Politechnika Wrocławska
Wydział Matematyki

Skład grupy:	Agata Sobczak 268873
	Jakub Franczak 262271
	Katarzyna Kudełko 268762
Prowadząca laboratorium:	dr inż. Aleksandra Grzesiek
Prowadząca wykładu:	dr hab. Alicja Jokiel-Rokita

Analiza Danych Ankietowych

Raport 1.

Lista 1.

WROCŁAW 13.11.2023

Spis treści

1 Zadanie 1.

1.1 Cel zadania

Celem zadania jest zaproponować i opisać badanie ankietowe dotyczące wybranego tematu.

Badanie ankietowe dotyczące ogólnego zadowolenia studentów I-stopnia Politechniki Wrocławskiej z wybranego kierunku na wydziale W13. Badania zakładają wybranie losowo (ze zwracaniem) 100 studentów wydziału W13. Poniżej zostały przedstawione zostały pytania ankietowe.

Płeć:	
Kobieta	
Mężczyzna	

Wiek:	
do 19 lat	
20-23 lata	
24-26 lata	
powyżej 27 lat	

Kierunek:	
Matematyka	
Matematyka stosowana	
Matematyka i analiza danych	

Semestr:	
1 semestr	
2 semestr	
3 semestr	
4 semestr	
5 semestr	
6 semestr	
7 semestr	

Zaznaczyć zgodnie z prawdą:

	-2	-1	0	1	2
Spędzam za dużo czasu na uczelni.					
Uczenie się sprawia mi przyjemność.					
Uważam swój kierunek za niewymagający.					
Uczelnia zapewnia mi warunki do nauki.					
Moje oceny są adekwatne do wkładu pracy.					
Nie mam czasu wolnego.					
Uważam, że po ukończeniu mojego kierunku z łatwością znajdę pracę.					
Gdybym mogła/mógł wybrałabym/wybrałbym inny kierunek.					
Uważam, że uczę się wartościowych i przydatnych rzeczy na zajęciach.					
Jestem zadowolony/a z wybranego przeze mnie kierunku.					

Gdzie:	
-2	Zdecydowanie się nie zgadzam.
-1	Nie zgadzam się.
0	Nie mam zdania.
1	Zgadzam się.
2	Zdecydowanie się zgadzam.

2 Zadanie 2.

2.1 Cel zadania

Celem zadania jest analiza danych zawartych w pliku Choroba.csv, które obejmują informacje o 196 osobach losowo wybranych z dwóch sektorów miasta. Informacje obejmują:

- wiek,
- status ekonomiczny (1 – wysoki, 2 – średni, 3 – niski),
- sektor (1– osoba mieszka w sektorze 1, 2 – w sektorze 2),
- oszczędności (1 – posiada oszczędności, 0 – nie posiada oszczędności),
- zdrowie (1 - chora, 0 - zdrowa).

```
# Wczytywanie danych z pliku "Choroba.csv"
data <- read.csv("Choroba.csv",sep = ";")
```

	WIEK	STATUS	SEKTOR	CHORY_ZD	OSZCZED
1	33	1	1	0	1
2	35	1	1	0	1
3	6	1	1	0	0
4	60	1	1	0	1
5	18	3	1	1	0
6	26	3	1	0	0

Rysunek 1: head(data)

2.2 a)

Tablice liczności dla zmiennych *Oszczed* biorące pod uwagę wszystkie dane.

```
# Tabele liczności dla zmiennej OSZCZED

table(data$OSZCZED, data$WIEK)
table(data$OSZCZED, data$STATUS)
table(data$OSZCZED, data$SEKTOR)
table(data$OSZCZED, data$CHORY_ZD)
```

WIEK	OSZCZED	
	0	1
1	1	3
2	4	2
3	0	4
4	1	2
5	4	0
6	5	5
7	3	1
8	4	3
9	5	1
10	0	1
11	4	2
12	0	4
13	2	3
14	3	1
15	3	4
16	4	2
17	3	1
18	5	1
19	1	0
20	1	2
21	1	3
22	2	1
23	2	1
24	3	2
25	1	3
26	2	1
27	3	4
28	1	1
29	2	0
30	2	2
31	1	3

WIEK	OSZCZED	
	0	1
32	2	3
33	1	2
34	0	2
35	3	2
37	2	1
38	0	2
39	2	2
40	1	0
42	1	2
43	0	1
44	1	0
46	1	2
48	1	1
50	0	2
51	0	1
52	0	1
53	1	1
56	0	1
59	0	1
60	0	2
61	0	2
64	1	0
65	0	3
67	0	1
68	0	1
69	0	1
70	0	2
73	0	1
74	0	1
79	0	1
85	0	1

Tabela 1: Tabela OSZCZED ze względu na WIEK

STATUS	OSZCZED	
	0	1
1	17	60
2	25	24
3	48	22

Tabela 2: Tabela OSZCZED ze względu na STATUS

SEKTOR	OSZCZED	
	0	1
1	65	25
2	25	54

Tabela 3: Tabela OSZCZED ze względu na SEKTOR

CHORY/ZD	OSZCZED	
[redacted]	0	1
0	69	71
1	21	35

Tabela 4: Tabela OSZCZED ze względu na CHORY/ZD

Tablice liczności dla zmiennych *Chory/Zd* biorące pod uwagę wszystkie dane.

```
# Tabele liczności dla zmiennej CHORY_ZD

table(data$CHORY_ZD, data$WIEK)
table(data$CHORY_ZD, data$STATUS)
table(data$CHORY_ZD, data$SEKTOR)
table(data$CHORY_ZD, data$OSZCZED)
```

STATUS	CHORY/ZD	
[redacted]	0	1
1	53	24
2	36	13
3	51	19

Tabela 5: Tabela CHORY/ZD ze względu na STATUS

SEKTOR	CHORY/ZD	
[redacted]	0	1
1	95	22
2	45	34

Tabela 6: Tabela CHORY/ZD ze względu na SEKTOR

OSZCZED	CHORY/ZD	
[redacted]	0	1
0	69	71
1	21	35

Tabela 7: Tabela CHORY/ZD ze względu na OSZCZED

WIEK	CHORY/ZD	
	0	1
1	4	0
2	6	0
3	4	0
4	3	0
5	4	0
6	9	1
7	4	0
8	6	1
9	5	1
10	1	0
11	4	2
12	4	0
13	2	3
14	3	1
15	5	2
16	4	2
17	2	2
18	5	1
19	1	0
20	1	2
21	2	2
22	1	2
23	3	0
24	4	1
25	3	1
26	2	1
27	5	2
28	2	0
29	0	2
30	4	0
31	3	1

WIEK	CHORY/ZD	
	0	1
32	2	3
33	2	1
34	1	1
35	3	2
37	2	1
38	1	1
39	2	2
40	0	1
42	2	1
43	0	1
44	1	0
46	2	1
48	1	1
50	2	0
51	1	0
52	0	1
53	1	1
56	1	0
59	0	1
60	1	1
61	2	0
64	1	0
65	1	2
67	0	1
68	0	1
69	1	0
70	1	1
73	1	0
74	0	1
79	1	0
85	1	0

Tabela 8: Tabela CHORY/ZD ze względu na WIEK

2.3 b)

Tabela wielodzielcza uwzględniająca zmienną *Chory/Zdrowy* i *Sektor*

```
# Tabela wielodzielcza dla  CHORY_ZD i SEKTOR
addmargins(ftable(data, col.vars = "CHORY_ZD", row.vars = "SEKTOR"))
```

STATUS	CHORY/ZD		SUMA
	0	1	
1	53	24	117
2	36	13	49
3	51	19	70
SUMA	140	56	196

Tabela 9: Tabela wielodzielcza *Chory/Zdrowy* i *Sektor*

2.4 c)

Tabela wielodzielcza uwzględniająca zmienną *Chory/Zdrowy* i *Status*

```
# Tabela wielodzielcza dla  CHORY_ZD i STATUS
addmargins(ftable(data, col.vars = "CHORY_ZD", row.vars = "STATUS"))
```

STATUS	CHORY/ZD		SUMA
	0	1	
1	53	24	117
2	36	13	49
3	51	19	70
SUMA	140	56	196

Tabela 10: Tabela wielodzielcza *Chory/Zdrowy* i *Status*

2.5 d)

Kategoryzacja zmiennej *Wiek*

```
# Kategoryzacja zmiennej WIEK
data$WIEK_KAT <- cut(data$WIEK, breaks = c(0,20,40,60,80,Inf),
                     labels = c("0-20","21-40","41-60","61-80","81+"))
```

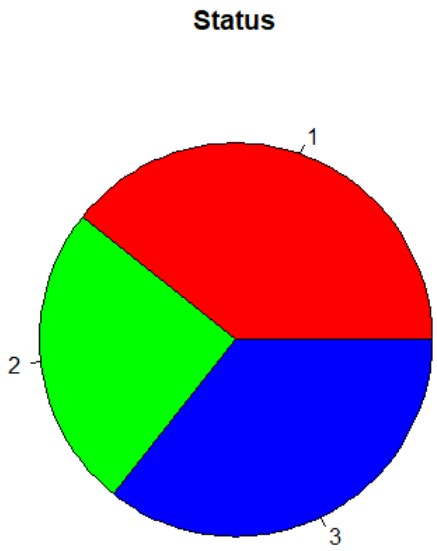
```
  WIEK STATUS SEKTOR CHORY_ZD OSZCZED WIEK_KAT
1   33      1      1        0       1    21-40
2   35      1      1        0       1    21-40
3    6      1      1        0       0     0-20
4   60      1      1        0       1    41-60
5   18      3      1        1       0     0-20
6   26      3      1        0       0    21-40
```

Rysunek 2: head(data) po kategoryzacji zmiennej *Wiek*

2.6 e)

Wykres kołowy dla zmiennej *Status*

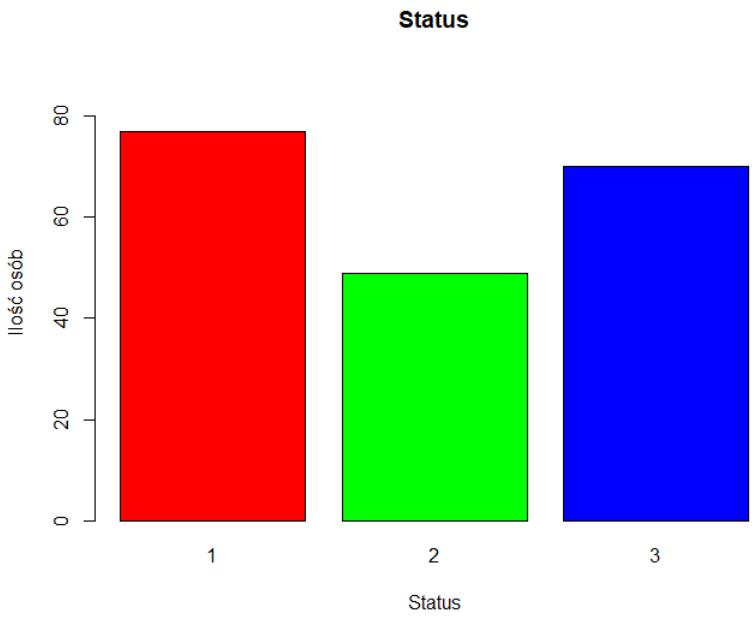
```
# wykres kołowy
data_status <- table(data$STATUS)
pie(data_status, main = "Status", col = c('red', 'green', 'blue'))
```



Rysunek 3: Wykres kołowy dla zmiennej *Status*

Wykres słupkowy dla zmiennej *Status*

```
# wykres słupkowy
barplot(data_status, main="Status", xlab = "Status", ylab = "Ilość osób",
        ylim = c(0, 90), col = c('red', 'green', 'blue'))
```

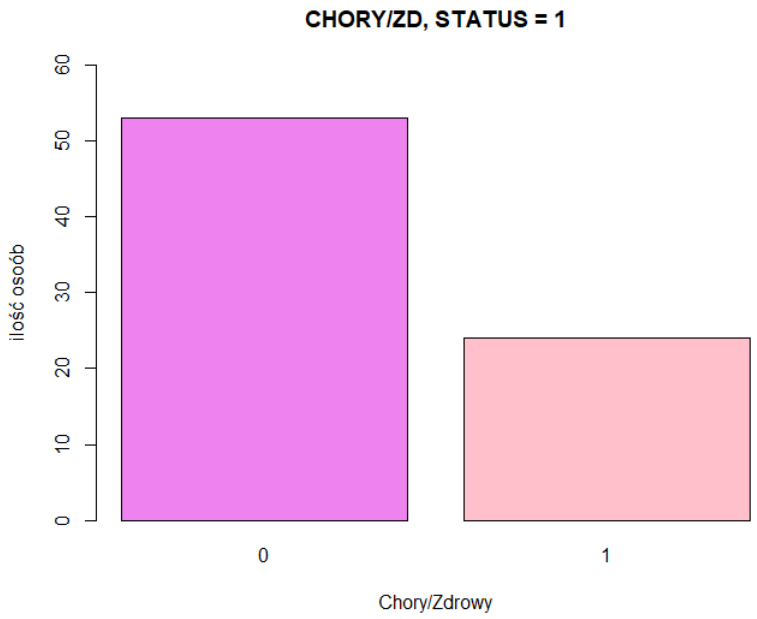


Rysunek 4: Wykres słupkowy dla zmiennej *Status*

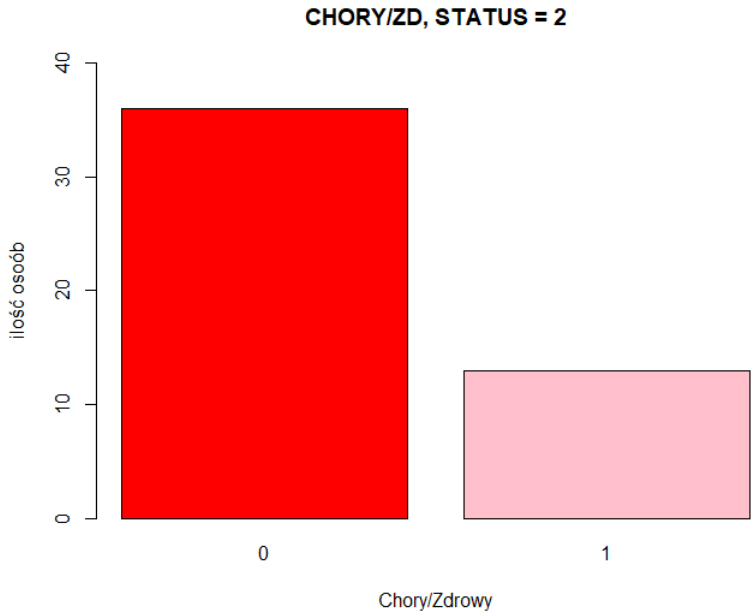
2.7 f)

Skategoryzowane wykresy zmiennej *Chory/Zdrowy* przyjmując za zmienną kategoryzującą zmienną *Sektor*

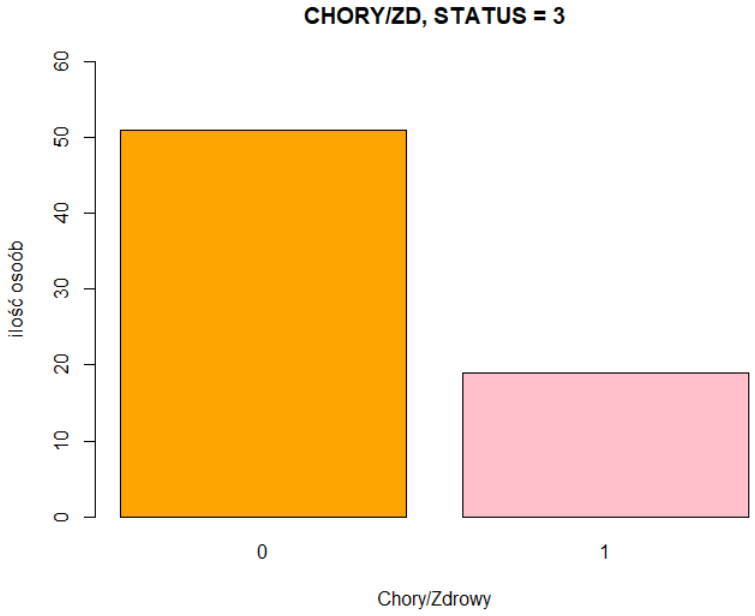
```
data$STATUS == 1
data$CHORY_ZD[data$STATUS == 1]
data1 <- table(data$CHORY_ZD[data$STATUS == 1])
barplot(data1, main = "CHORY/ZD, STATUS = 1",  xlab = "Chory/Zdrowy",
        ylab = "ilość osoób", ylim = c(0, 60), col = c("violet","pink"))

data$STATUS == 2
data2 <- table(data$CHORY_ZD[data$STATUS == 2])
barplot(data2, main = "CHORY/ZD, STATUS = 2",  xlab = "Chory/Zdrowy",
        ylab = "ilość osoób", ylim = c(0, 40), col = c("red","pink"))

data$STATUS == 3
data3 <- table(data$CHORY_ZD[data$STATUS == 3])
barplot(data3, main = "CHORY/ZD, STATUS = 3",  xlab = "Chory/Zdrowy",
        ylab = "ilość osoób", ylim = c(0, 60), col = c("orange","pink"))
```



Rysunek 5: Wykres skategoryzwanej zmiennej *Chory/Zdrowy* z zmienną kategoryzującą *Sektor* = 1



Rysunek 6: Wykres skategoryzwanej zmiennej *Chory/Zdrowy* z zmienną kategoryzującą *Sektor* = 2

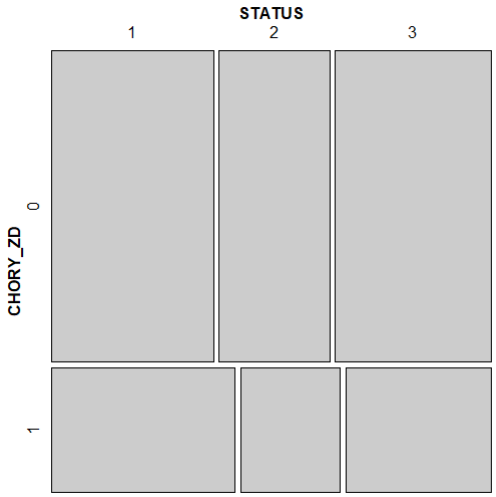


Rysunek 7: Wykres skategoryzwanej zmiennej *Chory/Zdrowy* z zmienną kategoryzującą *Sektor* = 3

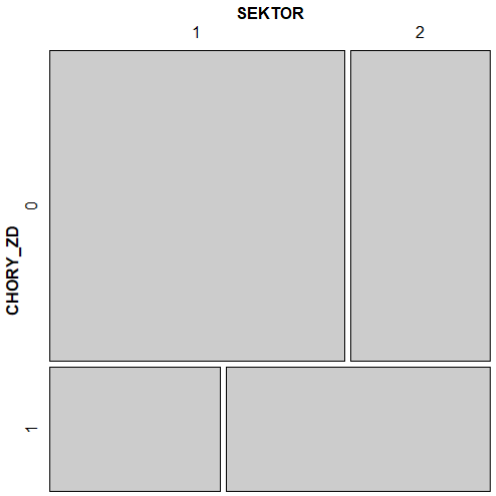
2.8 g)

Wykresy mozaikowe odpowiadające zmiennym *Chory/Zd* i *Status*.

```
# Wykresy mozaikowe odpowiadajace zmiennym CHORY/ZD i SEKTOR

mosaic(~CHORY_ZD + STATUS, data)
mosaic(~CHORY_ZD + SEKTOR, data)
```



Rysunek 8: Wykres mozaikowy dla pary zmiennych *Chory/Zd* i *Status*



Rysunek 9: Wykres mozaikowy dla pary zmiennych *Chory/Zd* i *Sektor*

Jak widać na wykresie mozaikowym, więcej osób zdrowych zamieszkuje sektor 1, a ilość osób o statusie wysokim i niskim jest zdrowe.

3 Zadanie 3.

3.1 Cel zadania

Celem zadania jest napisanie fragmentu programu, który ma na celu wylosowanie próbki danych z pewnej hipotetycznej bazy danych (np. z danych zawartych w zadaniu 2) o rozmiarze około 1/10 liczby przypadków w bazie. Program ma uwzględniać możliwość losowania próbki zarówno ze zwracaniem jak i bez zwracania.

```
# Losowanie próbki danych z zad2 o rozmiarze ok. 1/10 liczby przypadków w bazie

ind <- sample(nrow(data),1/10 * nrow(data),replace = FALSE)
data[ind,]

ind2 <- sample(nrow(data),1/10*nrow(data),replace = TRUE)
data[ind2,]
```

	WIEK	STATUS	SEKTOR	CHORY_ZD	OSZCZED
125	15	3	2	0	0
94	9	2	1	0	0
133	12	2	1	0	1
121	33	2	2	0	1
37	3	2	2	0	1
188	53	3	1	0	0
70	28	1	1	0	0
49	40	2	2	1	0
186	16	2	1	0	0
83	65	3	1	1	1
35	44	3	2	0	0
157	79	1	2	0	1
3	6	1	1	0	0
178	13	2	1	0	1
144	17	1	2	0	1
149	13	1	2	1	1
115	15	1	2	0	1
159	8	1	2	0	1
135	15	2	2	0	1

Rysunek 10: Próbka danych z zad.2 o rozmiarze ok. 1/10 bez zwracania

	WIEK	STATUS	SEKTOR	CHORY_ZD	OSZCZED
159	8	1	2	0	1
7	6	3	1	0	0
63	2	3	1	0	1
1	33	1	1	0	1
83	65	3	1	1	1
137	2	2	2	0	1
19	6	1	2	1	1
2	35	1	1	0	1
124	5	3	2	0	0
107	27	2	1	0	0
44	70	1	2	1	1
171	42	1	1	0	1
2.1	35	1	1	0	1
28	2	3	1	0	0
92	4	3	1	0	0
39	17	2	2	1	0
157	79	1	2	0	1
43	24	1	2	0	0
61	18	3	1	0	0

Rysunek 11: Próbka danych z zad.2 o rozmiarze ok. 1/10 ze zwracaniem

4 Zadanie 4.

4.1 Cel zadania

Celem zadania jest przeprowadzenie symulacji mających na celu porównanie prawdopodobieństwa pokrycia i długości przedziałów ufności dla trzech różnych metod:

- Cloppera-Pearsona,
- Walda,
- trzeciego dowolnego typu przedziału ufności - wykorzystano metodę Bayesa.

Analizę wykonano dla różnych rozmiarów próby ($n \in 30, 100, 1000$) oraz różnych wartości prawdopodobieństwa sukcesu ($p \in 0.1, 0.5, 0.8$), przy założonym poziomie ufności 0.95.

4.2 Opis programu

Program ma na celu sprawdzenie pokrycia oraz długości przedziałów ufności dla rozkładu dwumianowego przy różnych rozmiarach próbki i prawdopodobieństwach sukcesu. Wykorzystuje symulację Monte Carlo (ustalono $MC = 50$ aby przyspieszyć obliczenia), aby generować losowe próbki dla każdego p i obliczać przedziały ufności za pomocą trzech różnych metod: Cloppera-Pearsona (exact), Walda (asymptotic) i Bayesa (bayes). Dla każdej metody oblicza pokrycie oraz średnią długość przedziałów. Program korzysta z pakietu ggplot2, aby stworzyć wykresy liniowe dla pokrycia oraz długości przedziałów ufności. Pionowymi liniami oznaczono $p = 0.3$, $p = 0.5$ i $p = 0.8$.

4.3 Kod

```
MC <- 50
methods <- c("exact", "asymptotic", "bayes")
DF <- data.frame(Method = character(),
                 N = numeric(),
                 P = numeric(),
                 Coverage = numeric(),
                 Length = numeric(),
                 stringsAsFactors = FALSE)
```

Rysunek 12: Przypisanie zmiennych oraz inicjalizacja ramki zawierającej wynik

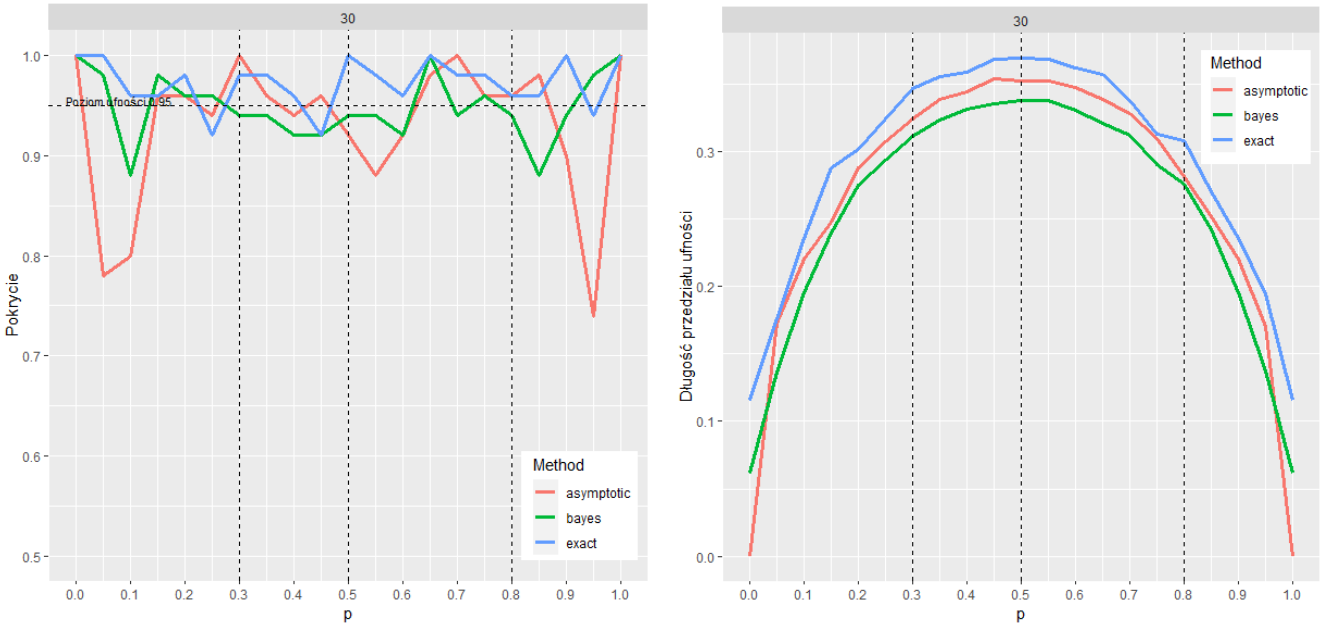
```r
for (i in p) {
  for (method in methods) {
    coverage <- numeric()
    ci_length <- numeric()
    for (step in 1:MC) {

      x <- rbinom(MC, n, i)
      ci <- binom.confint(x, n = n, conf.level = 0.95, method = method) #confidence interval

      check <- (ci$lower <= i) & (i <= ci$upper) #check if in ci

      if (any(check)) {
        ci_length[step] <- ci$upper[check] - ci$lower[check]
        coverage[step] <- check
      }
    }

    df <- data.frame(
      Method = method,
      N = n,
      P = i,
      Coverage = sum(coverage)/MC,
      Length = mean(ci_length),
      stringsAsFactors = FALSE
    )
    DF <- rbind(DF, df)
  }
}
```

Rysunek 13: Główna pętla obliczeniowa

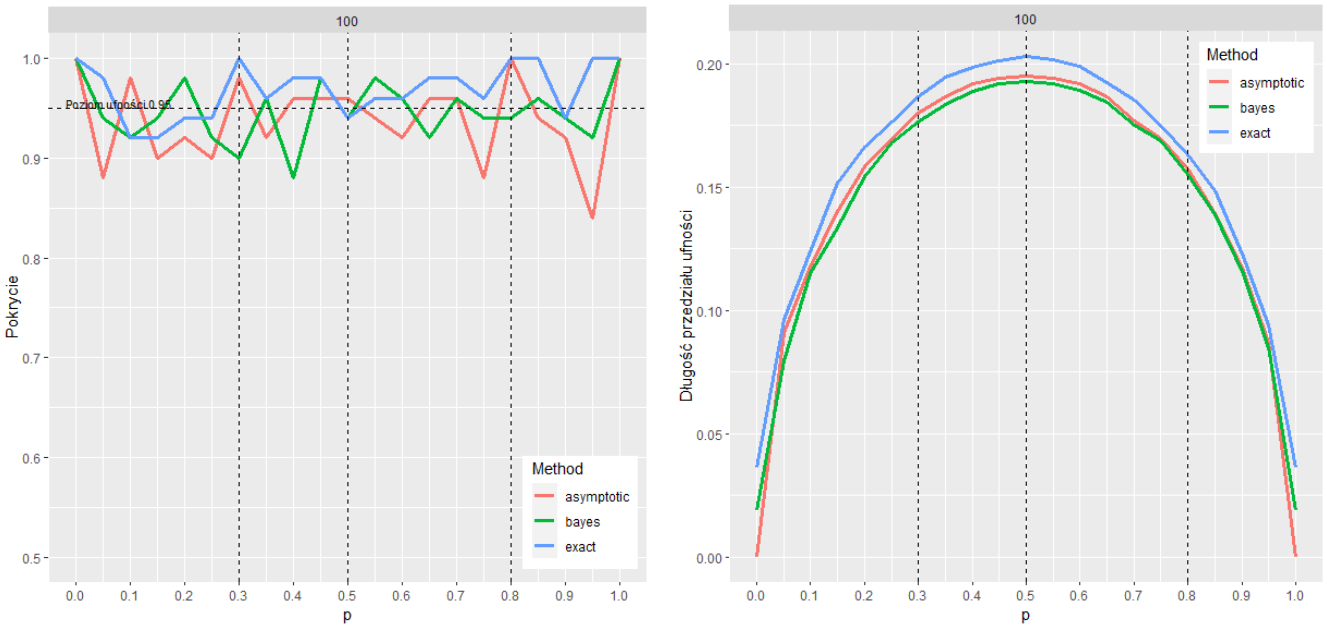
```r
ggplot(DF, aes(x = P, y = Coverage, group = Method, color = Method)) +
  geom_line(size = 1.2) +
  facet_wrap(~N) +
  geom_hline(yintercept = 0.95, linetype = "dashed") +
  geom_hline(yintercept = 0.5, linetype = "dashed", color = "NA") +
  geom_vline(xintercept = c(0.3, 0.5, 0.8), linetype = "dashed") +
  scale_x_continuous(breaks = seq(0, 1, 0.1)) +
  labs(x = "p", y = "Pokrycie") +
  annotate("text", x = 0.08, y = 0.955, label = "Poziom ufności 0.95", size = 3)

ggplot(DF, aes(x = P, y = Length, group = Method, color = Method)) +
  geom_line(size = 1.2) +
  facet_wrap(~N) +
  geom_vline(xintercept = c(0.3, 0.5, 0.8), linetype = "dashed") +
  scale_x_continuous(breaks = seq(0, 1, 0.1)) +
  labs(x = "p", y = "Długość przedziału ufności")
```

Rysunek 14: Rysowanie wykresów

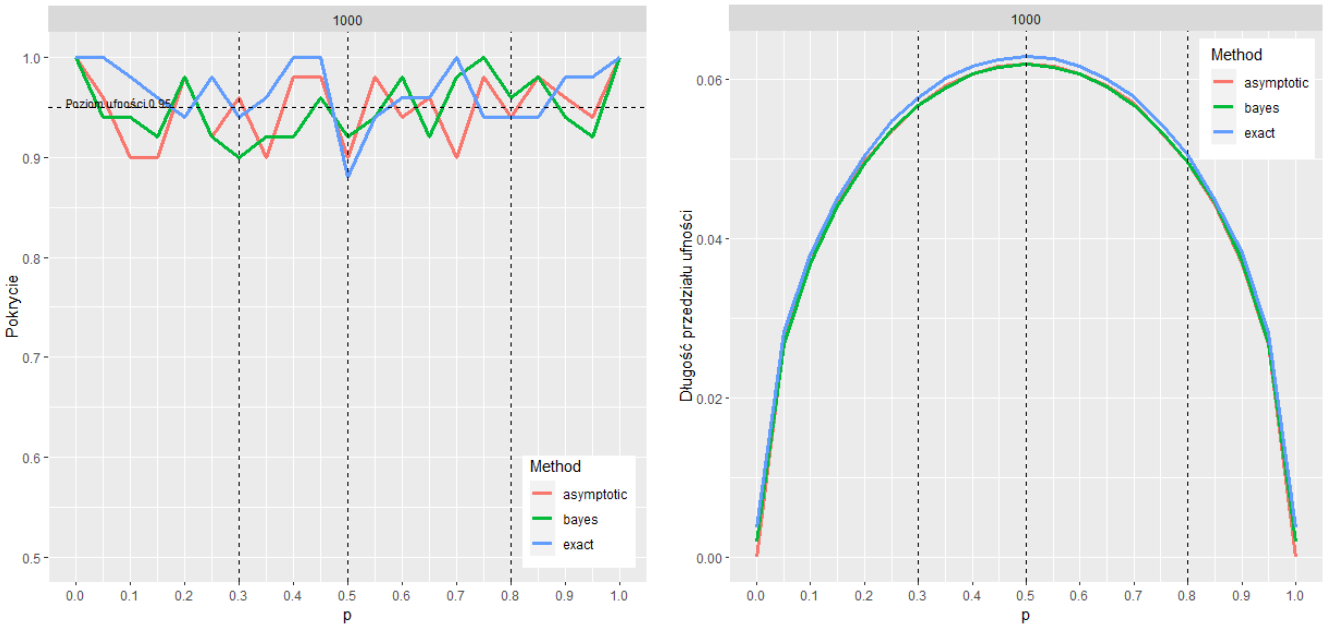
4.4 Wyniki



Rysunek 15: Wykresy pokrycia oraz długości przedziałów ufności dla $n = 30$



Rysunek 16: Wykresy pokrycia oraz długości przedziałów ufności dla $n = 100$



Rysunek 17: Wykresy pokrycia oraz długości przedziałów ufności dla $n = 1000$

	Method	N	P	Coverage	Length
1	exact	30	0.3	0.96	0.3470207
2	asymptotic	30	0.3	0.94	0.3227073
3	bayes	30	0.3	0.92	0.3049715
4	exact	30	0.5	1.00	0.3680739
5	asymptotic	30	0.5	0.98	0.3522531
6	bayes	30	0.5	0.96	0.3392330
7	exact	30	0.8	1.00	0.3025883
9	bayes	30	0.8	0.98	0.2791208
8	asymptotic	30	0.8	0.98	0.2764228

Tabela 11: Tabela wyników dla $n = 30$

Dla pozostałych n tabele prezentują się analogicznie.

4.5 Wnioski

Analizując przedstawione wykresy prawdopodobieństwa pokrycia przedziałów ufności, możemy wyciągnąć wniosek, że wraz ze wzrostem rozmiaru próbki n, przedziały są bardziej wiarygodne i częściej pokrywają się z założonym poziomem ufności 0.95. Pomimo tego ogólnego trendu, różnice między zastosowanymi metodami są zauważalne. Metoda Walda wykazuje tendencję do gorszego przybliżania założonego poziomu ufności w porównaniu do innych metod. Metoda Cloppera-Pearsona prezentuje się najlepiej, zdecydowana większość jej wyników znajduje się powyżej poziomu 0.95.

Przechodząc do analizy wykresów średniej długości przedziałów ufności, obserwujemy podobne zachowanie. Wraz ze wzrostem rozmiaru próbki n, rozproszenie wyników maleje. Dla skrajnych wartości prawdopodobieństwa uzyskujemy najkrótsze przedziały ufności. Metoda Cloppera-Pearsona generuje najdłuższe przedziały, a pozostałe metody prezentują krótsze, lecz zbliżone długości przedziałów.

W zależności od tego czy chcemy uzyskać najkrótsze przedziały ufności czy największy procent pokrycia, wybór metody może się różnić. Bardzo korzystnie w tym zestawieniu prezentuje się metoda Cloppera-Pearsona -

patrząc na wykresy pokrycia, zachowuje się bardziej regularnie niż pozostałe metody, zwłaszcza przy skrajnych p.

5 Zadanie 5.

5.1 Cel zadania

Celem zadania jest wyznaczenie realizacji przedziałów ufności na poziomie ufności 0.95 dla prawdopodobieństwa, że losowo wybrana osoba z badanej populacji jest chora, na podstawie danych zawartych w pliku Choroba.csv. Aby to zrobić, należy skorzystać z funkcji binom.confint w języku R, która umożliwia wyznaczenie przedziałów ufności dla rozkładu dwumianowego. W ramach zadania należy obliczyć i porównać realizacje różnych typów przedziałów ufności oraz przeanalizować ich długości, a następnie wybrać "najlepszy"z nich.

5.2 Kod

Fragment kodu realizujący podane zadanie przedstawiono poniżej.

```
conf_intervals <- binom.confint(x = liczba_chorych, n = 196, conf.level = 0.95, methods = "all")
interval_lengths <- numeric(length(conf_intervals))

for (i in 1:nrow(conf_intervals)) {
  interval_lengths[i] <- conf_intervals$upper[i] - conf_intervals$lower[i]
}

result <- data.frame(Method = conf_intervals[1], ci_length = interval_lengths)
```

Rysunek 18: Wycinek kodu dla zad. 5

Realizacja wykresu:

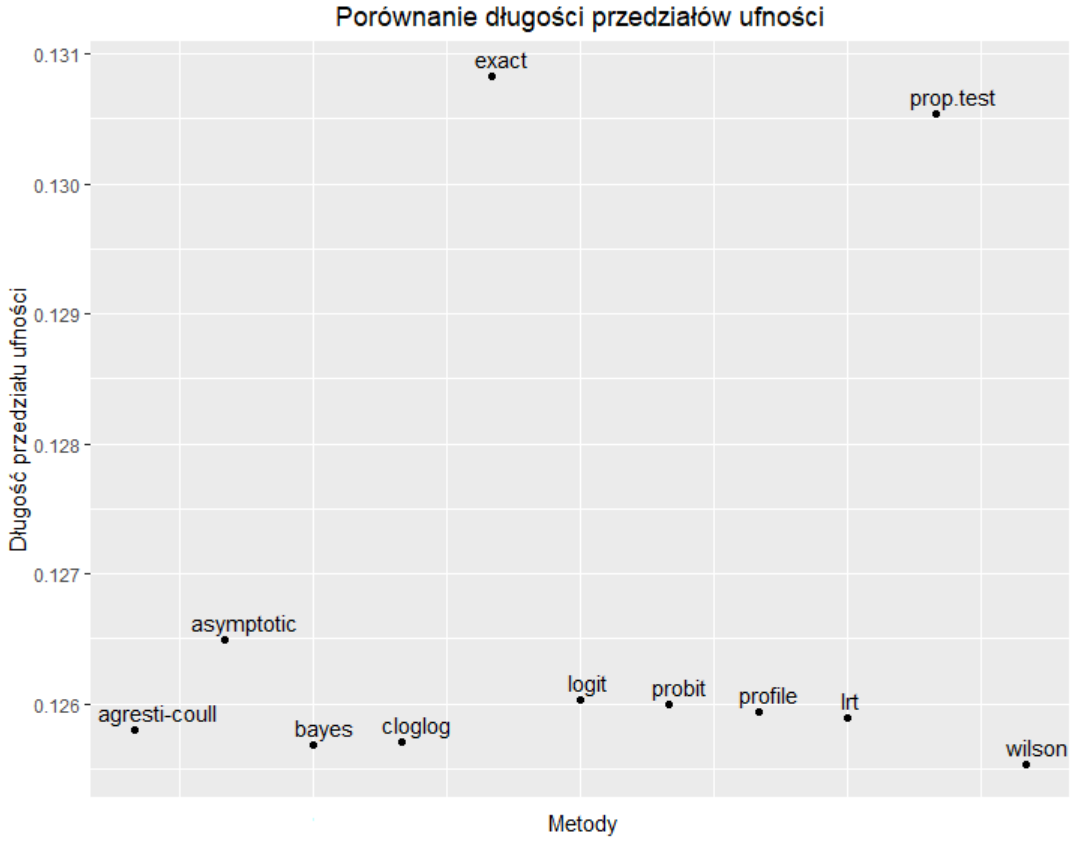
```
ggplot(result, aes(x = 1:11, y = ci_length, label = names)) +
  geom_point() +
  geom_text(hjust = 0.3, vjust = -0.5) +
  labs(title = "Porównanie długości przedziałów ufności",
       x = "Metody",
       y = "Długość przedziału ufności") +
  theme(axis.text.x = element_text(angle = 1, hjust = 1),
        plot.title = element_text(hjust = 0.5))
```

Rysunek 19: Rysowanie wykresów

5.3 Wyniki

Metoda	Długość
agresti-coull	0.1258026
asymptotic	0.1264888
bayes	0.1256781
cloglog	0.1257106
exact	0.1308311
logit	0.1260331
probit	0.1259942
profile	0.1259332
lrt	0.1258865
prop.test	0.1305394
wilson	0.1255378

Tabela 12: Tabela metod oraz ich długości wygenerowanych przedziałów ufności



Rysunek 20: Wykres metod oraz ich długości wygenerowanych przedziałów ufności

5.4 Wnioski

Patrząc na tabelę oraz wykres, można łatwo zauważyć, że metoda Wilsona (wilson) generuje najkrótsze przedziały ufności. Jeśli priorytetem jest mniejsza długość przedziałów przy jednoczesnym utrzymaniu odpowiedniego poziomu ufności, dobrym wyborem jest metoda Wilsona. Najdłuższe generuje natomiast metoda Clopper-Pearsona (exact), co może być korzystne w kontekście większej ostrożności przy oszacowywaniu parametru, a także zgadza się to z wnioskami z poprzedniego zadania.

6 Zadanie 7.

6.1 Cel zadania

Celem zadania jest zweryfikowanie następujących hipotez na podstawie danych zawartych w pliku Choroba.csv na poziomie istotności 0.05 oraz podanie wartości poziomu krytycznego (p-value):

- prawdopodobieństwo, że losowo wybrana osoba z badanej populacji jest chora jest mniejsze bądź równe 1/2,
- prawdopodobieństwo, że losowo wybrana osoba z sektora 1. jest chora jest równe prawdopodobieństwu, że losowo wybrana osoba z sektora 2. jest chora,
- powtórzyć wcześniejsze punkty, ale dla osoby o średnim statusie ekonomicznym.

6.2 a)

```
n <- length(choroba$CHORY_ZD)
test1 <-test1 <- binom.test(sum(choroba$CHORY_ZD), n, p=0.5, alternative= "g", conf.level=0.95)
test2 <- prop.test(sum(choroba$CHORY_ZD) ,n , alternative = "g", conf.level=0.95 , correct = TRUE)
test3 <- prop.test(sum(choroba$CHORY_ZD) ,n , alternative = "g", conf.level=0.95 ,  correct = FALSE)

print(test1) #p_value = 1
print(test2) #p_value = 1
print(test3) #p_value = 1
```

Rysunek 21: Kod do weryfikacji hipotezy w podpunkcie a.

6.2.1 Analiza wyników

Jak możemy zauważyć, p-wartości przeprowadzonych testów są równe 1, więc należy przyjąć hipoteze zerową - prawdopodobieństwo wybrania losowo osoby chorej z populacji danej w pliku jest mniejsze bądź równe 1/2.

6.3 b)

```r
#7b

dane1 <- choroba$CHORY_ZD[choroba$SEKTOR == 1]
n1 <- length((dane1))
ilosc_chorych_sektor_1 <- sum(dane1)


dane2 <- choroba$CHORY_ZD[choroba$SEKTOR == 2]
n2 <- length((dane2))
ilosc_chorych_sektor_2 <- sum(dane2)

x <- c(ilosc_chorych_sektor_1 , ilosc_chorych_sektor_2)
n <- c(n1 , n2)
test1 <- prop.test(x ,n , alternative = "t", correct = FALSE)
test2 <- prop.test(x ,n , alternative = "t",  correct = TRUE)

print(test1) #p_value=0.0002297
print(test2) #p_value=0.0004271
```

Rysunek 22: Kod do weryfikacji hipotezy w podpunkcie b.

6.3.1 Analiza wyników testów

Przeprowadzone testy asymptotyczne mają małe p-wartości (mniejsze od poziomu istotności), więc należy odrzucić hipotezę zerową - prawdopodobieństwo wyboru osoby chorej jest rożne w zależności od sektora.

6.4 c)

```r
#7c

x <- 12 #chorzy sredni status
n <- 49 #wszyscy sredni

test1 <- binom.test(x, n, p=0.5, alternative= "g", conf.level=0.95)
test2 <- prop.test(x, n, p=0.5, alternative= "g", conf.level=0.95, correct = TRUE)
test3 <- prop.test(x, n, p=0.5, alternative= "g", conf.level=0.95, correct = FALSE)

print(test1) #p_value = 0.9999
print(test2) #p_value = 0.9997
print(test3) #p_value = 0.9998


x <- c(3,10) #(liczba chorych w 1 sektorze o srednim statusie,
             #liczba chorych w 2 sektorze o srednim statusie)
n <- c(26,23) #(liczba osob w 1 sektorze o srednim statusie,
              #liczba osob w 2 sektorze o srednim statusie)

test1 <- prop.test(x, n, alternative="t", correct = FALSE)
test2 <- prop.test(x, n, alternative="t", correct = TRUE)

print(test1) #p_value = 0.01149
print(test2) #p_value = 0.02759
```

Rysunek 23: Kod do weryfikacji hipotez w podpunkcie c.

6.4.1 Analiza wyników testów

W przypadku ponownej analizy podpunktu a) dla osób o statucie średnim możemy zauważyć, że p-wartości nie są równe 1, ale są bardzo blisko tej wartości, więc ponownie przyjmujemy hipotezę zerową. W przypadku podpunktu b) ponownie otrzymaliśmy p-wartości poniżej poziomu istotności, więc należy w takim wypadku odrzucić hipotezę zerową.

7 Zadanie 8.

7.1 Cel zadania

Celem zadania jest przeprowadzenie symulacji mających na celu:

- Oszacowanie prawdopodobieństwa błędu I-go rodzaju dla testu dokładnego i testu asymptotycznego przy przyjętym poziomie istotności 0.05.
- Porównanie mocy testu dokładnego i testu asymptotycznego dla różnych wartości alternatywnych hipotez ($\theta \in 0.3, 0.4, 0.6$),

w przypadku weryfikacji hipotezy zerowej $H_0 : \theta = 0.5$ i hipotezy alternatywnej $H_1 : \theta \neq 0.5$. Analizy przeprowadzone są dla różnych rozmiarów próby $n \in \{30, 100, 1000\}$, a wyniki uzyskane z symulacji zestawione w tabelach i przedstawione na wykresach.

7.2 a)

```
for (i in 1:length(wartosci_n)) {
  n <- wartosci_n[i]
  odrzuc_binom <- 0
  odrzuc_prop_1 <- 0
  odrzuc_prop_2 <- 0

  for (j in 1:N) {
    data <- rbinom(n, 1, p0)

    binom_test <- binom.test(sum(data), n, p = p0, alternative = "t")
    prop_test_1 <- prop.test(sum(data), n, p = p0, alternative = "t", correct = TRUE)
    prop_test_2 <- prop.test(sum(data), n, p = p0, alternative = "t", correct = FALSE)


    if (binom_test$p.value < alpha) {
      odrzuc_binom <- odrzuc_binom + 1
    }
    if (prop_test_1$p.value < alpha) {
      odrzuc_prop_1 <- odrzuc_prop_1 + 1
    }
    if (prop_test_2$p.value < alpha) {
      odrzuc_prop_2 <- odrzuc_prop_2 + 1
    }
  }

  blad_I_rodzaju_binom <- odrzuc_binom / N
  blad_I_rodzaju_prop_1 <- odrzuc_prop_1 / N
  blad_I_rodzaju_prop_2 <- odrzuc_prop_2 / N

  macierz_wyniki[i, 1] <- blad_I_rodzaju_binom
  macierz_wyniki[i, 2] <- blad_I_rodzaju_prop_1
  macierz_wyniki[i, 3] <- blad_I_rodzaju_prop_2
}
```

Rysunek 24: Główne obliczenia

```r
N <- 1000
alpha <- 0.05
p0 <- 0.5
wartosci_n <- c(30, 100, 1000)

macierz_wyniki <- matrix(NA, nrow = 3, ncol = length(wartosci_n))
rownames(macierz_wyniki) <- c("binom_test", "prop_test_correct", "prop_test_uncorrected")
colnames(macierz_wyniki) <- as.character(wartosci_n)
```

Rysunek 25: Przypisanie wartości

7.2.1 Wyniki

Powyższy kod zwraca prawdopodobieństwo popełnienia błędu I rodzaju dla różnych rozmiarów prób.

	30	**100**	**1000**
Binom	0.038	0.038	0.038
Prop test correct	0.032	0.032	0.059
Prop test uncorrected	0.050	0.050	0.057

Tabela 13: Prawdopodobieństwo popełnienia błędu I rodzaju dla różnych rozmiarów prób n $\in (30, 100, 1000)$.

7.3 b)

Poniżej zaprezentowano kod do zrealizowania zadania. Zwraca on nam wykres mocy testów w zależności od $p*$.

```r
N <- 1000
alpha <- 0.05
wartosci_p <- seq(0.01, 0.99, by = 0.01)
wyniki <- list()
n <- 1000

n_power_binom <- vector("numeric", length(wartosci_p))
n_power_aymptotyczny_z_poprawka <- vector("numeric", length(wartosci_p))
n_power_asymptotyczny_bez_poprawki <- vector("numeric", length(wartosci_p))
```

Rysunek 26: Przypisanie wartości i inicjalizacja wektorów

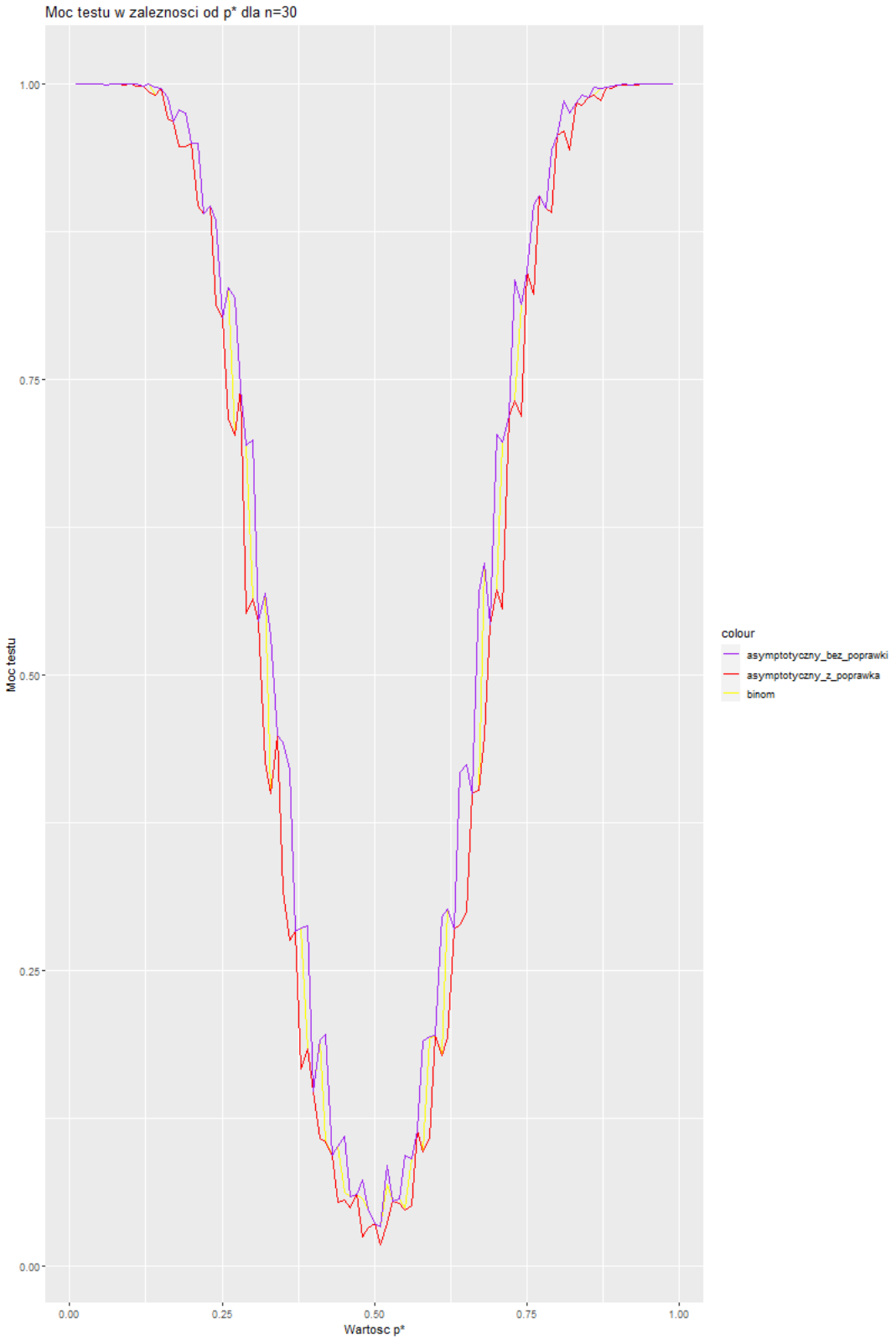
```r
for (i in 1:length(wartosci_p)) {
  p_star <- wartosci_p[i]
  odrzuc_binom <- 0
  odrzuc_asymptotyczny_z_poprawka <- 0
  odrzuc_asymptotyczny_bez_poprawki <- 0

  for (k in 1:N) {
    data <- rbinom(n, 1, 0.5)

    binom_test <- binom.test(sum(data), n, p = p_star, alternative = "t")
    prop_test_1 <- prop.test(sum(data), n, p = p_star, alternative = "t", correct = TRUE)
    prop_test_2 <- prop.test(sum(data), n, p = p_star, alternative = "t", correct = FALSE)

    if (binom_test$p.value < alpha) {
      odrzuc_binom <- odrzuc_binom + 1
    }
    if (prop_test_1$p.value < alpha) {
      odrzuc_asymptotyczny_z_poprawka <- odrzuc_asymptotyczny_z_poprawka + 1
    }
    if (prop_test_2$p.value < alpha) {
      odrzuc_asymptotyczny_bez_poprawki <- odrzuc_asymptotyczny_bez_poprawki + 1
    }
  }
  n_power_binom[i] <- odrzuc_binom / N
  n_power_aymptotyczny_z_poprawka[i] <- odrzuc_asymptotyczny_z_poprawka / N
  n_power_asymptotyczny_bez_poprawki[i] <- odrzuc_asymptotyczny_bez_poprawki / N
}

df <- data.frame(p_star = wartosci_p,
                 power_binom = n_power_binom,
                 power_asymptotyczny_z_poprawka = n_power_aymptotyczny_z_poprawka,
                 power_asymptotyczny_bez_poprawki = n_power_asymptotyczny_bez_poprawki,
                 n = as.factor(n))
wyniki[[as.character(n)]] <- df
```

Rysunek 27: Część obliczeniowa

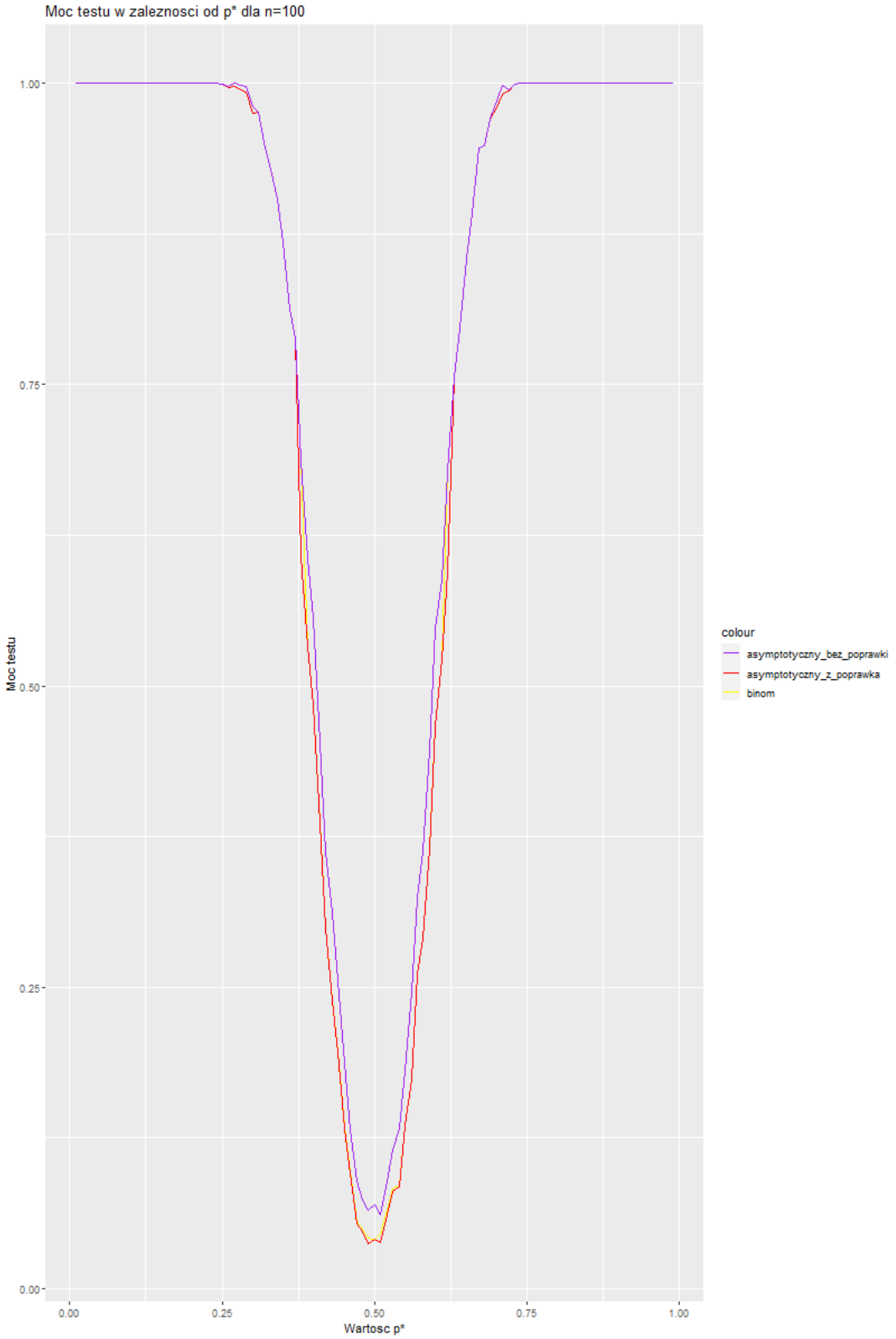
```r
plot <- ggplot(data = do.call(rbind, wyniki),
               aes(x = p_star)) +
  geom_line(aes(y = power_binom, color = "binom")) +
  geom_line(aes(y = power_asymptotyczny_z_poprawka, color = "asymptotyczny_z_poprawka")) +
  geom_line(aes(y = power_asymptotyczny_bez_poprawki, color = "asymptotyczny_bez_poprawki")) +
  labs(x = "Wartosc p*", y = "Moc testu") +
  scale_color_manual(values = c("binom" = "yellow", "asymptotyczny_z_poprawka" = "red",
                                "asymptotyczny_bez_poprawki" = "purple")) +
  ggtitle("Moc testu w zaleznosci od p* dla n=1000")
```

Rysunek 28: Rysowanie wykresów

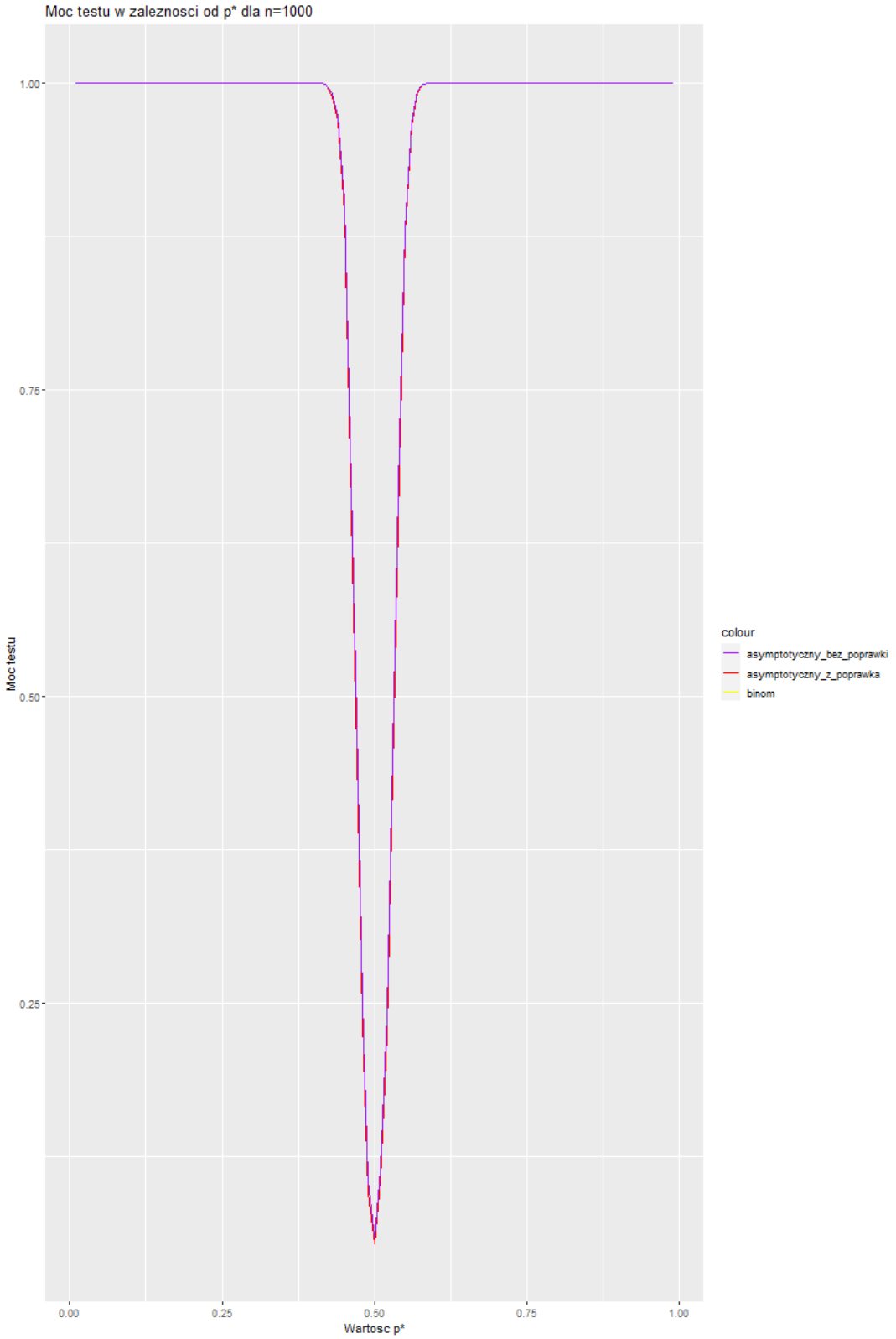
7.3.1 Wyniki



Rysunek 29: Wykres mocy dla $n = 30$



Rysunek 30: Wykres mocy dla $n = 100$



Rysunek 31: Wykres mocy dla $n = 1000$

7.4 Wnioski

Z tabeli w podpunkcie a) widzimy, że największe wartości prawdopodobieństwa popełnienia błędu I rodzaju otrzymujemy dla testu asymptotycznego bez poprawki. Ponadto widzimy, że dla testu dokładnego wartości nie zmieniają się w zależności od długości próby. Można w takim razie wywnioskować, że testy dokładne są mniej podatne na liczebność próby, czego nie można powiedzieć o testach asymptotycznych, gdzie prawdopodobieństwo popełnienia błędu I rodzaju rośnie wraz z długością próby.

Jeżeli natomiast przyjrzymy się wykresom w podpunkcie b), to zauważymy, że różnice pomiędzy testami zanikają wraz ze wzrostem próby. Jednak dla $n = 30$ i $n = 100$ widać, że to test asymptotyczny bez poprawki osiąga najwyższe moce. Wybór konkretnego testu powinien zależeć od naszych indywidualnych potrzeb - test dokładny daje nam kontrolę nad ryzykiem popełnienia błędu I rodzaju, podczas gdy test asymptotyczny sprawdza się lepiej, gdy zależy nam na większej mocy testu.