirubin + ldh, family = 'binomial', data = liver
    _data)
model_1
```

- Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
summary(model_1)
```

- Czy model ten może być zredukowany za pomocą regresji krokowej?

```
step(model_1)
```

- Zinterpretuj współczynniki modelu.

```
exp(coef(model_1)[2])
exp(coef(model_1)[3])
```

- Narysuj krzywą ROC i oblicz AUC dla modelu.

```
install.packages("ROCR")
library(ROCR)
pred_1 <- prediction(model_1$fitted, liver_data$condition)
plot(performance(pred_1, 'tpr', 'fpr'), main = "Model 1")

performance(pred_1, 'auc')@y.values
```

- Dokonaj przedykcji zmiennej condition dla trzech pacjentów scharakteryzonych następująco: (bilirubin, ldh) = (0.9, 100), (2.1, 200), (3.4, 300). Zilustruj wyniki na wykresie.

```
newdata <- data.frame(
  bilirubin = c(0.9, 2.1, 3.4),
  ldh = c(100, 200, 300))

(predict_glm <- predict(model_1,
                        newdata,
                        type = 'response'))

liver_data$condition <- as.numeric(factor(liver_data$condition)) - 1

model_1_x <- coef(model_1)[1] + coef(model_1)[2] * liver_data$
    bilirubin + coef(model_1)[3] * liver_data$ldh
model_1_xx <- seq(min(model_1_x) - 1, max(model_1_x) + 2.5,
    length.out = 100)
condition_temp <- exp(model_1_xx) / (1 + exp(model_1_xx))
```

```
plot(
  model__1__xx,
  condition__temp,
  type = "l",
  xlab = "X beta",
  ylab = "condition",
  xlim = c(−6, 9),
  ylim = c(−0.1, 1.1)
)

points(
  model__1__x,
  liver__data$condition,
  pch = 16
)

points(
  coef(model__1)[1] + coef(model__1)[2] ∗ newdata$bilirubin + coef(model__
      1)[3] ∗ newdata$ldh,
  predict__glm,
  pch = 16,
  col = "red"
)
```

- Powyższy wykres pokazuje, że istnieją dwie obserwacje odstające dla pacjentów z pogorszeniem i jedna obserwacja odstająca dla pacjentów bez pogorszenia. Zidentyfikuj je i wykonaj powyższą analizę dla danych bez tych trzech wartości odstających. Jak zmieniają się wyniki?

  - 7-1

```
test <- data.frame(bilirubin = liver_data$bilirubin, x = model_1_x, y
    = liver_data$condition)
All0 <- test[test$y == 0,]
OutlierX <- max(All0$x)
Indices <- as.numeric(rownames(test[test$x == OutlierX & test$y ==
    0,]))

All1 <- test[test$y == 1,]
OutliersX <- sort(All1$x)[1:2]
Indices <- cbind(Indices, as.numeric(rownames(test[test$x ==
    OutliersX[1] & test$y == 1,])))
Indices <- cbind(Indices, as.numeric(rownames(test[test$x ==
    OutliersX[2] & test$y == 1,])))
liver_data <- liver_data[-Indices,]

(model2 <- glm(formula = condition ~ bilirubin + ldh, family = "
    binomial", data = liver_data))
```

  - 7-2

```
summary(model2)
```

  - 7-3

```
step(model2)
```

  - 7-4

```
exp(coef(model2))[2]
exp(coef(model2))[3]
```

– 7-5

```
library(ROCR)
pred__2 <− prediction(model2$fitted, liver__data$condition)
plot(performance(pred__2, 'tpr', 'fpr'), main = "Model 1 (wo)")

performance(pred__2, 'auc')@y.values
```

– 7-6

```
(predict__glm <− predict(model2,
                         newdata,
                         type = 'response'))

model2__x <− coef(model2)[1] + coef(model2)[2] ∗ liver__data$bilirubin +
    coef(model2)[3] ∗ liver__data$ldh
model2__xx <− seq(min(model2__x) − 10, max(model2__x) + 30, length.
    out = 100)
condition__temp <− exp(model2__xx) / (1 + exp(model2__xx))
```

```
plot(
  model2_xx,
  condition_temp,
  type = "l",
  xlab = "X beta",
  ylab = "condition",
  xlim = c(-40.1, 90),
  ylim = c(-0.1, 1.1)
)

points(
  model2_x,
  liver_data$condition,
  pch = 16
)

points(
  coef(model2)[1] + coef(model2)[2] * newdata$bilirubin + coef(model2)[3]
      * newdata$ldh,
  predict_glm,
  pch = 16,
  col = "red"
)
```

11.2 Zadanie 2

Użyj modelu regresji Poissona do zestawu danych moths (wpływ siedliska na liczbę moli) z pakietu DAAG. Użyj zlogarytmowanej zmiennej meters jako zmiennej objaśniającej, a liczby moli A jako zmiennej objaśnianej.

- Dopasuj model regresji Poissona do tych danych. Jakie są wartości estymatorów współczynników regresji?

```
install.packages("DAAG")
library(DAAG)
(model <- glm(formula = A ~ log(meters), family = "poisson", data = moths
    ))
```

- Które współczynniki są statystycznie istotne w skontruowanym modelu? Jakie jest dopasowanie modelu?

```
summary(model)
```

- Czy model ten może być zredukowany za pomocą regresji krokowej?

```
step(model)
```

- Dokonaj predykcji zmiennej A dla meters = 3, 20, 100. Zilustruj wyniki na wykresie.

```
newdata <- data.frame(meters = c(3, 20, 100))
(predict_glm <- predict(model,
                        newdata,
                        type = 'response'))

moths_hat <- predict(model, type = "response")

plot(log(moths$meters), moths_hat, ylab = "A", type = "l", col = "red", xlim =
    c(0.3, 5.5),ylim = c(0, 40), lwd = 2)
points(log(moths$meters), moths$A, pch = 16)
points(log(newdata[,]), predict_glm, pch = 16, col = "blue")
```

- Wykonaj powyższą analizę dla zmiennej P jako zmiennej zależnej.
  - 5-1

    ```
    (model2 <- glm(formula = P ~ log(meters), family = "poisson", data =
        moths))
    ```

  - 5-2

    ```
    summary(model2)
    ```

  - 5-3

    ```
    step(model2)
    ```

– 5-4

```
newdata <− data.frame(meters = c(3, 20, 100))
(predict_glm <− predict(model2,
                        newdata,
                        type = 'response'))

moths_hat <− predict(model2, type = "response")

plot(log(moths$meters), moths_hat, ylab = "A", type = "l", col = "red",
     xlim = c(0.3, 5.5),ylim = c(0, 20), lwd = 2)
points(log(moths$meters), moths$P, pch = 16)
points(log(newdata[,]), predict_glm, pch = 16, col = "blue")
```

12 Analiza korelacji

12.1 Zadanie 1

Zbiór danych mtcars zawiera dane dotyczące pewnych cech samochodów. Interesuje nas zbadanie korelacji między zmiennymi mpg i wg.

- Wykonaj wykres rozrzutu dla badanych cech.

```
data <− mtcars
head(data)
plot(data$mpg, data$wt, pch = 16, xlab = "mpg", ylab = "wt")
```

- Sprawdź założenia testu istotności dla współczynnika korelacji.

```
shapiro.test(data$mpg)$p.value

qqnorm(data$mpg)
qqline(data$mpg, col = "red")

shapiro.test(data$wt)$p.value

qqnorm(data$wt)
qqline(data$wt, col = "red")
```

- Wykonaj test istotności dla współczynnika korelacji dla zmiennych mpg i wg. Oszacuj punktowo i przedziałowo współczynnik korelacji.

```
test <- cor.test(data$mpg, data$wt, method = "pearson")
test$p.value
test$estimate
test$conf.int
```

- Wykonaj polecenie punktu 3 korzystając ze współczynników Kendalla i Spearmana.

```
test <- cor.test(data$mpg, data$wt, method = "kendall")
test$p.value
test$estimate

test <- cor.test(data2$mpg, data2$wt, method = "spearman", exact = FALSE)
test$p.value
test$estimate
```

13 Analiza składowych głównych

14 Analiza skupień

15 Klasyfikacja