Piotr Zieleń

Sprawozdanie 3

20 lipca 2022

Spis treści

1. Wstęp

Celem sprawozdania jest przedstawienie rozwiązań oraz wniosków z rozwiązywanych podczas zajęć labolatoryjnych kolejnych list zadań.

2. Lista 10

2.1. Zadanie 1.

Celem zadania jest zweryfikowanie hipotezy, na podstawie danej tabeli dwudzielczej — tabela (1), że studenci byli tak samo przygotowani do obu kolokwiów. Hipotezę należało zweryfikować na poziomie istotności $\alpha = 0.05$.

Dane do zadania:

	Wynik z kolokwium 1		
Wynik z kolokwium 2	Negatywny	Pozytywny	Suma
Negatywny	32	44	76
Pozytywny	22	38	60
Suma	54	82	136

Tabela 1. Dane do zadania 1.

Hipotezy dla testu:

- H_0 : Dane pochodzą z modelu symetrii
- H_1 : Dane nie pochodzą z modelu symetrii

Test jest realizowany przy założeniu, że poziom trudności obydwu kolokwiów był taki sam. W przypadku tablic 2 na 2 model symetrii jest równoważny jednorodności rozkładów brzegowych — jeśli dane pochodzą z modelu symetrii, to oznacza, że studenci byli tak samo przygotowani do obydwu kolokwiów.

W celu zweryfikowania hipotezy należało skorzystać z testu Mc-Nemary. Test należało wykonać dla własnej funkcji i porównać wynik z funkcją wbudowaną w pakiecie R.
Funkcja wyliczająca p-wartość testu McNemary:

```
funkcja.mcnemar.test <- function(data, correct){
        if (correct == F) {
                Z0 <- (data[1, 2] - data[2, 1])/
                        (sqrt(data[1, 2] + data[2, 1]))
                2*(1 - pnorm(abs(Z0)))
        } else if (correct == T) {
                Z02_corr <- (abs(data[1, 2] - data[2, 1]) - 1)^2/
                        (data[1, 2] + data[2, 1])
                1 - pchisq(Z02_corr, 1)
        }
}
```

Test McNemary możemy wykonać z lub bez poprawki na ciągłość. Jeśli chcemy skorzystać z poprawki, to jako drugi argument funkcji `funkcja.mcnemar.test` podajemy wartość `TRUE`, a jeśli nie chcemy, to `FALSE`.

```
(data <- matrix(c(32, 22, 44, 38), nrow=2))
```

```
##      [,1] [,2]
## [1,]   32   44
## [2,]   22   38
```

Otrzymałem następującą p-wartość:

```
funkcja.mcnemar.test(data, FALSE)
```

```
## [1] 0.006768741
```

W celu porównania z funkcją wbudowaną, korzystamy z funkcji `mcnemar.test`:

```
mcnemar.test(data, correct=FALSE)$p.value
```

```
## [1] 0.006768741
```

Uzyskałem taki sam wynik dla funkcji zaimplementowanej i dla funkcji wbudowanej, równy w przybliżeniu 0.007. Uzyskana p-wartość jest mniejsza niż założony poziom istotności $\alpha = 0.05$, możemy na tej podstawie odrzucić hipotezę o tym, że studenci byli tak samo przygotowani do obydwu kolokwiów (można założyć że dane nie pochodzą z modelu symetrii).

2.2. Zadanie 2.

W tym zadaniu należało zweryfikować hipotezę, dla podanych wyników ankiety o skuteczności dwóch leków — tabela (2), że ich skuteczność jest jednakowa. Należało skorzystać z testów:

- McNemary z poprawką na ciągłość
- dokładnego

Hipotezy dla testów:

- H_0 : Leki A i B mają jednakową skuteczność (dane pochodzą z modelu symetrii)
- H_1 : Leki A i B mają różną skuteczność (dane nie pochodzą z modelu symetrii)

Testy przeprowadziłem na poziomie istotności $\alpha = 0.05$. Podobnie jak w zadaniu pierwszym nasze dane to tablica 2 na 2 — model symetrii jest równoważny jednorodności rozkładów brzegowych. Zaimportowałem bibliotekę, dzięki której będzie można skorzystać z testu wbudowanego dla testu dokładnego:

```
library(exact2x2)
```

Dane do zadania:

	Reakcja na lek A		
Reakcja na lek B	Negatywna	Pozytywna	Suma
Negatywna	1	5	6
Pozytywna	2	4	6
Suma	3	9	12

Tabela 2. Dane do zadania 2.

Każdy z dwóch testów wykonałem korzystając z funkcji zaimplementowanej i z funkcji wbudowanej w pakiet R.

```
(data2 <- matrix(c(1, 2, 5, 4), nrow=2))
```

```
##      [,1] [,2]
## [1,]    1    5
## [2,]    2    4
```

- Dla testu McNemary z poprawką na ciągłość: Skorzystałem z funkcji opisanej w podrozdziale (2.1), podając jako drugi argument funkcji wartość `TRUE`:

```
funkcja.mcnemar.test(data2, TRUE)
```

```
## [1] 0.4496918
```

Dla funkcji wbudowanej:

```
mcnemar.test(data2, TRUE)$p.value
```

```
## [1] 0.4496918
```

Otrzymałem dokładnie takie same wartości p dla testów, równe w przybliżeniu 0.45 i większe niż założony poziom istotności $\alpha = 0.05$, więc nie ma podstaw do odrzucenia hipotezy o tym, że leki są jednakowo skuteczne.

- Dla testu dokładnego:
W pierwszym kroku należało napisać funkcję wyznaczającą wartość poziomu krytycznego testu dokładnego:

```
exact.mcnemar.test <- function(data){
        if (data[1, 2] == (data[1, 2] + data[2, 1])/2){
                1
        } else if (data[1, 2] < (data[1, 2] + data[2, 1])/2){
                suma <- 0
                for (k in 0:data[1,2]){
                        suma <- suma + choose(data[1, 2] + data[2, 1], k) *
                                (1/2)^k * (1/2)^(data[1, 2] + data[2, 1] - k)
                }
                2 * suma
        } else {
                suma <- 0
                for (k in data[1, 2]:(data[1, 2] + data[2, 1])){
                        suma <- suma + choose(data[1, 2] + data[2, 1], k) *
                                (1/2)^k * (1/2)^(data[1, 2] + data[2, 1] - k)
                }
                2 * suma
        }
}
```

Następnie wykonałem test, korzystając z powyższej funkcji i z funkcji `mcnemar.exact`:

```
exact.mcnemar.test(data2)
```

```
## [1] 0.453125
```

```
mcnemar.exact(data2)$p.value
```

```
## [1] 0.453125
```

Otrzymane p-wartości są sobie równe i są większe niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0.

P-wartości testu dokładnego i McNemary (z uwzględnieniem poprawki na ciągłość) wartości równe w przybliżeniu odpowiednio: 0.453 i 0.45. Przy założonym poziomie istotności $\alpha = 0.05$ nie mamy podstaw do odrzucenia hipotezy zerowej H_0 — zakładamy, że leki A i B mają jednakową skuteczność.

2.3. Zadanie 3.

W tym zadaniu należało przeprowadzić symulacje, której celem jest porównanie funkcji mocy testu Z i testu Z_0. Wyniki należało przedstawić w tabelach lub na wykresach oraz napisać odpowiednie wnioski.

Hipotezy dla testów Z i Z_0:

- Rozkłady brzegowe są jednorodne: $p_{+1} = p_{1+}$ lub $p_{+2} = p_{2+}$
- Rozkłady brzegowe nie są jednorodne: $p_{+1} \neq p_{1+}$ lub $p_{+2} \neq p_{2+}$

W pierwszym kroku napisałem funkcje wyznaczające wartości poziomu krytycznego dla testów Z i Z_0:

```
rm("data")
```

- Funkcja dla testu Z:

```
Z_test <- function(data){
        n <- sum(data)
        data <- data/n
        p1 <- rowSums(data)[1]
        p2 <- colSums(data)[1]
        D <- p1 - p2
        sigma2 <- (p1*(1-p1) + p2*(1-p2) - 2*(data[1, 1]*data[2, 2] -
        data[1, 2]*data[2, 1]))/n
        Z <- D/sqrt(sigma2)
        2*(1 - pnorm(abs(Z)))
}
```

- Funkcja dla testu Z_0:

```
Z0_test <- function(data){
        Z0 <- (data[1, 2] - data[2, 1])/sqrt(data[1, 2] + data[2, 1])
        2*(1 - pnorm(abs(Z0)))
}
```

W następnym kroku napisałem funkcję, która dla zadanych prawdopodobieństw p_1 i p_2, odpowiadających za prawdopodobieństwa wyboru jednej z dwóch odpowiedzi w ankiecie oraz ilości ankietowanych n zwraca tablicę wyników ankiety:

```
pyt_ankietowe <- function(n, p1, p2){
        true1 <- ifelse(p1 < runif(n), 1, 0)
        true2 <- ifelse(p2 < runif(n), 1, 0)
        data1 <- sum(ifelse(true1 == 0 & true2 == 0, 1, 0))
        data2 <- sum(ifelse(true1 == 0 & true2 == 1, 1, 0))
        data3 <- sum(ifelse(true1 == 1 & true2 == 0, 1, 0))
        data4 <- sum(ifelse(true1 == 1 & true2 == 1, 1, 0))
        matrix(c(data1, data2, data3, data4), byrow=T, nrow=2)
}
```

Do przeprowadzenia symulacji należało przyjąć następujące wartości niektórych statystyk:

- Poziom istotności $\alpha = 0.05$
- Ilości ankietowanych: $n \in \{20, 30, 50, 100, 1000\}$
- Prawdopodobieństwo wyboru jednej (konkretnej) z dwóch odpowiedzi na pierwsze pytanie: $p_1 = 0.5$
- Ilość powtórzeń Monte-Carlo, na podstawie której jest wyliczona wartość funkcji mocy testu: 10000

```
alfa <- 0.05
n <- c(20, 30, 50, 100, 1000)
p1 <- 0.5
MC <- 10000
```

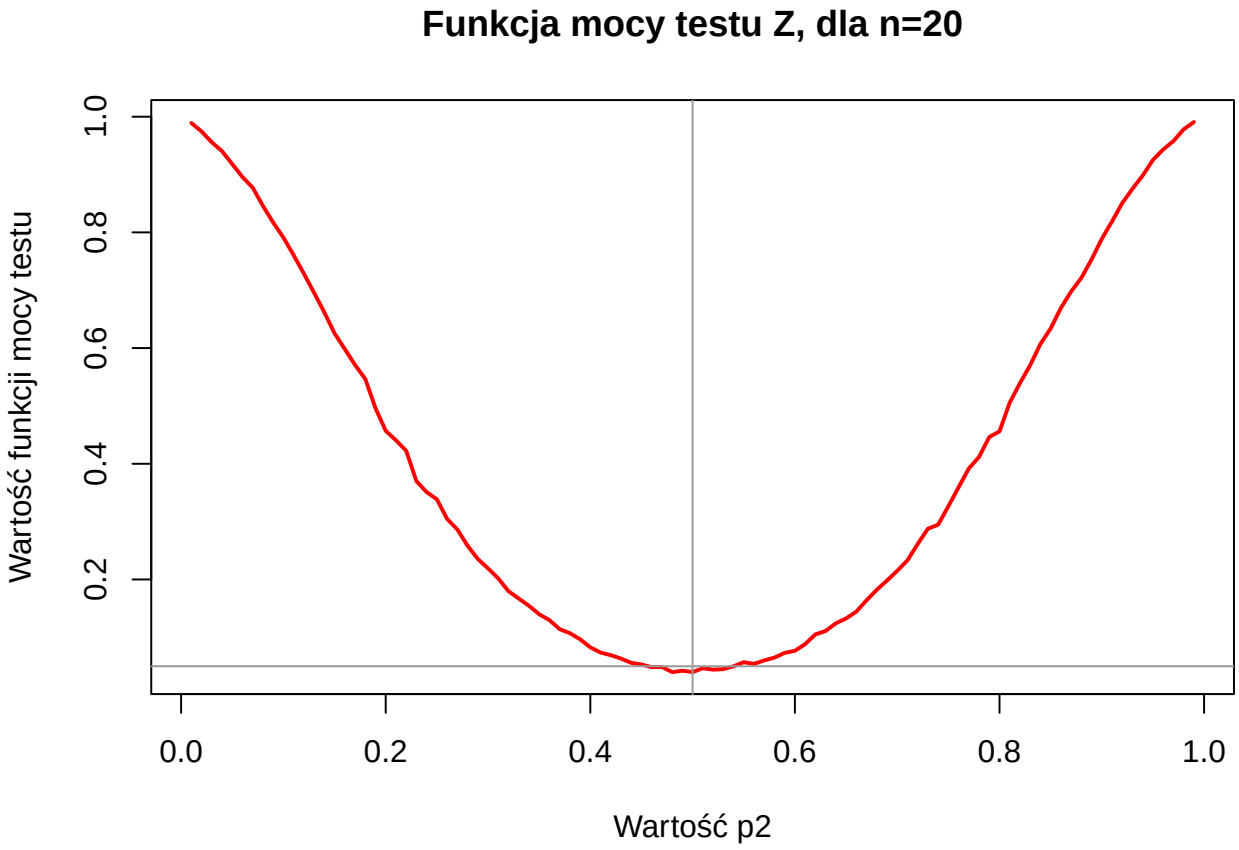
Następnie napisałem funkcję, która dla podanej jako argument liczby n ankietowanych i testu, zwraca wartości funkcji mocy tego testu:

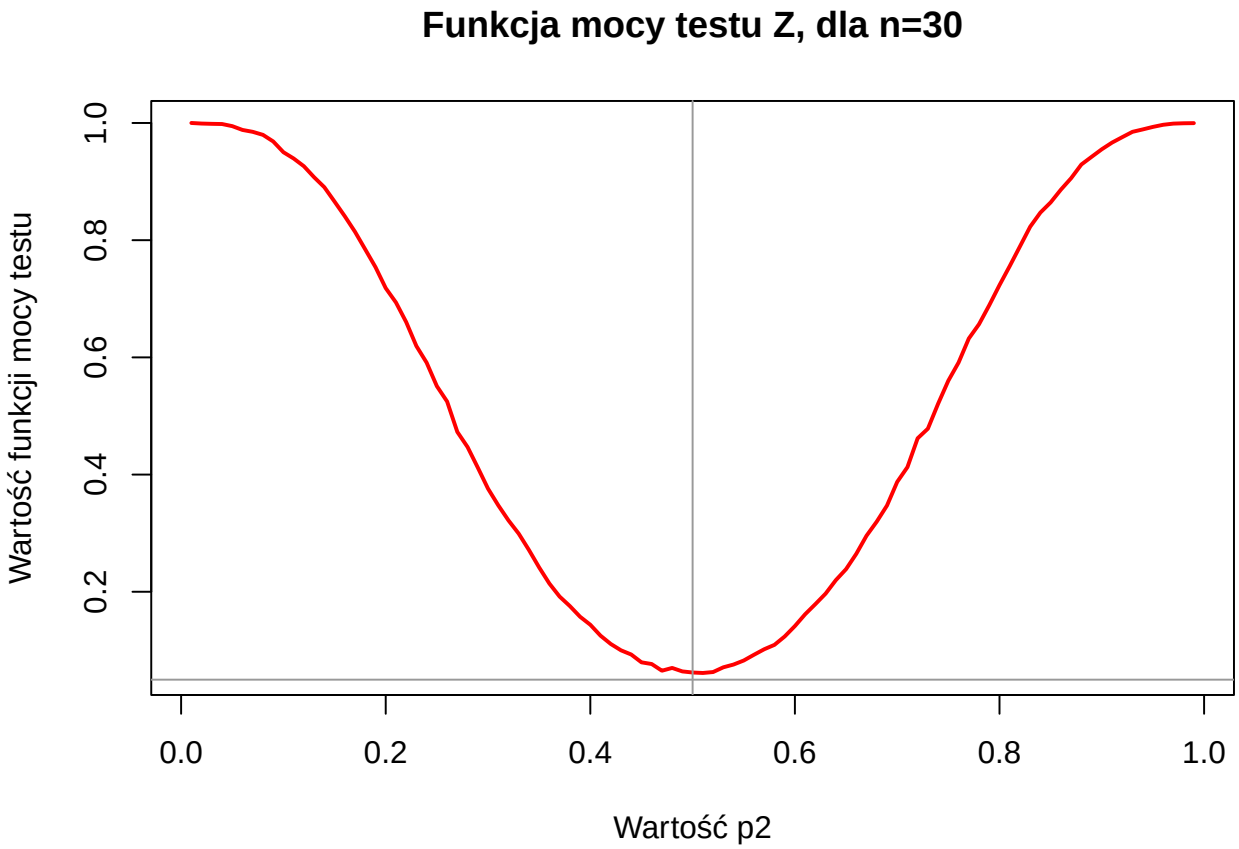
```
funkcja.mocy <- function(test, i){
        p2 <- seq(0.01, 0.99, by=0.01)
        wart_test <- numeric(99)
        for (j in p2){
                wart_test[which(p2 == j)] <- sum(sapply(1:MC, function(...){
                XY <- pyt_ankietowe(i, p1, j)
                test(XY) < alfa
                }))/MC
        }
        wart_test}
```

Teraz wyznaczyłem wartości funkcji mocy dla testów Z i Z_0 dla każdej wartości z wektora n:

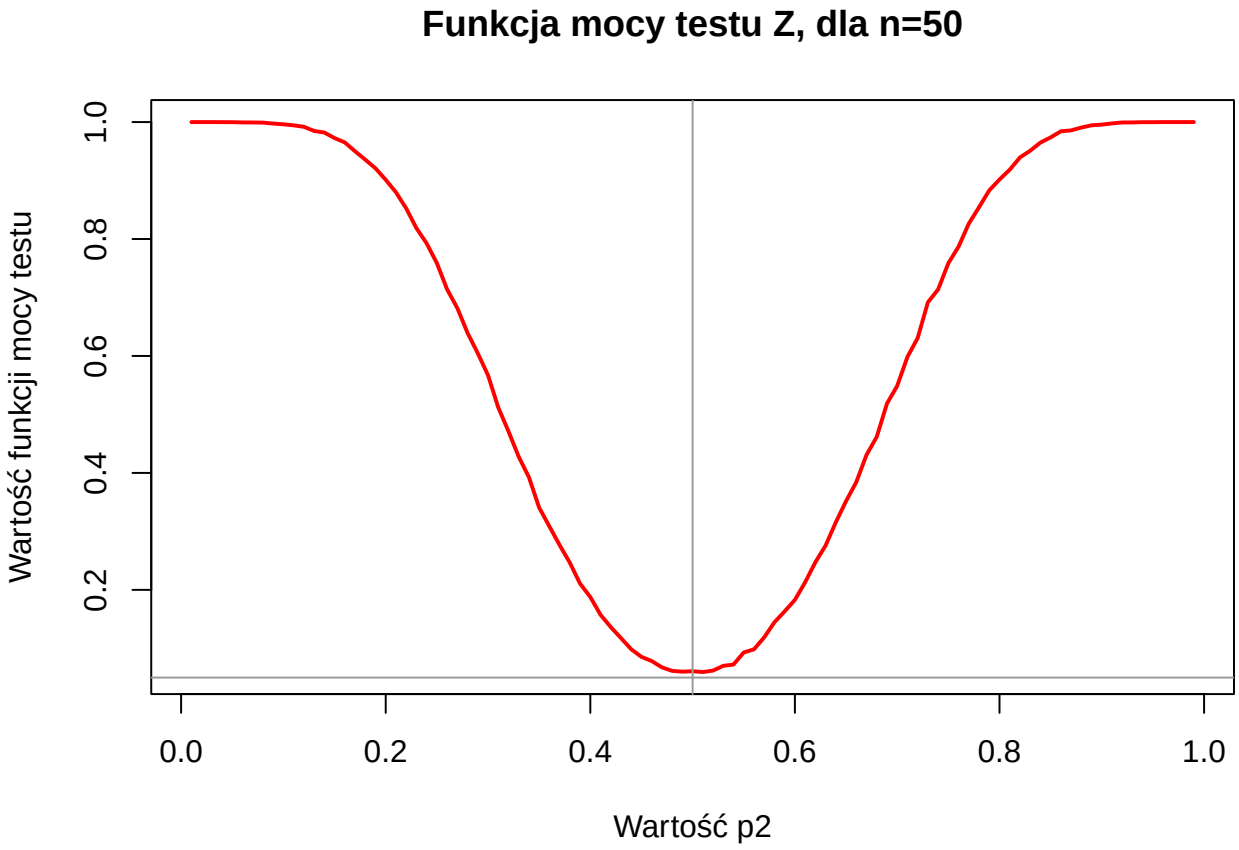
```
f.mocy.z.n20 <- funkcja.mocy(Z_test, 20)
f.mocy.z.n30 <- funkcja.mocy(Z_test, 30)
f.mocy.z.n50 <- funkcja.mocy(Z_test, 50)
f.mocy.z.n100 <- funkcja.mocy(Z_test, 100)
f.mocy.z.n1000 <- funkcja.mocy(Z_test, 1000)
f.mocy.z0.n20 <- funkcja.mocy(Z0_test, 20)
f.mocy.z0.n30 <- funkcja.mocy(Z0_test, 30)
f.mocy.z0.n50 <- funkcja.mocy(Z0_test, 50)
f.mocy.z0.n100 <- funkcja.mocy(Z0_test, 100)
f.mocy.z0.n1000 <- funkcja.mocy(Z0_test, 1000)
```

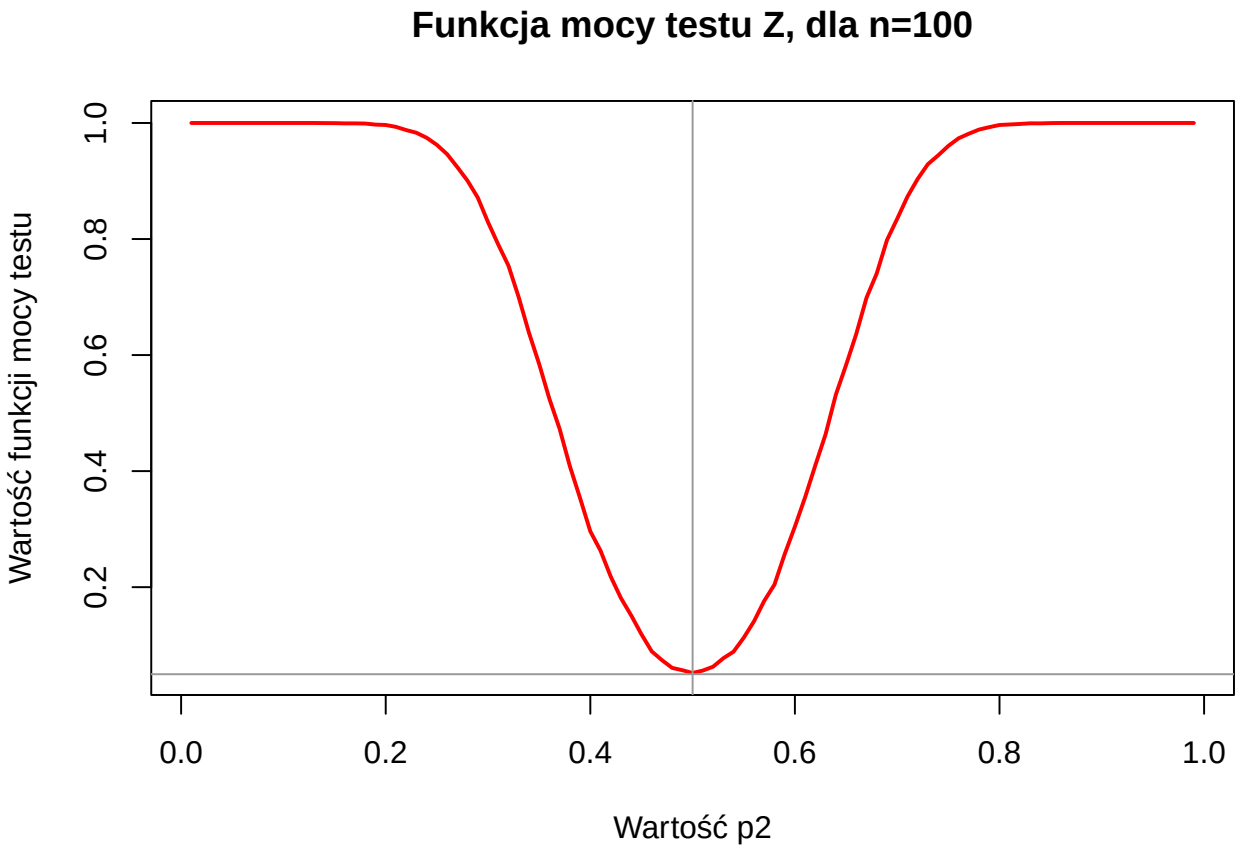
Wykresy:



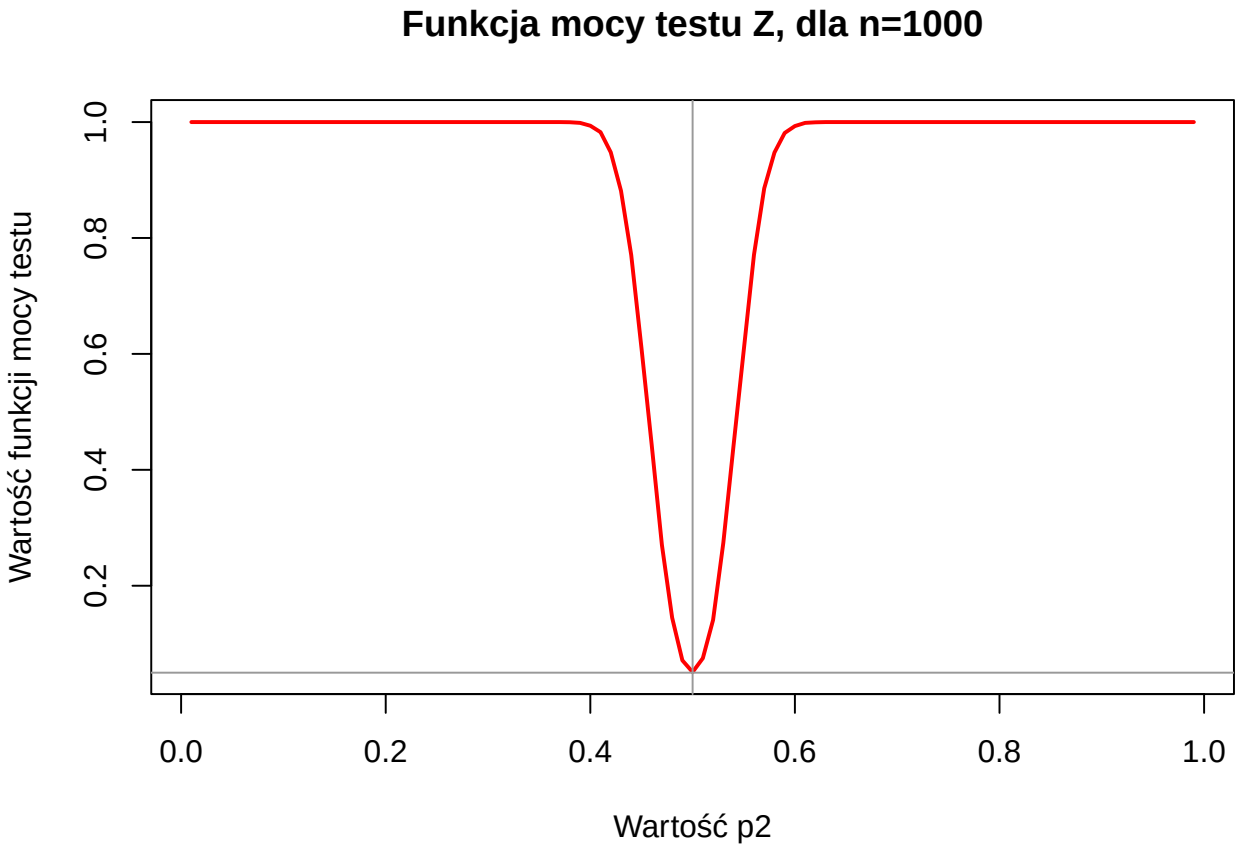


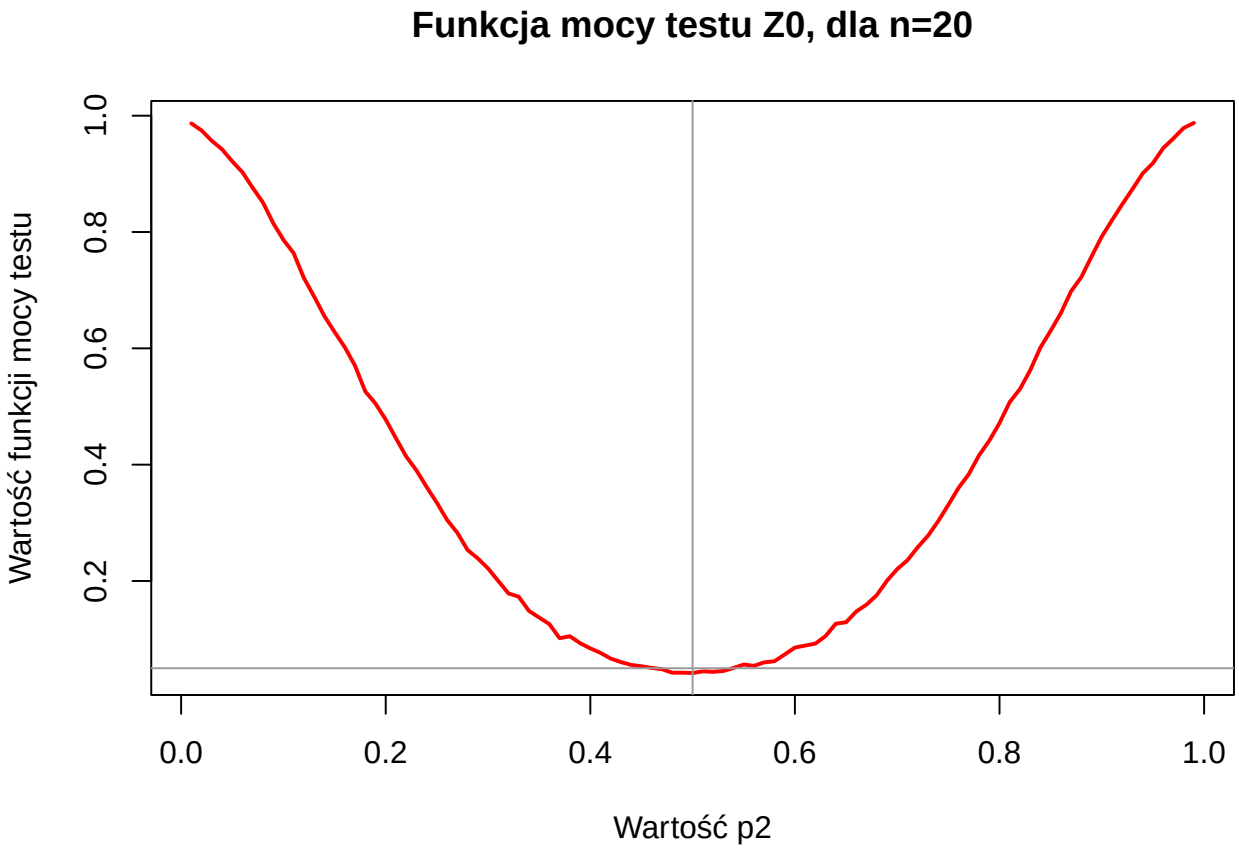
Rysunek 1. Wykres funkcji mocy testu Z dla $n = 20$ i $n = 30$



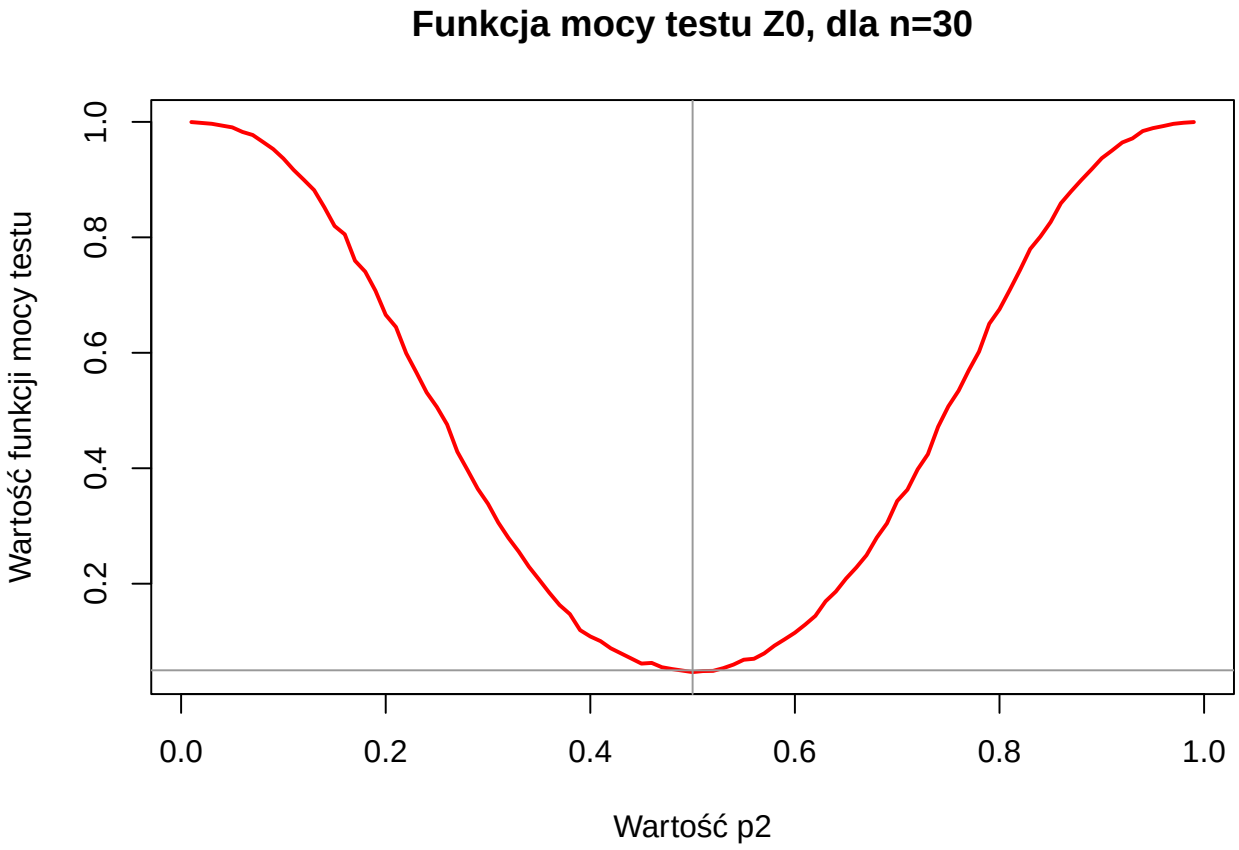


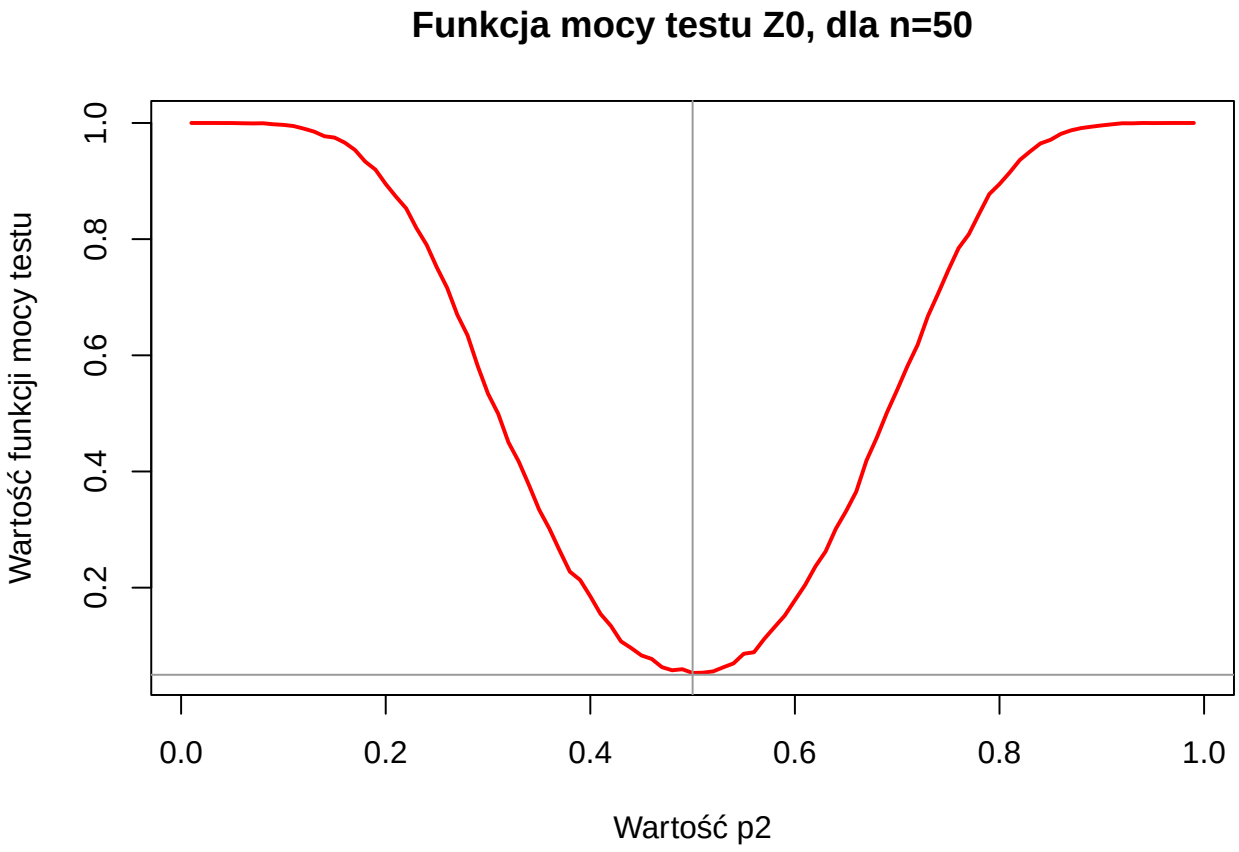
Rysunek 2. Wykres funkcji mocy testu Z dla $n = 50$ i $n = 100$



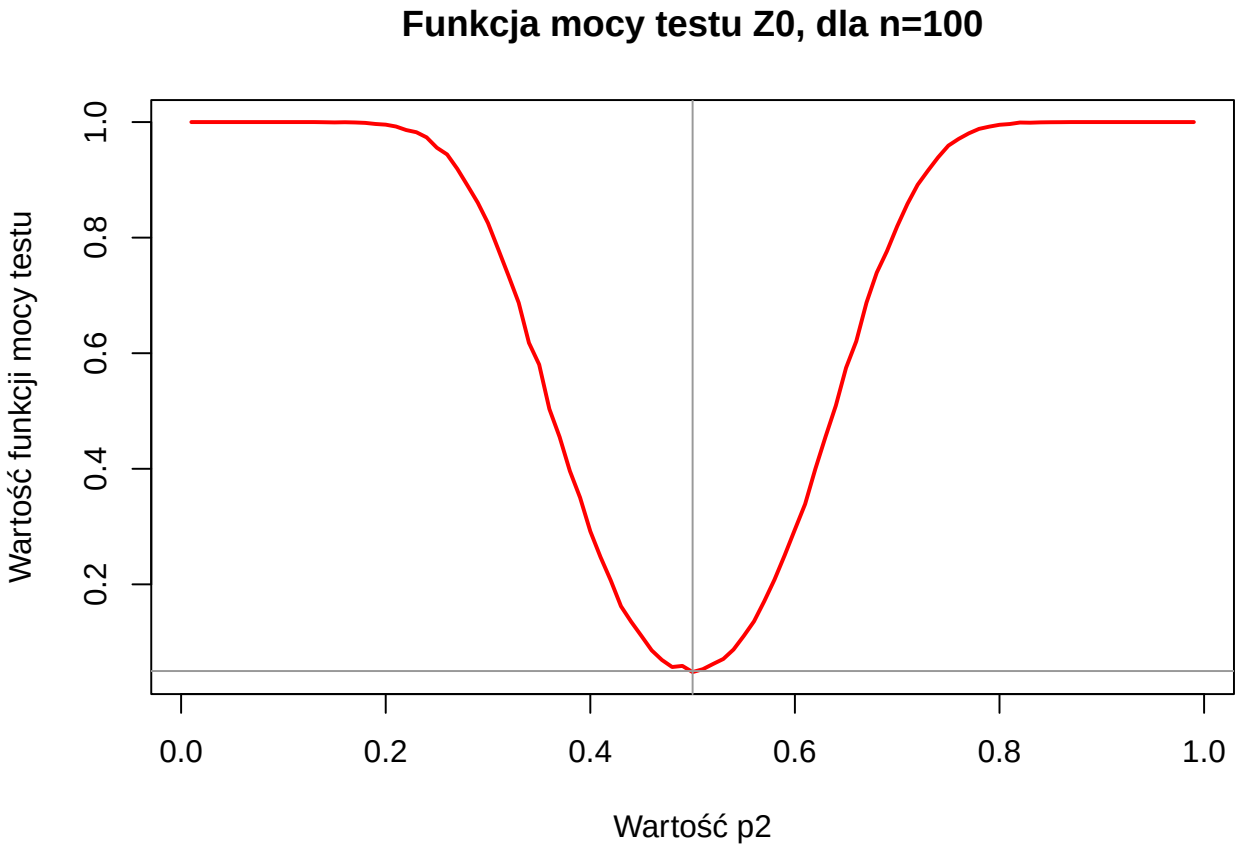


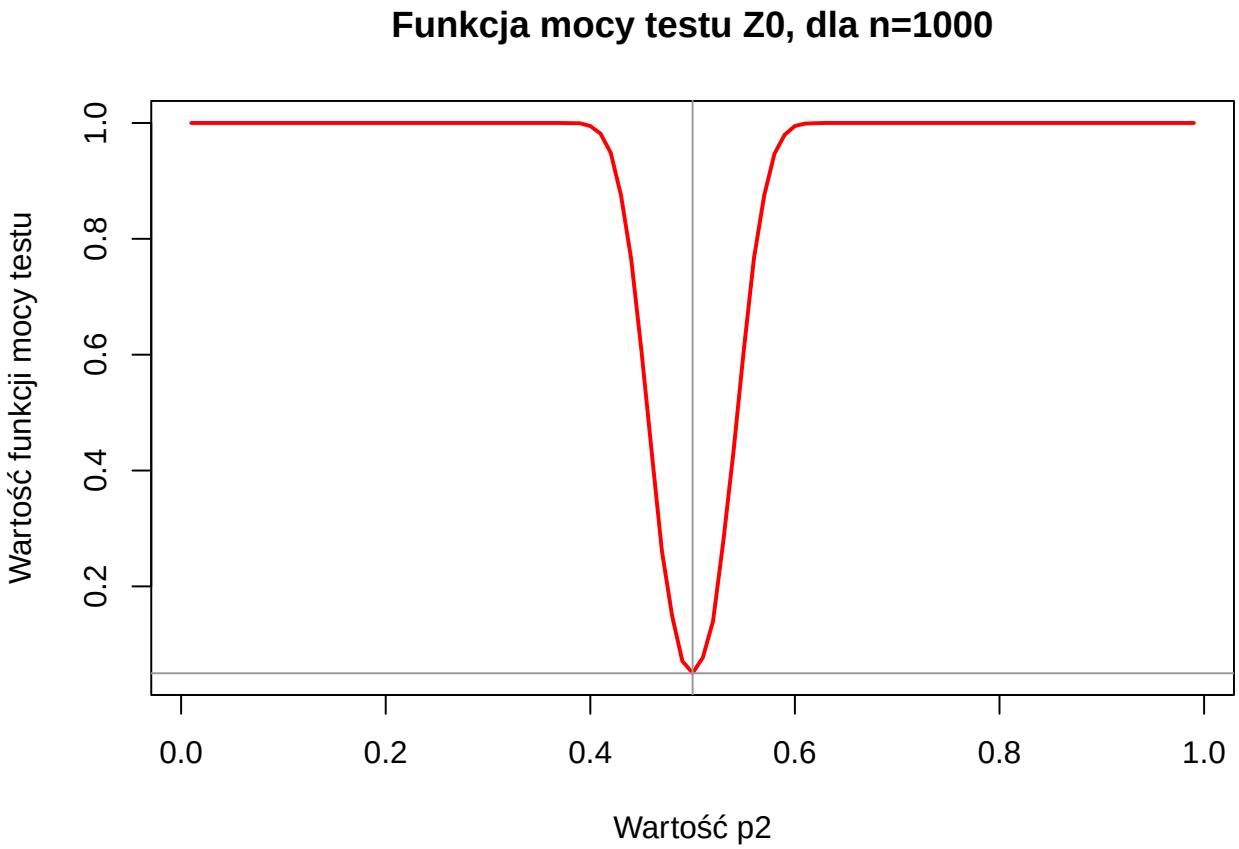
Rysunek 3. Wykres funkcji mocy testu Z dla $n = 1000$ i testu Z_0 dla $n = 20$





Rysunek 4. Wykres funkcji mocy testu Z_0 dla $n = 30$ dla $n = 50$





Rysunek 5. Wykres funkcji mocy testu Z_0 dla $n = 100$ i $n = 1000$

Na rysunkach (1), (2), (3), (4), (5) przedstawiłem wykresy funkcji mocy testów Z i Z_0 dla $n \in \{20, 30, 50, 100, 1000\}$, na podstawie symulacji Monte-Carlo dla 10^4 powtórzeń. Dodatkowo na każdym wykresie narysowałem dwie linie szarym kolorem — poziomą, dla wartości 0.05 i pionową, dla argumentu 0.5. Punkt przecięcia tych linii oznacza miejsce przez które powinna przechodzić funkcja mocy testu, ponieważ jest to miejsce, w którym prawdopodobieństwa p_1 i p_2 odpowiedzi na pytanie ankietowe są takie same, więc wtedy hipoteza zerowa powinna być przyjmowana z prawdopodobieństwem $1-\alpha$, czyli funkcja mocy testu powinna przyjmować wartość $\alpha = 0.05$. Dla zwiększających się wartości n widzimy, że wartości funkcji mocy są większe, dla $p_2 \neq 0.5$. Można było tego oczekiwać, ponieważ wraz ze wzrastającą liczbą prób (ankietowanych), test powinien być częściej odrzucany dla $p_1 \neq p_2$, bo moc testu rośnie. Dla najmniejszych z rozważanych wartości n można zauważyć, że wartość funkcji mocy testu w punkcie p_2 nieznacznie różni się od wartości α, różnice są bardzo małe, jednak dla większych n już nie są one widoczne. Można na tej podstawie wyciągnąć wniosek, że testy Z i Z_0 są asymptotycznie nieobciążone.

2.4. Wnioski

W zadaniu pierwszym należało na podstawie danych, zweryfikować hipotezę na poziomie istotności $\alpha = 0.05$, że studenci byli tak samo przygotowani do obydwu kolokwiów, przy założeniu, że ich poziom trudności był taki sam.
W tym przypadku skorzystałem z testu McNemary, zaimplementowanego i wbudowanego w pakiecie R i uzyskałem dokładnie tę samą wartość poziomu krytycznego, pozwalającą przyjąć, że studenci nie byli tak samo przygotowani na dwa kolokwia.

W zadaniu drugim należało na podstawie danych zweryfikować hipotezę na poziomie istotności $\alpha = 0.05$, że skuteczność dwóch rozważanych leków jest jednakowa. Zadanie należało wykonać dla dwóch testów — dokładnego i McNemary z poprawką na ciągłość. Obydwa testy wykonałem korzystąc z własnej i wbudowanej w pakiet R funkcji.
Uzyskane p-wartości dla obydwu testów nieznacznie się różniły, a wartości funkcji zaimplementowanych zgadzały się z wartościami funkcji wbudowanych. Uzyskane wyniki pozwoliły założyć, że hipoteza zerowa H_0 jest prawdziwa, czyli mogłem przyjąć, że skuteczność dwóch rozważanych leków jest jednakowa.

W ostatnim zadaniu z tej listy należało porównać symulacyjnie funkcje mocy testów Z i Z_0. W tym celu zaimplementowałem własne funkcje dla testów Z i Z_0. Następnie zaimplementowałem funkcję, która dla podanej jako argumenty liczby prób n (ankietowanych) i prawdopodobieństw p_1 i p_2 odpowiedzi na konkretne pytanie, zwracała macierz z ilością odpowiedzi na pierwsze i drugie pytanie. Następnie dla zadanych parametrów przeprowadziłem symulację Monte-Carlo, której celem było narysowanie wykresów i porównanie funkcji mocy rozważanych testów.
Dla obydwu testów wyciągnąłem wniosek, że są asymptotycznie nieobciążone — dla małych n wartości funkcji mocy nieco odbiegały od wartości α w punkcje p_2, które powinny osiągnąć (różnice te były bardzo małe). Funkcja mocy obydwu testów przyjmuje większe wartości (dla $p_2 \neq 0.5$) wraz ze wzrastającą liczbą prób n.

3. Lista 11

3.1. Dane do zadań

W tabeli (3) przedstawiłem dane do zadań z tej listy:

3.2. Zadanie 1.

W tym zadaniu należało zweryfikować hipotezę, że dane z tabeli (3) podlegają modelowi:

(a) symetrii,

	Wyniki z kolokwium 1.						
Wynik z kolokwium 2.	2	3	+3	4	+4	5	Suma
2	5	2	1	0	0	0	8
3	6	3	2	2	0	0	13
+3	1	4	5	5	2	2	19
4	0	10	15	18	5	2	50
+4	1	2	5	3	2	2	15
5	0	1	3	4	3	2	13
Suma	13	22	31	32	12	8	118

Tabela 3. Dane do zadania 1. i 2.

(b) quasi-symetrii,
(c) quasi-niezależności,

korzystając z odpowiednich testów. Hipotezę należało zweryfikować na poziomie istotności $\alpha = 0.05$. Trzeba było również zwrócić uwagę na problem z zastosowaniem do analizowanych danych testu Bowkera.

```
data <- matrix(
        c(5,2,1,0,0,0,6,3,2,2,0,0,1,4,5,5,2,
        2,0,10,15,18,5,2,1,2,5,3,2,2,0,1,3,4,3,2),
        nrow=6, dimnames = list("Wyniki z kolokwium 1" = c("2","3",
        "+3","4","+4","5"),
        "Wyniki z kolokwium 2" = c("2","3","+3","4","+4","5")))
data
```

```
##                       Wyniki z kolokwium 2
## Wyniki z kolokwium 1 2 3 +3  4 +4 5
##                    2  5 6  1  0  1 0
##                    3  2 3  4 10  2 1
##                    +3 1 2  5 15  5 3
##                    4  0 2  5 18  3 4
##                    +4 0 0  2  5  2 3
##                    5  0 0  2  2  2 2
```

- Dla modelu symetrii:
  Hipotezy:
  * H_0: dane pochodzą z modelu symetrii
  * H_1: dane nie pochodzą z modelu symetrii

  W pierwszym kroku skorzystamy z testu McNemary, korzystając z funkcji wbudowanej `mcnemar.test`:

```
mcnemar.test(data)$p.value
```

```
## [1] NaN
```

Uzyskałem p-wartość `NaN`, wynika to z tego, że w tych danych pojawiają się zera, więc w tym przypadku test McNemary jest nieodpowiedni.
Korzystając z wykładu, wykonamy test ilorazu wiarygodności zaimplementowanego w bibliotece `gnm`:

```
library(gnm)
```

Nasze dane muszą być przedstawione jako ramka danych

```
count <- c(5,2,1,0,0,0,6,3,2,2,0,0,1,
        4,5,5,2,2,0,10,15,18,5,2,1,2,5,3,2,2,0,1,3,4,3,2)
kol1 <- gl(6,1,labels=c("2","3","+3","4","+4","5"))
kol2 <- gl(6,6,labels=c("2","3","+3","4","+4","5"))
wyniki <- data.frame(kol1, kol2, count)
```

```
symmetry <- glm(count ~Symm(kol1, kol2), data=wyniki,
        family=poisson)
```

Wyznaczamy p-wartość

```
x <- symmetry$deviance
x
```

```
## [1] 22.28825
```

```
r <- 15
p <- 1-pchisq(x,r)
p
```

```
## [1] 0.1004656
```

Uzyskałem p-wartość większą niż założony poziom istotności $\alpha = 0.05$, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, czyli można założyć, że nasze dane pochodzą z modelu symetrii.

- Dla modelu quasi-symetrii:
  Hipotezy:
  * H_0: dane pochodzą z modelu quasi-symetrii
  * H_1: dane nie pochodzą z modelu quasi-symetrii

```
quasi.symmetry <- glm(count ~ kol1+kol2 + Symm(kol1, kol2), data=wyniki,
        family=poisson)

x <- quasi.symmetry$deviance
x
```

```
## [1] 3.72455
```

```
r <- 10
p <- 1-pchisq(x,r)
p
```

```
## [1] 0.9589187
```

Uzyskałem p-wartość znacznie wiżkszą niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, więc można założyć, że nasze dane pochodzą z modelu quasi-symetrii.

- Dla modelu quasi-niezależności:
  Hipotezy:
  * H_0: dane pochodzą z modelu quasi-niezależności

* H_1: dane nie pochodzą z modelu quasi-niezależności

```
quasi.indep <- glm(count ~ kol1 + kol2 + Diag(kol1, kol2),data=wyniki,
        family = poisson)

x <- quasi.indep$deviance
x

## [1] 36.32547

r <- 19
p <- 1-pchisq(x,r)
p

## [1] 0.00962481
```

P-wartość testu `quasi.indep` jest równa w przybliżeniu 0.009 i jest mniejsza niż założony poziom istotności α, więc są podstawy do odrzucenia hipotezy zerowej H_0 i przyjęcia hipotezy alternatywnej H_1, czyli można założyć, że nasze dane nie pochodzą z modelu quasi-niezależności.

3.3. Zadanie 2.

W tym zadaniu należało zweryfikować hipotezę, na poziomie istotności $\alpha = 0.05$, że studenci byli tak samo przygotowani do obu kolokwiów, zakładając że poziom trudności zadań był taki sam na pierwszy i drugim kolokwium. Dane do zadania przedstawione są w tabeli (3). Hipotezy dla testu:

- H_0 : rozkłady brzegowe są jednorodne — studenci byli tak samo przygotowani do obu kolokwiów
- H_1 : rozkłady brzegowe nie są jednorodne — studenci nie byli tak samo przygotowani do obu kolokwiów.

Model symetrii w przypadku tabel większych niż 2 na 2 implikuje jednorodność. W związku z tym, że w tym zadaniu analizujemy dokładnie te same dane, co w zadaniu 1. z tej listy, a w zadaniu 1a) (3.2) nie odrzuciliśmy hipotezy o modelu symetrii, to korzystając z tej implikacji, że model symetrii implikuje jednorodność rozkładów brzegowych, nie ma podstaw do odrzucenia hipotezy zerowej H_0, więc można założyć, że studenci byli tak samo przygotowani do obu kolokwiów.

3.4. Wnioski

W zadaniu 1. (3.2) należało sprawdzić, czy dane z tabeli (3), pochodzą z modelu:

- symetrii
- quasi-symetrii
- quasi-niezależności.

Przyjęłem poziom istotności $\alpha = 0.05$ i korzystając z odpowiednich testów możemy założyć, że rozważane dane pochodzą z modelu symetrii i quasi-symetrii, ale nie pochodzą z modelu quasi-niezależności.

W zadaniu drugim należało sprawdzić, czy studenci byli tak samo przygotowani na dwa kolokwia, przy założeniu, że poziom trudności obu kolokwiów był taki sam. W tym zadaniu powołałem się na wynik z zadania 1a), ponieważ dla tabel większych niż 2 na 2 model symetrii implikuje jednorodność rozkładów brzegowych, a w zadaniu 1a) nie było podstaw do odrzucenia hipotezy, że nasze dane pochodzą z modelu symetrii, więc można było na tej podstawie założyć jednorodność rozkładów brzegowych, a tym samym przyjąć hipotezę, że studenci byli tak samo przygotowani na dwa kolokwia.

4. Lista 12_13_14

4.1. Dane do zadań

Wszystkie zadania z tej listy należało wykonać dla danych z pliku *Ankieta.csv*.

```
dane <- read.csv2("Ankieta.csv")
summary(dane)
```

```
##       SEN           BIEGANIE         PIES
##  Min.   :0.000   Min.   :0.00   Min.   :0.000
##  1st Qu.:0.000   1st Qu.:0.00   1st Qu.:0.000
##  Median :1.000   Median :1.00   Median :1.000
##  Mean   :0.725   Mean   :0.55   Mean   :0.575
##  3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.000
##  Max.   :1.000   Max.   :1.00   Max.   :1.000
```

Dane zawierają wyniki ankietowania 40 losowo wybranych studentów PWr. W ankiecie należało odpowiedzieć na trzy pytania:

- Czy dobrze sypiasz?
- Czy regularnie biegasz?
- Czy masz psa?

Liczba 1 oznacza odpowiedź „tak”, a liczba 0 – „nie”.

Uwaga: Jako reprezentacje zmiennych (SEN, BIEGANIE, PIES) do budowania modeli log-liniowych przyjęłem:

- 1 — SEN
- 2 — BIEGANIE
- 3 — PIES

4.2. Zadanie 1.

W tym zadaniu należało podać interpretację następujących modeli log-liniowych:

$$\begin{array}{lll} \text{(a)}[1\ 3] & \text{(b)}[13] & \text{(c)}[1\ 2\ 3] \qquad (1) \\ \text{(d)}[12\ 3] & \text{(e)}[12\ 13] & \text{(f)}[1\ 23], \qquad (2) \end{array}$$

zbudować model na podstawie danych z pliku, przeprowadzić test statystyczny, że dane pochodzą z określonego modelu log-liniowego i porównać wyznaczone przez model liczności z licznościami danych.

W pierwszym kroku zaimportowałem potrzebne biblioteki i wczytałem oraz odpowiednio przygotowałem dane:

```
library(dplyr)
library(tidyverse)
```

```
dane <- read.csv2("Ankieta.csv")
dane <- mutate(dane, across(c("SEN", "BIEGANIE", "PIES"), as.factor))
dane <- ftable(dane)
dane.df <- as.data.frame(dane)
dane.df
```

```
##   SEN BIEGANIE PIES Freq
## 1   0        0    0    6
## 2   1        0    0    5
## 3   0        1    0    1
## 4   1        1    0    5
## 5   0        0    1    2
## 6   1        0    1    5
## 7   0        1    1    2
## 8   1        1    1   14
```

Dla każdego modelu przeprowadziłem test statystyczny, sprawdzający czy dane pochodzą z tego modelu.
Hipotezy dla testów:

- H_0 : Dane pochodzą z tego (określonego) modelu
- H_1 : Dane pochodzą z modelu pełnego (uwzględniającego czynniki główne, interakcje pierwszego i drugiego rzędu)

Testy wykonujemy na poziomie istotności $\alpha = 0.05$, a rozważane modele są hierarchiczne uporządkowane.

(a) [1 3] — zmienne „SEN" i „PIES" mają dowolne rozkłady oraz zmienne te są niezależne, a zmienna „BIEGANIE" ma rozkład równomierny

```
model_a <- glm(Freq ~ SEN + PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_a), df = df.residual(model_a))
```

```
## [1] 0.04818022
```

P-wartość jest mniejsza niż założony poziom istotności α, więc można założyć, że hipoteza zerowa H_0 jest nieprawdziwa — nasze dane nie pochodzą z modelu [1 3].

```
cbind(model_a$data, fitted(model_a))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_a)
## 1   0        0    0    6          2.3375
## 2   1        0    0    5          6.1625
## 3   0        1    0    1          2.3375
## 4   1        1    0    5          6.1625
## 5   0        0    1    2          3.1625
## 6   1        0    1    5          8.3375
## 7   0        1    1    2          3.1625
## 8   1        1    1   14          8.3375
```

(b) [13] — zmienne „SEN" i „PIES" mają dowolne rozkłady oraz zmienne te nie są niezależne, a zmienna „BIEGANIE" ma rozkład równomierny

```
model_b <- glm(Freq ~ SEN + PIES + SEN * PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_b), df = df.residual(model_b))
```

```
## [1] 0.07793519
```

P-wartość jest większa niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, czyli można założyć, że nasze dane pochodzą z modelu [13].

```
cbind(model_b$data, fitted(model_b))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_b)
## 1   0        0    0    6             3.5
## 2   1        0    0    5             5.0
## 3   0        1    0    1             3.5
## 4   1        1    0    5             5.0
## 5   0        0    1    2             2.0
## 6   1        0    1    5             9.5
## 7   0        1    1    2             2.0
## 8   1        1    1   14             9.5
```

(c) [1 2 3] — zmienne „SEN", „BIEGANIE" i „PIES" mają dowolne rozkłady oraz zmienne te są niezależne

```
model_c <- glm(Freq ~ SEN + BIEGANIE  + PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_c), df = df.residual(model_c))
```

```
## [1] 0.02932791
```

P-wartość jest mniejsza niż założony poziom istotności α, więc można założyć, ze hipoteza zerowa H_0 jest nieprawdziwa — nasze dane nie pochodzą z modelu [1 2 3].

```
cbind(model_c$data, fitted(model_c))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_c)
## 1   0        0    0    6         2.10375
## 2   1        0    0    5         5.54625
## 3   0        1    0    1         2.57125
## 4   1        1    0    5         6.77875
## 5   0        0    1    2         2.84625
## 6   1        0    1    5         7.50375
## 7   0        1    1    2         3.47875
## 8   1        1    1   14         9.17125
```

(d) [12 3] — zmienne „SEN", „BIEGANIE" i „PIES" mają dowolne rozkłady, zmienna „PIES" jest niezależna od zmiennej „SEN" i „BIEGANIE", zmienne „SEN" i „BIEGANIE" nie są niezależne

```
model_d <- glm(Freq ~ SEN + BIEGANIE + SEN * BIEGANIE + PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_d), df = df.residual(model_d))
```

```
## [1] 0.1131637
```

P-wartość jest większa niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, czyli można założyć, że nasze dane pochodzą z modelu [12 3].

```
cbind(model_d$data, fitted(model_d))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_d)
## 1   0        0    0    6          3.400
## 2   1        0    0    5          4.250
## 3   0        1    0    1          1.275
## 4   1        1    0    5          8.075
## 5   0        0    1    2          4.600
## 6   1        0    1    5          5.750
## 7   0        1    1    2          1.725
## 8   1        1    1   14         10.925
```

(e) [12 13] — zmienne „SEN", „BIEGANIE" i „PIES" mają dowolne rozkłady, zmienne „SEN" i „BIEGANIE" nie są niezależne, zmienne „SEN" i „PIES" nie są niezależne, a zmienne „BIEGANIE" i „PIES" są niezależne

```
model_e <- glm(Freq ~ SEN + BIEGANIE + SEN * BIEGANIE + SEN * PIES + PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_e), df = df.residual(model_e))
```

```
## [1] 0.201565
```

P-wartość jest większa niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, czyli można założyć, że nasze dane pochodzą z modelu [12 13].

```
cbind(model_e$data, fitted(model_e))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_e)
## 1   0        0    0    6        5.090909
## 2   1        0    0    5        3.448276
## 3   0        1    0    1        1.909091
## 4   1        1    0    5        6.551724
## 5   0        0    1    2        2.909091
## 6   1        0    1    5        6.551724
## 7   0        1    1    2        1.090909
## 8   1        1    1   14       12.448276
```

(f) [1 23] — zmienne „SEN", „BIEGANIE" i „PIES" mają dowolne rozkłady, zmienne „BIEGANIE" i „PIES" nie są niezależne, a zmienne „SEN" i „BIEGANIE" oraz „SEN" i „PIES" są niezależne

```
model_f <- glm(Freq ~ SEN + BIEGANIE + BIEGANIE * PIES + PIES,
data = dane.df, family = poisson)

1-pchisq(deviance(model_f), df = df.residual(model_f))
```

```
## [1] 0.1089104
```

P-wartość jest większa niż założony poziom istotności α, więc nie ma podstaw do odrzucenia hipotezy zerowej H_0, czyli można założyć, że nasze dane pochodzą z modelu [1 23].

```
cbind(model_f$data, fitted(model_f))
```

```
##   SEN BIEGANIE PIES Freq fitted(model_f)
## 1   0        0    0    6           3.025
## 2   1        0    0    5           7.975
## 3   0        1    0    1           1.650
## 4   1        1    0    5           4.350
## 5   0        0    1    2           1.925
## 6   1        0    1    5           5.075
## 7   0        1    1    2           4.400
## 8   1        1    1   14          11.600
```

	Modele					
	[1 3]	[13]	[1 2 3]	[12 3]	[12 13]	[1 23]
P-wartość	0.0481802	0.0779352	0.0293279	0.1131637	0.201565	0.1089104

Tabela 4. P-wartości testów statystycznych

Liczności danych	Liczności modeli					
	[1 3]	[13]	[1 2 3]	[12 3]	[12 13]	[1 23]
6	2.3375	3.5	2.10375	3.4	5.0909091	3.025
5	6.1625	5	5.54625	4.25	3.4482759	7.975
1	2.3375	3.5	2.57125	1.275	1.9090909	1.65
5	6.1625	5	6.77875	8.075	6.5517241	4.35
2	3.1625	2	2.84625	4.6	2.9090909	1.925
5	8.3375	9.5	7.50375	5.75	6.5517241	5.075

Tabela 5. Porównanie wyznaczonych liczności na podstawie modelów z rzeczywistymi licznościami danych

W tabelach (4) i (5) przedstawiłem zbiorczo p-wartości testów statystycznych i porównanie liczności z danych z licznościami wynikającymi z modeli.

4.3. Zadanie 2.

W tym zadaniu należało oszacować prawdopodobieństwo:

(a) dobrej jakości snu studenta, który regularnie biega,
(b) tego, że student biega regularnie, gdy posiada psa.

przyjmując model log-liniowy [12 3]. Należało również odpowiedzieć na pytanie jakie byłyby oszacowania powyższych prawdopodobieństw, przy założeniu modelu [12 23].
Aby wykonać to zadanie musimy wyznaczyć prawdopodobieństwa warunkowe. Wzór na prawdopodobieństwo warunkowe:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad \text{gdzie:} \tag{3}$$

- A i B to jakieś zdarzenia losowe
- $P(B) > 0$

W pierwszym kroku wyznaczyłem model [12 3]:

```
dane.df
```

```
##   SEN BIEGANIE PIES Freq
## 1   0        0    0    6
## 2   1        0    0    5
## 3   0        1    0    1
## 4   1        1    0    5
## 5   0        0    1    2
## 6   1        0    1    5
## 7   0        1    1    2
## 8   1        1    1   14
```

```
model <- glm(Freq ~ SEN*BIEGANIE + PIES, data = dane.df, family = poisson)
(result <- cbind(model$data, fitted(model)))
```

```
##   SEN BIEGANIE PIES Freq fitted(model)
## 1   0        0    0    6         3.400
## 2   1        0    0    5         4.250
## 3   0        1    0    1         1.275
## 4   1        1    0    5         8.075
## 5   0        0    1    2         4.600
## 6   1        0    1    5         5.750
## 7   0        1    1    2         1.725
## 8   1        1    1   14        10.925
```

Wartości w kolumnach z licznościami modelu i licznościami wyznaczonymi na podstawie modelu log-liniowego [12 3] zmieniłem na wartości prawdopodobieństw otrzymania tych wartości:

```
result$`fitted(model)` <- result$`fitted(model)`/sum(result$`fitted(model)`)
result$Freq <- result$Freq/sum(result$Freq)
result
```

```
##   SEN BIEGANIE PIES  Freq fitted(model)
## 1   0        0    0 0.150      0.085000
## 2   1        0    0 0.125      0.106250
## 3   0        1    0 0.025      0.031875
## 4   1        1    0 0.125      0.201875
## 5   0        0    1 0.050      0.115000
## 6   1        0    1 0.125      0.143750
## 7   0        1    1 0.050      0.043125
## 8   1        1    1 0.350      0.273125
```

- Dla podpunktu (a):
  Wyznaczyłem prawdopodobieństwa warunkowe dla liczności z przyjętego modelu log-liniowego, korzystając ze wzoru (3), gdzie:
  * A — zdarzenie, że student dobrze śpi
  * B — zdarzenie, że student regularnie biega

```
(sum(result$`fitted(model)`[result$BIEGANIE == 1 & result$SEN == 1])
)/(sum(result$`fitted(model)`[result$BIEGANIE == 1]))
```

```
## [1] 0.8636364
```

i dla liczności wynikających z danych:

```
(sum(result$Freq[result$BIEGANIE == 1 & result$SEN == 1]
))/(sum(result$Freq[result$BIEGANIE == 1]))
```

```
## [1] 0.8636364
```

- dla podpunktu (b):
  Wyznaczyłem prawdopodobieństwa warunkowe dla liczności z przyjętego modelu log-liniowego, korzystając ze wzoru (3), gdzie:
  * A — zdarzenie, że student biega regularnie
  * B — zdarzenie, że student posiada psa

```
(sum(result$`fitted(model)`[result$PIES == 1 & result$BIEGANIE == 1])
)/((sum(result$`fitted(model)`[result$PIES == 1])))
```

```
## [1] 0.55
```

i dla liczności wynikających z danych:

```
(sum(result$Freq[result$PIES == 1 & result$BIEGANIE == 1])
)/((sum(result$Freq[result$PIES == 1])))
```

```
## [1] 0.6956522
```

Analogicznie dla modelu [12 23]:

```
model <- glm(Freq ~ SEN*BIEGANIE + BIEGANIE*PIES, data = dane.df, family = poisson)
result <- cbind(model$data, fitted(model))
result$`fitted(model)` <- result$`fitted(model)`/sum(result$`fitted(model)`)
result$Freq <- result$Freq/sum(result$Freq)
```

- Dla podpunktu (a):
  * Dla liczności wynikających z modelu:

    ```
    ## [1] 0.8636364
    ```

  * Dla liczności wynikających z danych:

    ```
    ## [1] 0.8636364
    ```

- Dla podpunktu (b):
  * Dla liczności wynikających z modelu:

    ```
    ## [1] 0.6956522
    ```

  * Dla liczności wynikających z danych:

    ```
    ## [1] 0.6956522
    ```

4.4. Zadanie 3.

W tym zadaniu należało zweryfikować następujące hipotezy:

(a) Zmienne losowe *Sen*, *Bieganie* i *Pies* są wzajemnie niezależne
(b) Zmienna losowa *Pies* jest niezależna od pary zmiennych *Sen* i *Bieganie*
(c) zmienna losowa *Sen* jest niezależna od zmiennej *Pies*, przy ustalonej zmiennej *Bieganie*.

W celu rozwiązania tego zadania, należało zbudować odpowiedni model log-liniowy i przeprowadzić odpowieni test statystyczny:

- Dla podpunktu (a):
  Zmienne losowe („SEN", „BIEGANIE" i „PIES") są wzajemnie niezależne.
  Hipotezy dla testów:
  - H_0: Zmienne losowe *Sen*, *Bieganie* i *Pies* są wzajemnie niezależne,
  - H_1: Zmienne losowe *Sen*, *Bieganie* i *Pies* nie są wzajemnie niezależne

  Zbudowałem model, który jest dobrą interpretacją tego polecenia:

```
model_a <- glm(Freq ~ SEN + BIEGANIE + PIES, data = dane.df, family = poisson)
```

  oraz pewne dwa modele, w których jest jeden pełny (zawierający wszystkie interakcje) a drugi jest nadmodelem rozważanego modelu, ale nie jest modelem pełnym.

```
model_pelny <- glm(Freq ~ (SEN + BIEGANIE + PIES)^3, data = dane.df,
        family = poisson)
nadmodel_a <- glm(Freq ~ SEN + BIEGANIE + PIES + BIEGANIE*PIES,
        data = dane.df, family = poisson)
```

  Teraz przeprowadziłem test równości wariancji, korzystając z funkcji `anova` w pakiecie R, i wyznaczyłem p-wartość:
  * Dla modelu ogólnego:

```
test <- anova(model_a, model_pelny)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.02932791
```

  * Dla nadmodelu:

```
test <- anova(model_a, nadmodel_a)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.02999623
```

- Dla podpunktu (b):
  Hipotezy dla testu:
  - H_0: Zmienna losowa *Pies* jest niezależna od pary zmiennych *Sen* i *Bieganie*,
  - H_1: Zmienna losowa *Pies* nie jest niezależna od pary zmiennych *Sen* i *Bieganie*.

  Analogicznie jak dla podpunktu (a), zbudowałem model:

```
model_b <- glm(Freq ~ SEN * BIEGANIE + PIES + SEN + BIEGANIE,
        data = dane.df, family = poisson)
```

  teraz nadmodel:

```
nadmodel_b <- glm(Freq ~ SEN * BIEGANIE + PIES * SEN,
data = dane.df, family = poisson)
```

  Testy statystyczne i p-wartości:
  * Dla modelu ogólnego:

```
test <- anova(model_b, model_pelny)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.1131637
```

  * Dla nadmodelu:

```
test <- anova(model_b, nadmodel_b)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.09634701
```

- Dla podpunktu (c):
  Hipotezy dla testów:
  - H_0: Zmienna losowa *Sen* jest niezależna od zmiennej *Pies*, przy ustalonej zmiennej *Bieganie*
  - H_1: Zmienna losowa *Sen* nie jest niezależna od zmiennej *Pies*, przy ustalonej zmiennej *Bieganie*

  Analogicznie jak dla powyższych podpunktów, zbudowałem model:

```
model_c <- glm(Freq ~ SEN * BIEGANIE + BIEGANIE * PIES,
        data = dane.df, family = poisson)
```

  teraz nadmodel:

```
nadmodel_c <- glm(Freq ~ SEN * BIEGANIE + BIEGANIE * PIES + SEN * PIES,
        data = dane.df, family = poisson)
```

  Testy statystyczne i p-wartości:
  * Dla modelu ogólnego:

```
test <- anova(model_c, model_pelny)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.5329187
```

  * Dla nadmodelu:

```
test <- anova(model_c, nadmodel_c)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.3057874
```

4.5. Zadanie 4.

W tym zadaniu należało dokonać wyboru modelu log-liniowego w oparciu o:

(a) testy
(b) kryterium AIC
(c) kryterium BIC

W pierwszym kroku zaimportowałem potrzebne biblioteki:

```
library(tidyverse)
library(dplyr)
```

wczytałem i odpowiednio sformatowałem dane:

```
dane <- read.csv2("Ankieta.csv")
dane <- mutate(dane, across(c("SEN", "BIEGANIE", "PIES"), as.factor))
dane <- ftable(dane)
dane.df <- as.data.frame(dane)
```

(a) dla testów:
Wybór modelu w oparciu o testy wykonywałem w ten sposób, że najpierw wziąłem model, w którym nie występują żadne interakcje — [1 2 3] i wykonałem test ilorazu wiarygodności

(anova z pakietu R), gdzie hipotezą alternatywną H_1 był model w którym dodawałem kolejne interakcje. Jeśli uzyskana p-wartość była większa niż założony poziom istotności $\alpha = 0.05$, to przyjmowałem że model z hipotezy zerowej H_0 lepiej opisuje nasze dane, w przeciwnym wypadku, dla następnych testów przyjmowałem model z hipotezy alternatywnej jako model wyjściowy.
Postępując w ten sposób doszedłem do modelu, który według testów najlepiej opisuje dane.

- model [1 2 3] przeciwko modelowi [12 3]:

```
model_1_2_3 <- glm(Freq ~ SEN + BIEGANIE + PIES,
data = dane.df, family = poisson)
model_12_3 <- glm(Freq ~ SEN + BIEGANIE + SEN * BIEGANIE + PIES,
data = dane.df, family = poisson)

test <- anova(model_1_2_3, model_12_3)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.02850318
```

  uzyskana p-wartość nie przekracza założonego poziomu istotności, więc można przyjąć, że model [12 3] jest teraz modelem wyjściowym,
- model [12 3] przeciwko modelowi [13 2]:

```
model_13_2 <- glm(Freq ~ SEN + BIEGANIE + PIES + SEN*PIES,
data = dane.df, family = poisson)

test <- anova(model_12_3, model_13_2)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 1
```

  Uzyskana p-wartość jest większa niż założony poziom istotności $\alpha = 0.05$, więc model [12 3] pozostaje jako model wyjściowy
- model [12 3] przeciwko modelowi [1 23]:

```
model_1_23 <- glm(Freq ~ SEN + (BIEGANIE + PIES)^2,
data = dane.df, family = poisson)

test <- anova(model_12_3, model_1_23)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 1
```

  uzyskana p-wartość jest równa 1 więc model z hipotezy zerowej H_0 — [12 3] pozostaje modelem wyjściowym
- model [12 3] przeciwko modelowi [12 23]:

```
model_12_23 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (BIEGANIE + PIES)^2,
data = dane.df, family = poisson)

test <- anova(model_12_3, model_12_23)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.02999623
```

  Uzyskana wartość poziomu krytycznego jest mniejsza niż założony poziom istotności α, więc przyjmujemy model z hipotezy alternatywnej H_1 — [12 23] jako model wyjściowy,
- model [12 23] przeciwko modelowi [12 13]:

```
model_12_13 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (SEN + PIES)^2,
data = dane.df, family = poisson)
```

```
test <- anova(model_12_23, model_12_13)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 1
```

P-wartość w teście jest równa 1, więc model [12 23] pozostaje jako model wyjściowy,

- model [12 23] przeciwko modelowi [12 13 23]:

```
model_12_13_23 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (SEN + PIES)^2 + (BIEGANIE + PIES
data = dane.df, family = poisson)

test <- anova(model_12_23, model_12_13_23)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.3057874
```

P-wartość jest większa niż założony poziom istotności, więc model [12 23] pozostaje jako model wyjściowy,

- model [12 23] przeciwko modelowi [123]:

```
model_123 <- glm(Freq ~ (SEN + BIEGANIE + PIES)^3,
data = dane.df, family = poisson)

test <- anova(model_12_23, model_123)
1-pchisq(test$Deviance[2], df = test$Df[2])
## [1] 0.5329187
```

Uzyskana p-wartość jest większa niż założony poziom istotności, więc zakładamy, że model [12 23] lepiej się dopasowuje do danych niż model [123].

Przeprowadzając testy, zaczynając od modelu [1 2 3], po kolei dodając interakcje, doszedłem do testu, w którym testowałem model [12 23] przeciwko modelowi [123] i uzyskałem p-wartość, która pozwala założyć, że model [12 23] lepiej opisuje dane niż model [123]. Jednocześnie model [123] jest modelem zawierającym wszystkie możliwe interakcje, więc możemy założyć, że jeśli chodzi o testy ilorazu wiarygodności, to model [12 23] najlepiej opisuje nasze dane.

(b) wybór modelu w oparciu o kryterium AIC:
w tym przypadku skorzystałem z funkcji `AIC` i wyznaczyłem wartości kryterium AIC dla każdego z możliwych 19 modeli:

```
model_ <- glm(Freq ~ 1, data = dane.df, family = poisson)
model_1 <- glm(Freq ~ SEN, data = dane.df, family = poisson)
model_2 <- glm(Freq ~ BIEGANIE, data = dane.df, family = poisson)
model_3 <- glm(Freq ~ PIES, data = dane.df, family = poisson)
model_1_2 <- glm(Freq ~ SEN + BIEGANIE, data = dane.df, family = poisson)
model_1_3 <- glm(Freq ~ SEN + PIES, data = dane.df, family = poisson)
model_2_3 <- glm(Freq ~ BIEGANIE + PIES, data = dane.df, family = poisson)
model_12 <- glm(Freq ~ (SEN + BIEGANIE)^2, data = dane.df, family = poisson)
model_13 <- glm(Freq ~ (SEN + PIES)^3, data = dane.df, family = poisson)
model_23 <- glm(Freq ~ (BIEGANIE + PIES)^2,
        data = dane.df, family = poisson)
model_1_2_3 <- glm(Freq ~ SEN + BIEGANIE + PIES,
        data = dane.df, family = poisson)
model_12_3 <- glm(Freq ~ (SEN + BIEGANIE)^2 + PIES,
        data = dane.df, family = poisson)
model_13_2 <- glm(Freq ~ (SEN + PIES)^2 + BIEGANIE,
        data = dane.df, family = poisson)
model_1_23 <- glm(Freq ~ SEN + (BIEGANIE + PIES)^2,
```

```
        data = dane.df, family = poisson)
model_12_23 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (BIEGANIE + PIES)^2,
        data = dane.df, family = poisson)
model_13_23 <- glm(Freq ~ (SEN + PIES)^2 + (BIEGANIE + PIES)^2,
        data = dane.df, family = poisson)
model_12_13 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (SEN + PIES)^2,
        data = dane.df, family = poisson)
model_12_23_13 <- glm(Freq ~ (SEN + BIEGANIE)^2 + (BIEGANIE + PIES)^2 +
        (SEN + PIES)^2, data = dane.df, family = poisson)
model_123 <- glm(Freq ~ (SEN + BIEGANIE + PIES)^3,
        data = dane.df, family = poisson)
```

Najlepszym modelem będzie ten, dla którego wartość kryterium AIC będzie najmniejsza:

```
models_AIC <- c(AIC(model_), AIC(model_1), AIC(model_2), AIC(model_3),
AIC(model_1_2), AIC(model_1_3), AIC(model_2_3), AIC(model_12),
AIC(model_13), AIC(model_23), AIC(model_1_2_3), AIC(model_12_3),
AIC(model_13_2), AIC(model_1_23), AIC(model_12_23), AIC(model_13_23),
AIC(model_12_13), AIC(model_12_23_13), AIC(model_123))

min(models_AIC)

## [1] 39.07422

which(min(models_AIC) == models_AIC)

## [1] 15
```

Najmniejsza wartość kryterium AIC wynosi 39.0742229 i model dla którego ta wartość została osiągnięta, to model nr 15 — w naszym wektorze `models_AIC`, czyli model [12 23]. Wartości kryterium AIC dla wszystkich modeli umieściłem w tabeli (6).
Sprawdziłem jeszcze jaki model zostanie wybrany, korzystając z funkcji wbudowanej `step` w pakiecie R. W tym celu jako argument tej funkcji podałem model, w którym występują wszystkie interakcje:

```
model <- glm(Freq ~ (SEN + BIEGANIE + PIES)^3,
data = dane.df, family = poisson)

step(model)

## Start:  AIC=41.82
## Freq ~ (SEN + BIEGANIE + PIES)^3
##
##                     Df Deviance    AIC
## - SEN:BIEGANIE:PIES  1  0.20999 40.025
## <none>                  0.00000 41.815
##
## Step:  AIC=40.03
## Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + SEN:PIES + BIEGANIE:PIES
##
##                 Df Deviance    AIC
## - SEN:PIES       1   1.2588 39.074
## <none>               0.2100 40.025
```

```
## - BIEGANIE:PIES  1   3.2033 41.019
## - SEN:BIEGANIE   1   3.2912 41.107
##
## Step:  AIC=39.07
## Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + BIEGANIE:PIES
##
##                 Df Deviance    AIC
## <none>               1.2588 39.074
## - BIEGANIE:PIES  1   5.9683 41.784
## - SEN:BIEGANIE   1   6.0561 41.872
##
## Call:  glm(formula = Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + BIEGANIE:PIES,
##     family = poisson, data = dane.df)
##
## Coefficients:
##     (Intercept)             SEN1        BIEGANIE1            PIES1
##          1.5870           0.2231          -1.7876          -0.4520
##  SEN1:BIEGANIE1  BIEGANIE1:PIES1
##          1.6227           1.4328
##
## Degrees of Freedom: 7 Total (i.e. Null);  2 Residual
## Null Deviance:     20.47
## Residual Deviance: 1.259  AIC: 39.07
```

Model, na którym zatrzymał się program, to model [12 23], zgadza się to z naszymi obliczeniami.

(c) wybór modelu w oparciu o kryterium BIC:
W tym przypadku sposób wyboru jest analogiczny jak dla kryterium AIC, zmieniamy jedynie kryterium z AIC na BIC.
Najlepszym modelem będzie ten, dla którego wartość kryterium BIC będzie najmniejsza:

```
models_BIC <- c(BIC(model_), BIC(model_1), BIC(model_2), BIC(model_3),
BIC(model_1_2), BIC(model_1_3), BIC(model_2_3), BIC(model_12),
BIC(model_13), BIC(model_23), BIC(model_1_2_3), BIC(model_12_3),
BIC(model_13_2), BIC(model_1_23), BIC(model_12_23), BIC(model_13_23),
BIC(model_12_13), BIC(model_12_23_13), BIC(model_123))

min(models_BIC)

## [1] 39.55087

which(min(models_BIC) == models_BIC)

## [1] 15
```

Najmniejsza wartość kryterium BIC wynosi 39.5508722 i model dla którego ta wartość została osiągnięta, to model nr 15 — w naszym wektorze `models_BIC`, czyli model [12 23]. Wartości kryterium BIC dla wszystkich modeli umieściłem w tabeli (6).
Sprawdziłem jeszcze jaki model zostanie wybrany, korzystając z funkcji wbudowanej `step` w pakiecie R. W tym celu jako argument tej funkcji podałem model, w którym występują wszystkie interakcje oraz argument `k`, który dla kryterium BIC wynosi $\ln n$, gdzie n to ilość ankietowanych:

```
n <- nrow(read.csv2("Ankieta.csv"))
step(model, k=log(n))
```

```
## Start:  AIC=55.33
## Freq ~ (SEN + BIEGANIE + PIES)^3
##
##                     Df Deviance    AIC
## - SEN:BIEGANIE:PIES  1  0.20999 51.848
## <none>                  0.00000 55.326
##
## Step:  AIC=51.85
## Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + SEN:PIES + BIEGANIE:PIES
##
##                 Df Deviance    AIC
## - SEN:PIES       1   1.2588 49.207
## - BIEGANIE:PIES  1   3.2033 51.152
## - SEN:BIEGANIE   1   3.2912 51.240
## <none>               0.2100 51.848
##
## Step:  AIC=49.21
## Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + BIEGANIE:PIES
##
##                 Df Deviance    AIC
## <none>               1.2588 49.207
## - BIEGANIE:PIES  1   5.9683 50.228
## - SEN:BIEGANIE   1   6.0561 50.316
##
## Call:  glm(formula = Freq ~ SEN + BIEGANIE + PIES + SEN:BIEGANIE + BIEGANIE:PIES,
##     family = poisson, data = dane.df)
##
## Coefficients:
##     (Intercept)             SEN1        BIEGANIE1            PIES1
##          1.5870           0.2231          -1.7876          -0.4520
##  SEN1:BIEGANIE1  BIEGANIE1:PIES1
##          1.6227           1.4328
##
## Degrees of Freedom: 7 Total (i.e. Null);  2 Residual
## Null Deviance:     20.47
## Residual Deviance: 1.259  AIC: 39.07
```

Funkcja `step` również zatrzymała się na modelu [12 23].

4.6. Wnioski

Wszystkie zadania z tej listy należało przeprowadzić na podstawie danych z pliku *Ankieta.csv*.

W zadaniu pierwszym należało podać interpretację zadanych modeli log-liniowych, zbudować model na ich podstawie, przeprowadzić test statystyczny, że dane pochodzą z tego modelu log-liniowego i porównać liczności wynikające z modelu i liczności z danych.

Po podaniu interpretacji zadanych modeli, zbudowałem modele i przeprowadziłem testy. Porównując liczności modeli — kolumny `fitted`, w których p-wartość pozwoliła przyjąć, że dane pochodzą z tego modelu, z licznościami wynikającymi z danych — kolumny `Freq`, można zauważyć, że rozbieżności nie były zbyt duże.

Model	Kryterium AIC	Kryterium BIC
[]	48.283451	48.3628925
[1]	41.8851787	42.0440618
[2]	49.8827816	50.0416647
[3]	49.3800452	49.5389283
[1 2]	43.4845094	43.722834
[1 3]	42.981773	43.2200976
[2 3]	50.9793759	51.2177005
[12]	40.6871369	41.0049031
[13]	42.2167784	42.5345445
[23]	48.2698676	48.5876337
[1 2 3]	44.5811036	44.8988698
[12 3]	41.7837312	42.1809389
[13 2]	43.816109	44.2133167
[1 23]	41.8715953	42.268803
[12 23]	39.0742229	39.5508722
[13 23]	41.1066007	41.58325
[12 13]	41.0187366	41.4953858
[12 23 13]	40.0254391	40.5815299
[123]	41.8154503	42.4509826

Tabela 6. Wartości kryteriów AIC i BIC dla modeli

W drugim zadaniu należało oszacować prawdopodobieństwa dobrej jakości snu studenta, który regularnie biega oraz tego, że student regularnie biega, gdy posiada psa. Należało przyjąć model log-liniowy [12 3] oraz odpowiedzieć na pytanie, jakie byłyby oszacowania tych samych prawdopodobieństw, przy założeniu modelu [12 23].
Wartości prawdopodobieństw warunkowych, dla modelu [12 3] są takie same dla liczności wynikających z danych i liczności wynikających ze zbudowanego modelu, w przypadku wyliczania prawdopodobieństwa, że student dobrze śpi, jeśli regularnie biega, natomiast w przypadku wyznaczania drugiego zadanego prawdopodobieństwa, otrzymałem rozbieżność w wynikach. Dla przyjętego modelu [12 23] uzyskane wartości (z liczności danych i z liczności wynikających z modelu), dla obydwu rozważanych prawdopodobieństw warunkowych, są dokładnie takie same.

W kolejnym zadaniu należało zweryfikować hipotezy:

(a) Zmienne losowe *Sen*, *Bieganie* i *Pies* są wzajemnie niezależne
(b) Zmienna losowa *Pies* jest niezależna od pary zmiennych *Sen* i *Bieganie*
(c) Zmienna losowa *sen* jest niezależna od zmiennej *Pies*, przy ustalonej zmiennej *Bieganie*.

Aby rozwiązać to zadanie, należało zbudować dobry model log-liniowy i przeprowadzić test statystyczny, w którym jako hipotezę alternatywną należało wziąć nadmodel zbudowanego modelu, który nie jest modelem pełnym i model pełny.
Interpretując uzyskane wyniki, można założyć, że:

- Zmienne losowe *SEN*, *BIEGANIE* i *PIES* nie są wzajemnie niezależne,
- Zmienna losowa *Pies* jest niezależna od pary zmiennych *Sen* i *Bieganie*,
- Zmienna losowa *Sen* jest niezależna od zmiennej *Pies*, przy ustalonej zmiennej *Bieganie*.

W ostatnim zadaniu należało dokonać wyboru modelu log-liniowego w oparciu o testy, kryterium AIC i kryterium BIC.
Biorąc pod uwagę testy dostałem, że najlepszym modelem będzie model [12 23]. W oparciu o kryteria AIC i BIC również mogłem wyciągnąć wniosek, że model [12 23] jest najlepszy z moż-

liwych. W przypadku kryteriów informacyjnych, dobór najlepszego modelu wykonałem poprzez dwa podejścia — wyznaczając minimalną wartość kryteriów z wszystkich możliwych 19 modeli oraz korzystając z wbudowanej w pakiet R funkcji `step` Wartości kryteriów informacyjnych dla każdego modelu umieściłem w tabeli (6).