

Analiza danych ankietowych

Raport 1

Klaudia Janicka 262268, Natalia Iwańska 262270

2023-04-09

Część I

Tablice licznosci dla zmiennej A1

A1	n	prop
-2	14	0.070
-1	17	0.085
0	40	0.200
1	100	0.500
2	29	0.145

Tab. 1: Tablica licznosci dla A1.

A1	n	prop
-2	5	0.1219512
-1	6	0.1463415
0	8	0.1951220
1	19	0.4634146
2	3	0.0731707

Tab. 2: Tablica licznosci dla A1 ze wzgledu na Wyk=1.

A1	n	prop
-2	5	0.0357143
-1	10	0.0714286
0	26	0.1857143
1	75	0.5357143
2	24	0.1714286

Tab. 3: Tablica licznosci dla A1 ze wzgledu na Wyk=2.

A1	n	prop
-2	4	0.2105263
-1	1	0.0526316
0	6	0.3157895
1	6	0.3157895
2	2	0.1052632

Tab. 4: Tablica licznosci dla A1 ze wzgledu na Wyk=3.

A1	n	prop
-2	2	0.0645161
-1	2	0.0645161
0	5	0.1612903
1	19	0.6129032
2	3	0.0967742

Tab. 5: Tablica licznosci dla A1 ze wzgledu na D=Z.

A1	n	prop
-2	9	0.0918367
-1	10	0.1020408
0	17	0.1734694
1	51	0.5204082
2	11	0.1122449

Tab. 6: Tablica licznosci dla A1 ze wzgledu na D=P.

A1	n	prop
-2	3	0.0666667
-1	3	0.0666667
0	14	0.3111111
1	15	0.3333333
2	10	0.2222222

Tab. 7: Tablica liczności dla A1 ze względu na D=S.

A1	n	prop
-1	2	0.0769231
0	4	0.1538462
1	15	0.5769231
2	5	0.1923077

Tab. 8: Tablica liczności dla A1 ze względu na D=O.

A1	n	prop
-2	3	0.0422535
-1	7	0.0985915
0	14	0.1971831
1	36	0.5070423
2	11	0.1549296

Tab. 9: Tablica liczności dla A1 ze względu na P=k.

A1	n	prop
-2	11	0.0852713
-1	10	0.0775194
0	26	0.2015504
1	64	0.4961240
2	18	0.1395349

Tab. 10: Tablica liczności dla A1 ze względu na P=m.

Tablice liczności dla zmiennej W1

A1	n	prop
-2	14	0.070
-1	17	0.085
0	40	0.200
1	100	0.500
2	29	0.145

Tab. 11: Tablica liczności dla W1.

A1	n	prop
-2	5	0.1219512
-1	6	0.1463415
0	8	0.1951220
1	19	0.4634146
2	3	0.0731707

Tab. 12: Tablica liczności dla W1 ze względu na Wyk=1.

A1	n	prop
-2	5	0.0357143
-1	10	0.0714286
0	26	0.1857143
1	75	0.5357143
2	24	0.1714286

Tab. 13: Tablica liczności dla W1 ze względu na Wyk=2.

A1	n	prop
-2	4	0.2105263
-1	1	0.0526316
0	6	0.3157895
1	6	0.3157895
2	2	0.1052632

Tab. 14: Tablica liczności dla W1 ze względu na Wyk=3.

A1	n	prop
-2	2	0.0645161
-1	2	0.0645161
0	5	0.1612903
1	19	0.6129032
2	3	0.0967742

Tab. 15: Tablica liczności dla W1 ze względu na D=Z.

A1	n	prop
-2	9	0.0918367
-1	10	0.1020408
0	17	0.1734694
1	51	0.5204082
2	11	0.1122449

Tab. 16: Tablica liczności dla W1 ze względu na D=P.

A1	n	prop
-2	3	0.0666667
-1	3	0.0666667
0	14	0.3111111
1	15	0.3333333
2	10	0.2222222

Tab. 17: Tablica liczności dla W1 ze względu na D=S.

A1	n	prop
-1	2	0.0769231
0	4	0.1538462
1	15	0.5769231
2	5	0.1923077

Tab. 18: Tablica liczności dla W1 ze względu na D=O.

A1	n	prop
-2	3	0.0422535
-1	7	0.0985915
0	14	0.1971831
1	36	0.5070423
2	11	0.1549296

Tab. 19: Tablica liczności dla W1 ze względu na P=k.

A1	n	prop
-2	11	0.0852713
-1	10	0.0775194
0	26	0.2015504
1	64	0.4961240
2	18	0.1395349

Tab. 20: Tablica liczności dla W1 ze względu na P=m.

Tabele wielodzielcze

```
kable(structable(W1~P, personel) %>% addmargins(),
      caption="Tabela wielodzielcza uwzględniająca zmienną W1 i P.") %>%
  column_spec(1, border_left = TRUE) %>%
  column_spec(6, border_right = TRUE) %>%
  kable_styling(latex_options = "HOLD_position")
```

Tab. 21: Tabela wielodzielcza uwzględniająca zmienną W1 i P.

	-2	-1	1	2	Sum
K	25	10	1	35	71
M	49	10	1	69	129
Sum	74	20	2	104	200

Tab. 22: Tabela wielodzielcza uwzględniająca zmienną W1 i S.

	-2	-1	1	2	Sum
0	64	18	0	91	173
1	10	2	2	13	27
Sum	74	20	2	104	200

Tab. 23: Tabela wielodzielcza uwzględniająca zmienną A1 i D.

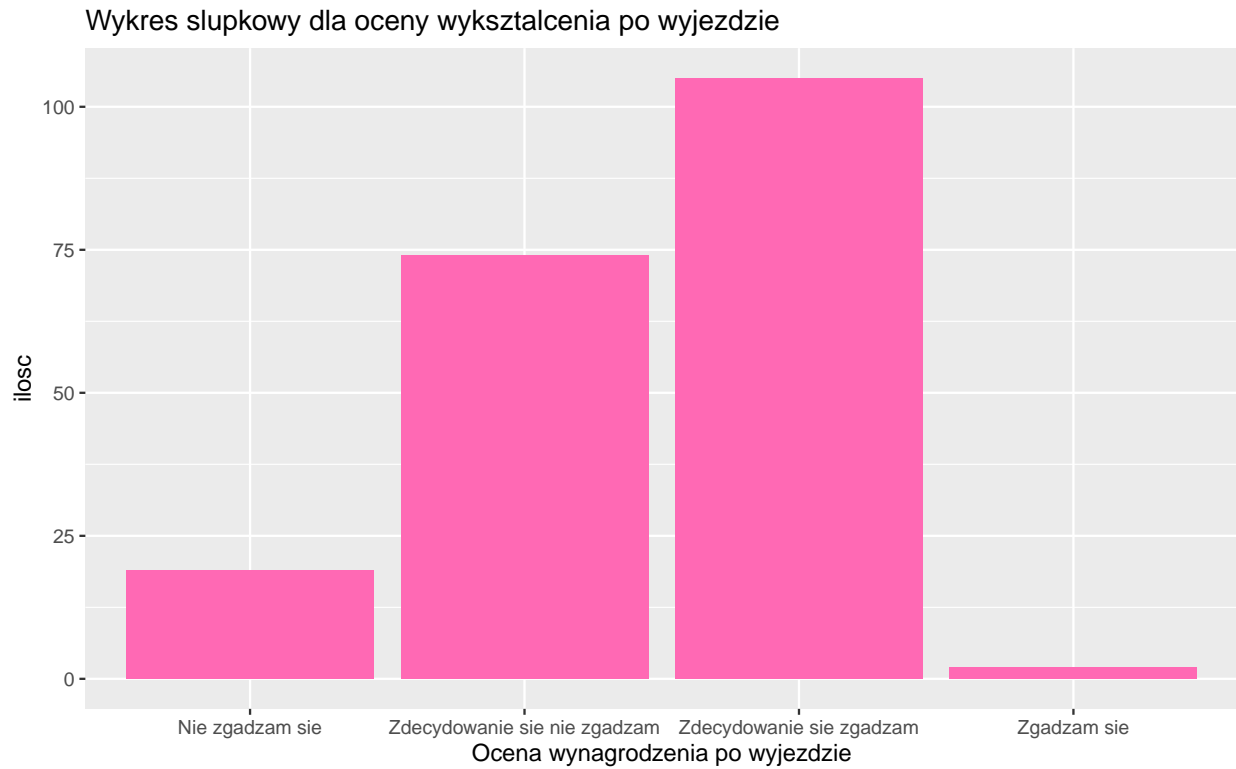
	-2	-1	0	1	2	Sum
O	0	2	4	15	5	26
P	9	10	17	51	11	98
S	3	3	14	15	10	45
Z	2	2	5	19	3	31
Sum	14	17	40	100	29	200

Wykres słupkowy

```
daneW1 <- personel %>% count(W1) %>% data.frame()

ggplot(daneW1, aes(x=W1, y=n)) +
  geom_bar(stat = "identity", fill="hotpink") +
  ggtitle("Wykres słupkowy dla oceny wykształcenia przed wyjazdem") +
  xlab('Ocena wynagrodzenia przed wyjazdem') +
  ylab('ilość')
```



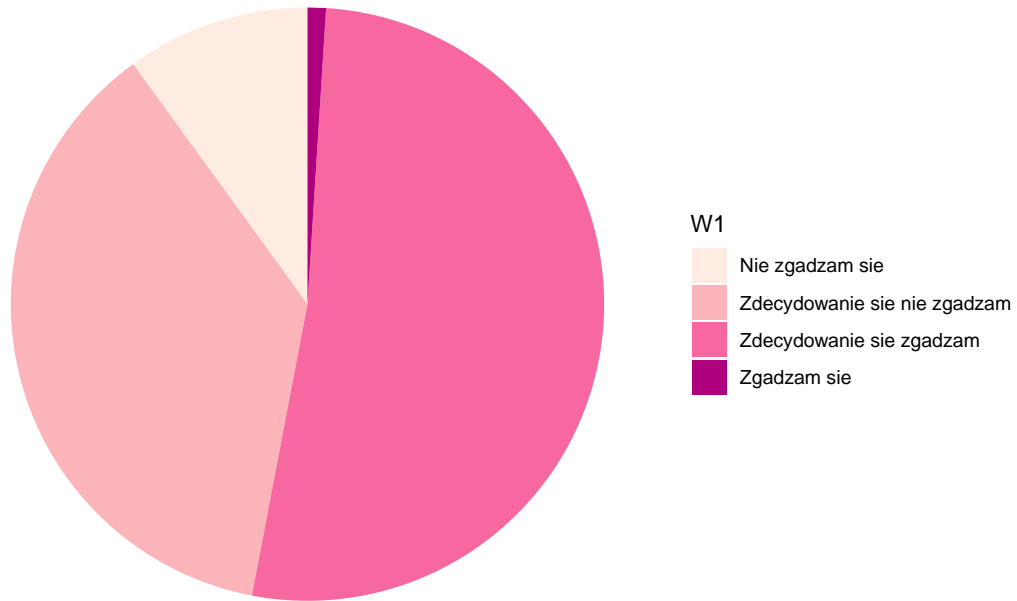


Jakieś dwa zdanie tak jak mówiła, ale jej nie słuchałam, więc nie wiem.

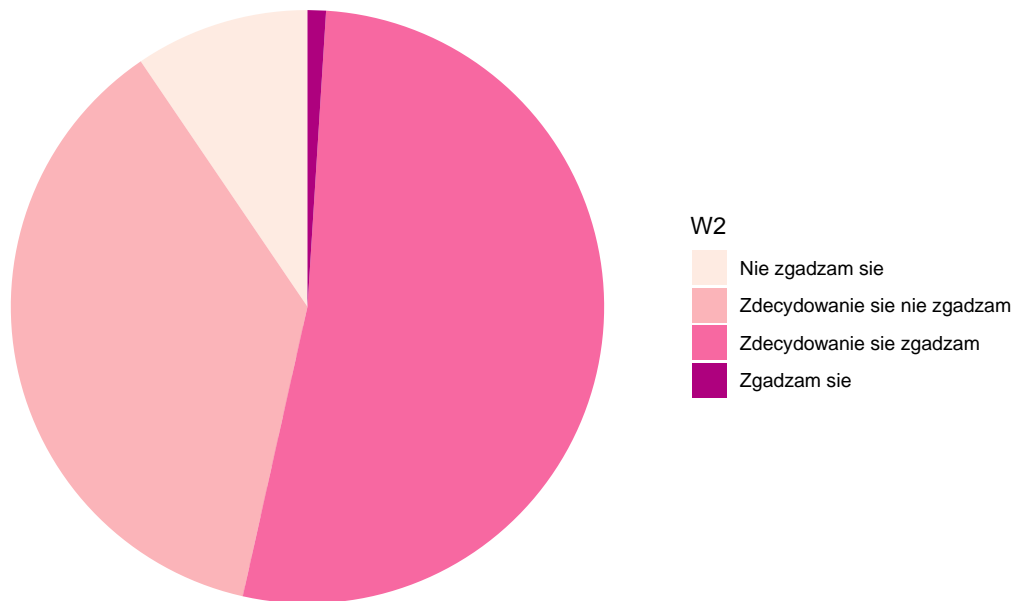
Wykres kołowy

```
ggplot(daneW1, aes(x="", y=n, fill=W1)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() +
  scale_fill_brewer(palette="RdPu") +
  ggtitle("Wykres kołowy dla oceny wynagrodzenia przed wyjazdem")
```

Wykres kołowy dla oceny wynagrodzenia przed wyjazdem

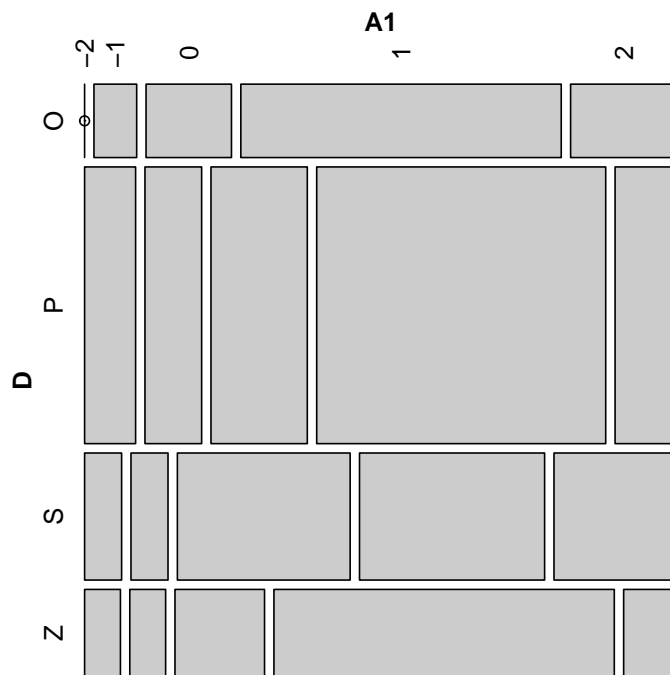


Wykres kołowy dla oceny wynagrodzenia po wyjeździe

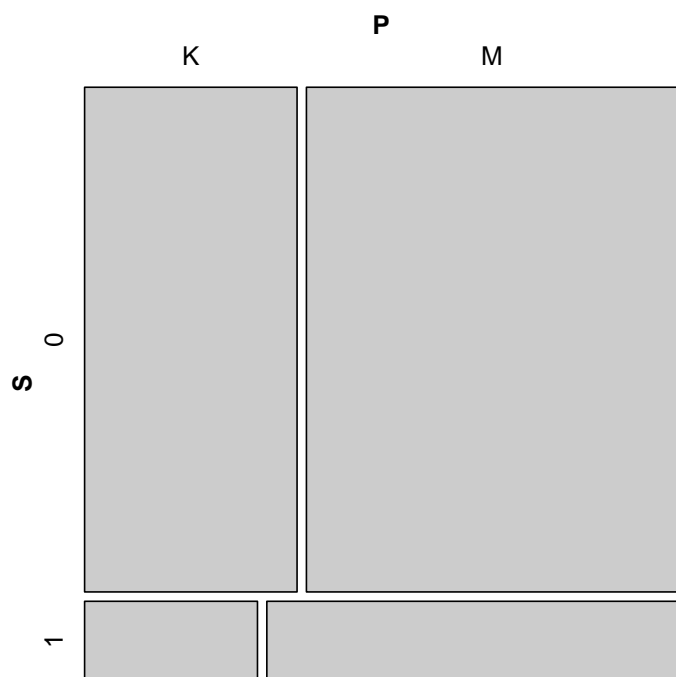
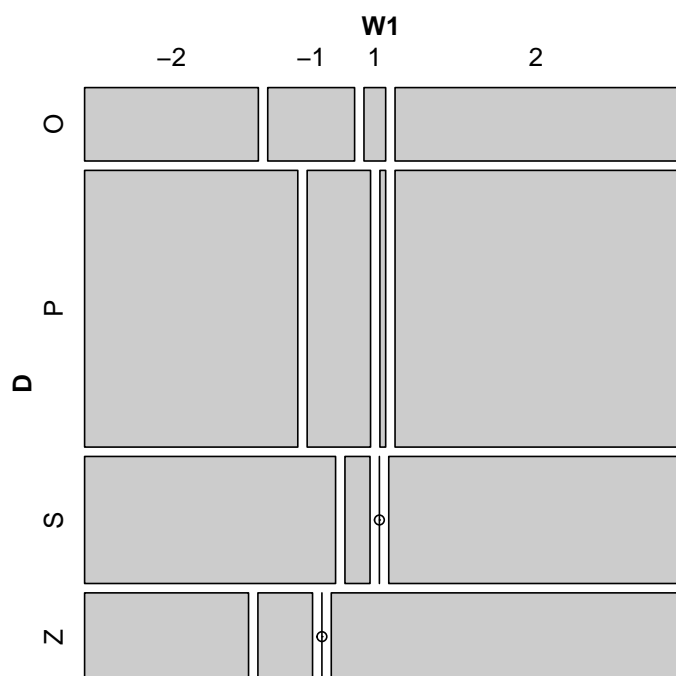


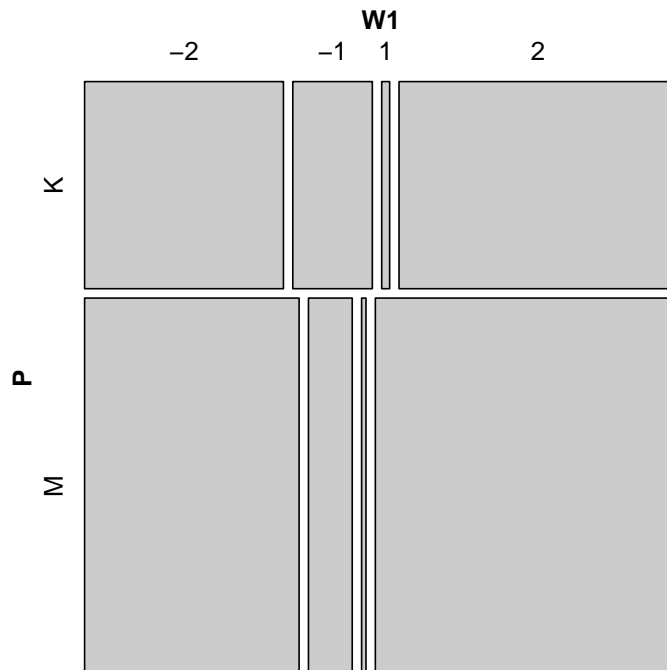
Wykresy mozaikowe

```
mosaic(~D+A1, personel, labeling = vcd::labeling_border(rot_labels = c(90, 90)))
```



Na razie zostawiam bo pierdolca dostaje





Część II

Funkcja losująca ze zwracaniem i bez

```
f <- function(x='bez'){
  if (x=='zwracanie'){
    s <- sample(1:nrow(mtcars),3,replace=TRUE)
  } else{
    s <- sample(1:nrow(mtcars),3)
  }
  mtcars[s, ]
}
```

```
f('zwracanie')
```

```
##           mpg cyl  disp  hp drat   wt  qsec vs am gear carb
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0   3    2
## Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0   3    3
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0   3    1
```

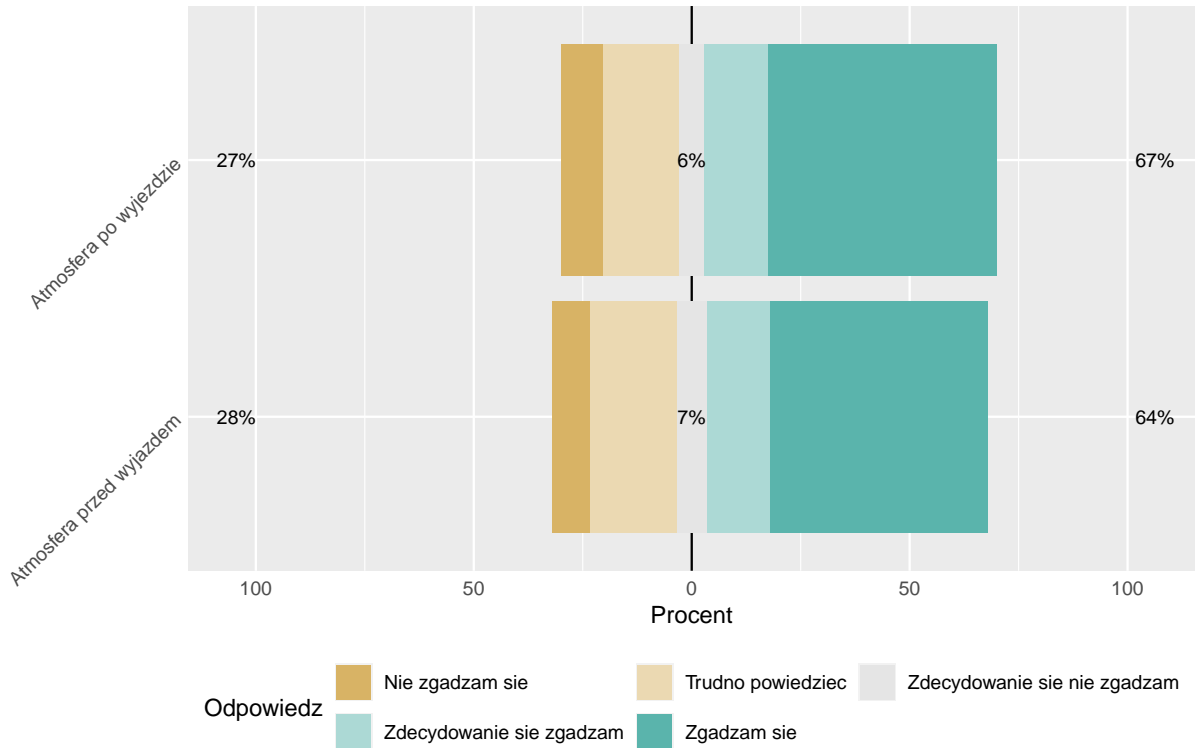
Funkcja likert

```
df <- data.frame(personel$`Atmosfera przed wyjazdem`,personel$`Atmosfera po wyjeździe`)
colnames(df) <- c('Atmosfera przed wyjazdem', 'Atmosfera po wyjeździe')
likt_atmo <- likert(df)
summary(likt_atmo)
```

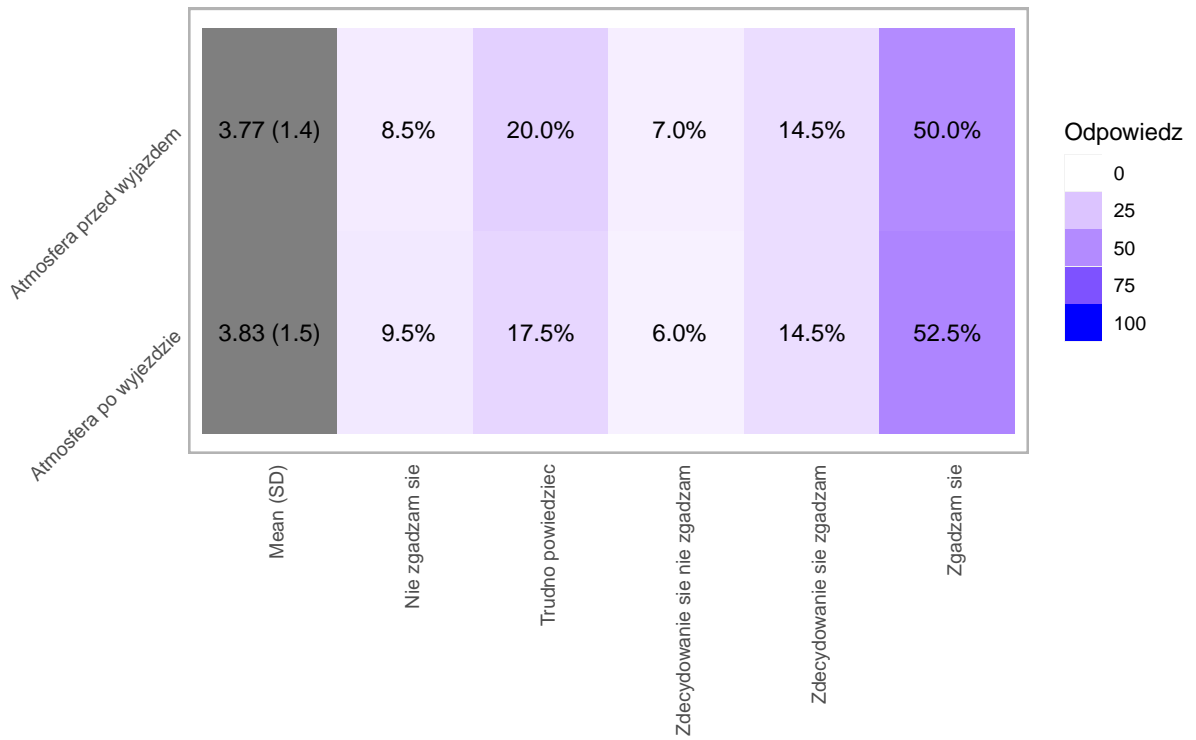
```
##           Item  low neutral high  mean      sd
```

```
## 2   Atmosfera po wyjeździe 27.0      6 67.0 3.830 1.456609
## 1   Atmosfera przed wyjazdem 28.5     7 64.5 3.775 1.447359
```

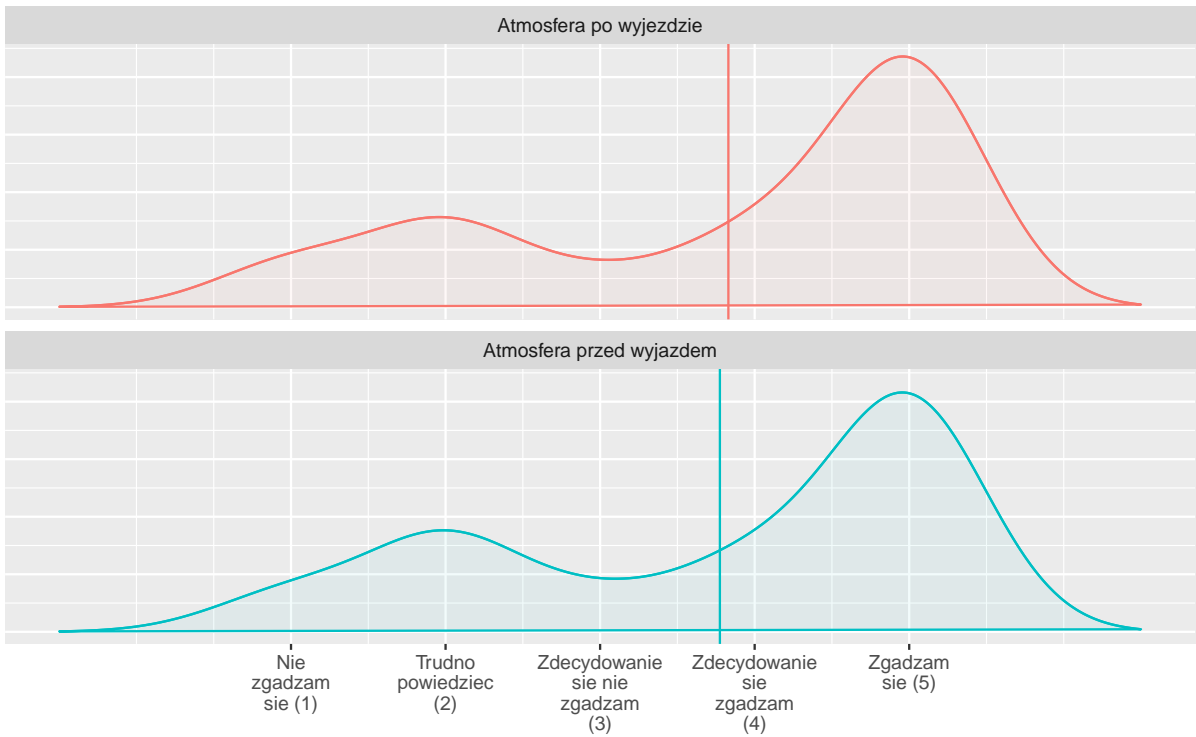
```
likert.bar.plot(likert_atmo) + guides(fill=guide_legend(nrow=2,byrow=TRUE,title='Odpowiedź')) + theme(axis.
  ylab("Procent")
```



```
plot(likert_atmo,type='heat') + theme(axis.text.x=element_text(angle=90,hjust=1),axis.text.y=element_text
```

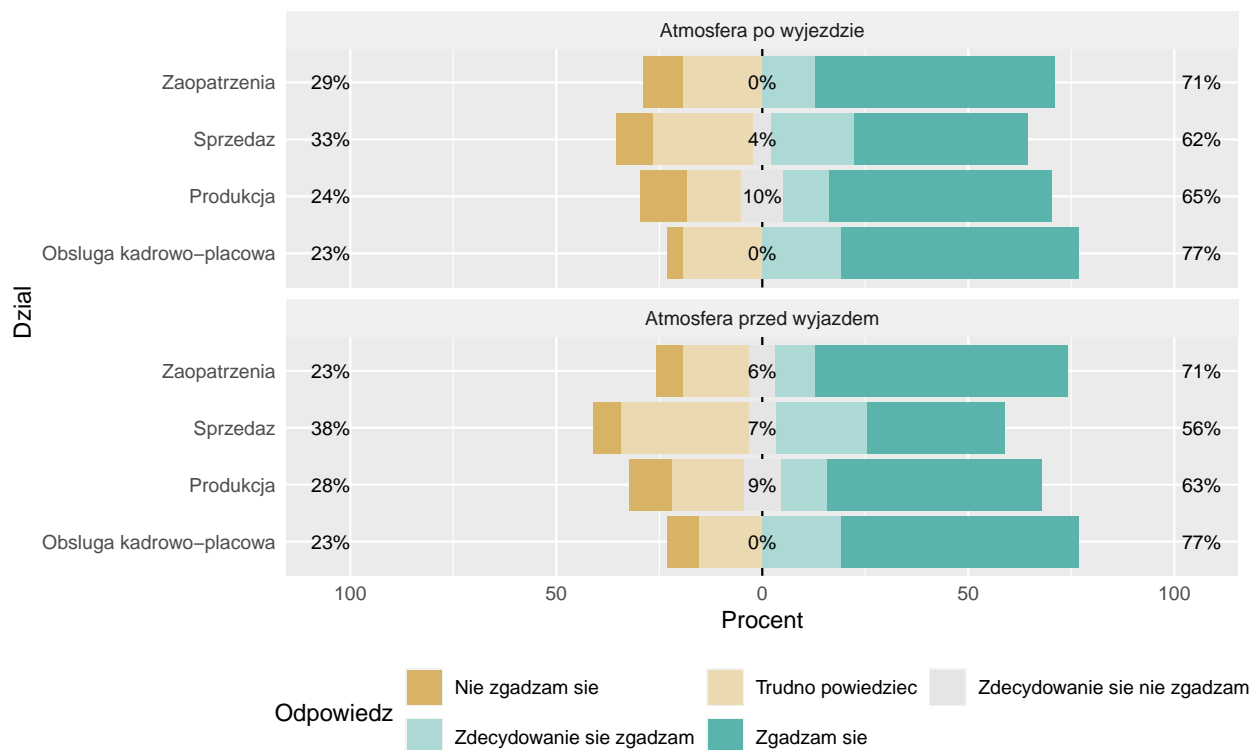


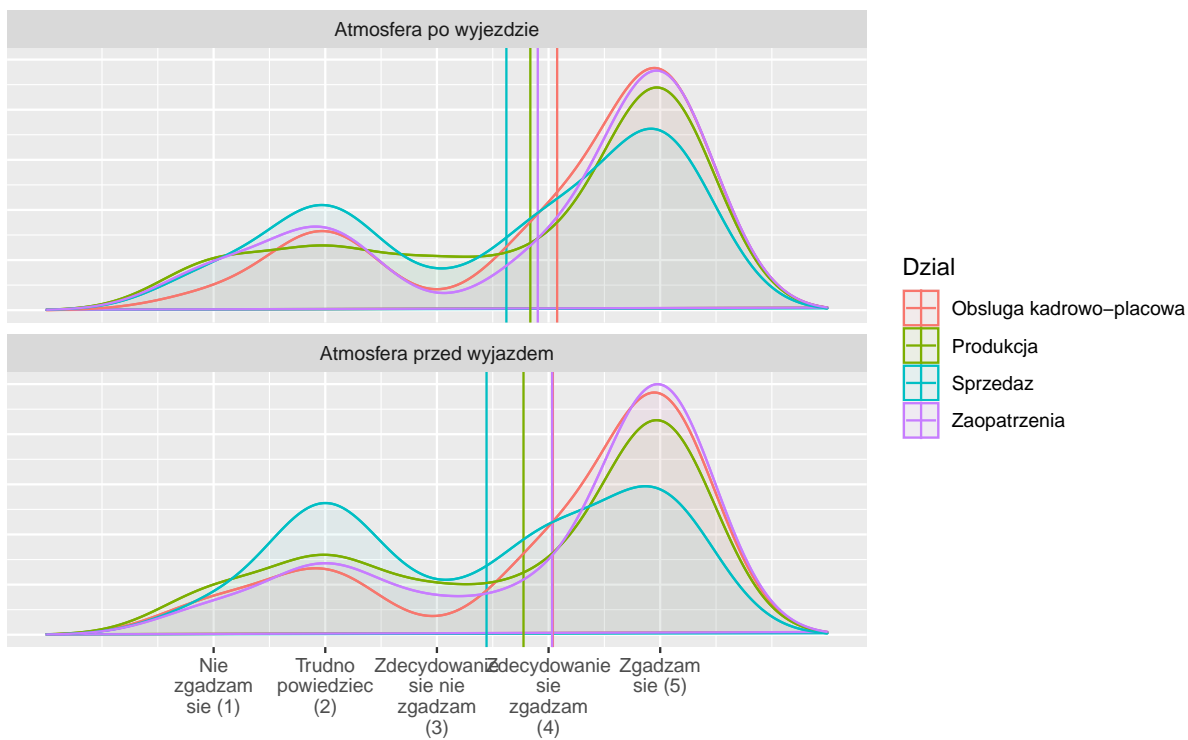
```
likert.density.plot(lik_t_atmo)
```



Podgrupa ze względu na dział

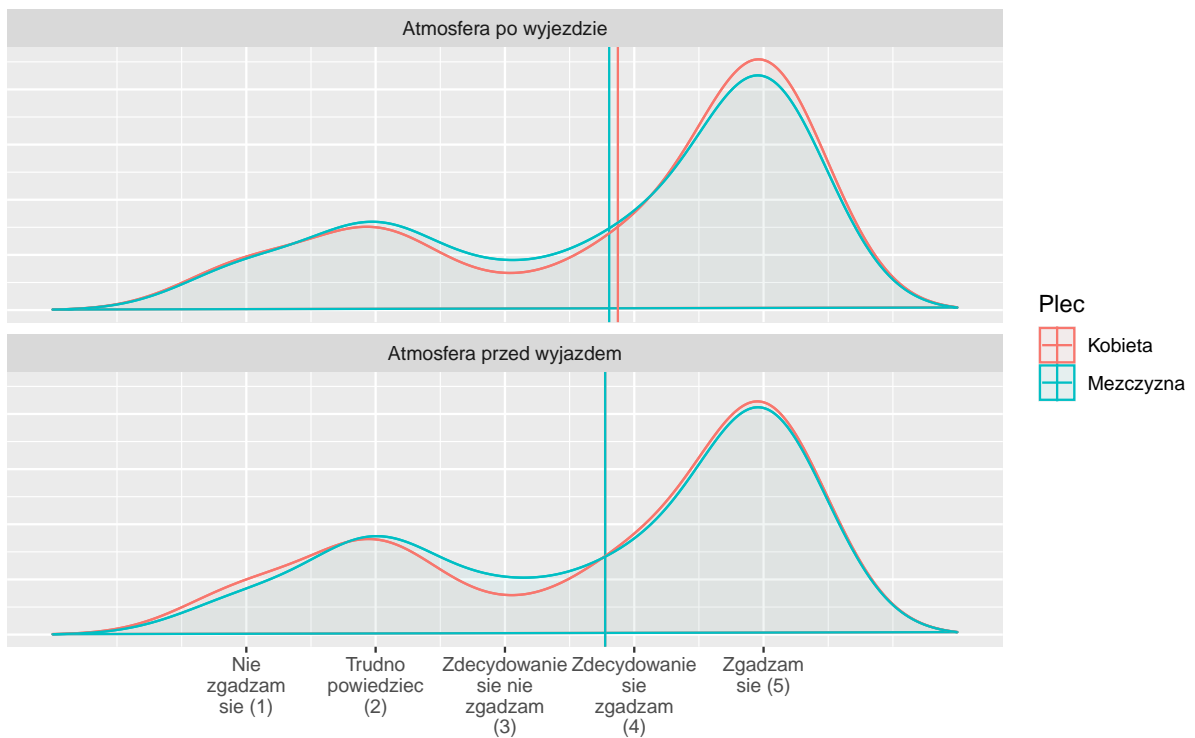
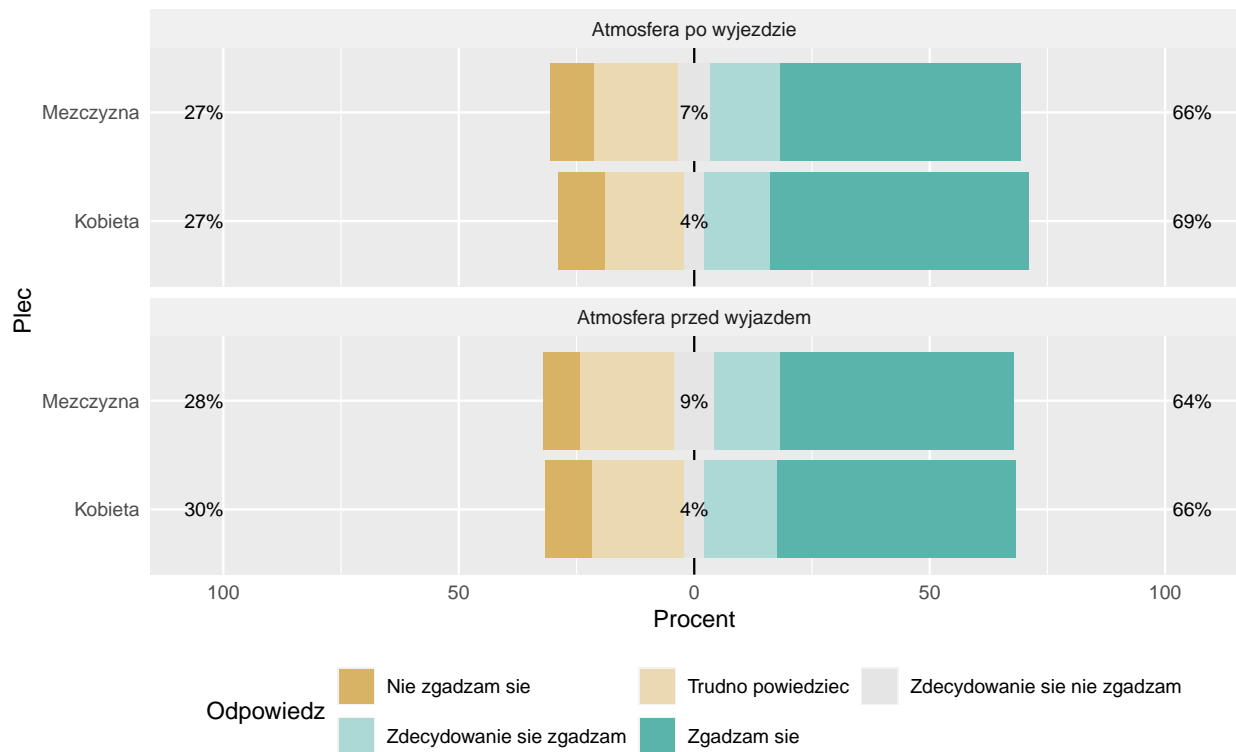
##	Group	Item	low	neutral	high
## 1	Obsługa kadrowo-płacowa	Atmosfera przed wyjazdem	23.07692	0.000000	76.92308
## 2	Obsługa kadrowo-płacowa	Atmosfera po wyjeździe	23.07692	0.000000	76.92308
## 3	Produkcja	Atmosfera przed wyjazdem	27.55102	9.183673	63.26531
## 4	Produkcja	Atmosfera po wyjeździe	24.48980	10.204082	65.30612
## 5	Sprzedaż	Atmosfera przed wyjazdem	37.77778	6.666667	55.55556
## 6	Sprzedaż	Atmosfera po wyjeździe	33.33333	4.444444	62.22222
## 7	Zaopatrzenia	Atmosfera przed wyjazdem	22.58065	6.451613	70.96774
## 8	Zaopatrzenia	Atmosfera po wyjeździe	29.03226	0.000000	70.96774
##	mean	sd			
## 1	4.038462	1.399450			
## 2	4.076923	1.324329			
## 3	3.775510	1.482062			
## 4	3.836735	1.476230			
## 5	3.444444	1.407053			
## 6	3.622222	1.466116			
## 7	4.032258	1.401996			
## 8	3.903226	1.513381			





Podgrupa ze względu na płeć

##	Group	Item	low	neutral	high	mean
## 1	Kobieta	Atmosfera przed wyjazdem	29.57746	4.225352	66.19718	3.774648
## 2	Kobieta	Atmosfera po wyjeździe	26.76056	4.225352	69.01408	3.873239
## 3	Mężczyzna	Atmosfera przed wyjazdem	27.90698	8.527132	63.56589	3.775194
## 4	Mężczyzna	Atmosfera po wyjeździe	27.13178	6.976744	65.89147	3.806202
##	sd					
## 1	1.485138					
## 2	1.472894					
## 3	1.432002					
## 4	1.452786					



Przedział ufności Cloppera-Pearsona

```

p.lower <- function(x, n, a){
  if(x == 0){
    return(0)
  }
  else{
    return(qbeta(a/2,x,n-x+1))
  }
}

p.upper <- function(x, n, a){
  if(x == n){
    return(1)
  }
  else{
    return(qbeta(1-a/2, x+1, n-x))
  }
}

clopper_pearson_ci <- function(x, n=NULL, a=0.05){
  if(is.null(n)){
    n <- length(x)
    x <- sum(x==1)
    return(c(p.lower(x, n, a), p.upper(x, n, a)))
  }
  else{
    return(c(p.lower(x, n, a), p.upper(x, n, a)))
  }
}

clopper_pearson_ci2 <- function(x, n=NULL, a=0.05){
  if(is.null(n)){
    n <- length(x)
    x <- sum(x==1)
    return(data.frame(x=x, n=n, lower=p.lower(x, n, a), upper=p.upper(x, n, a)))
  }
  else{
    return(data.frame(x=x, n=n, lower=p.lower(x, n, a), upper=p.upper(x, n, a)))
  }
}

```

Przykład użycia

- funkcja wbudowana

```
## method x n mean lower upper
## 1 exact 7 20 0.35 0.1539092 0.5921885
```

- funkcja clopper_pearcon_ci

```
## x n lower upper
## 1 7 20 0.1539092 0.5921885
```

Zadowanie z wynagrodzenia w całej badanej grupie

- funkcja `clopper_pearson_ci`

```
##      x      n      lower      upper
## 1 106 200 0.4583305 0.6007671
```

- funkcja wbudowana

```
##  method      x      n mean      lower      upper
## 1   exact 106 200 0.53 0.4583305 0.6007671
```

Zadowanie z wynagrodzenia ze względu na dział

- funkcja `clopper_pearson_ci`

```
##      x      n      lower      upper
## 1 14 26 0.3337082 0.7341288
## 2 50 98 0.4071736 0.6126014
## 3 23 45 0.3577404 0.6629663
## 4 19 31 0.4218696 0.7815004
```

- funkcja wbudowana

```
##  method      x      n      mean      lower      upper
## 1   exact 14 26 0.5384615 0.3337082 0.7341288
## 2   exact 50 98 0.5102041 0.4071736 0.6126014
## 3   exact 23 45 0.5111111 0.3577404 0.6629663
## 4   exact 19 31 0.6129032 0.4218696 0.7815004
```

Zadowanie z wynagrodzenia ze względu na stanowisko

- funkcja `clopper_pearson_ci`

```
##      x      n      lower      upper
## 1 91 173 0.4488278 0.6022889
## 2 15 27 0.3532642 0.7452012
```

- funkcja wbudowana

```
##  method      x      n      mean      lower      upper
## 1   exact 91 173 0.5260116 0.4488278 0.6022889
## 2   exact 15 27 0.5555556 0.3532642 0.7452012
```

Część III

Generowanie rozkładu dwumianowego

Algorytm

1. Ustal n i p .
2. Generuj Y_i z rozkładu $P(Y_i = 1) = p = 1 - P(Y_i = 0)$.
3. Powtórz krok 2. n razy.
4. Wstaw $X = \sum_{i=1}^n Y_i$.
5. Powtórz kroki 2-4 N razy.

```
generuj_dwumianowy <- function(n, p, N){
  X <- rep(NA, N)
  for(i in 1:N){
    Y <- rep(NA, n)
```



```

for(j in 1:n){
  Y[j] <- sample(c(0,1), size=1, prob=c(1-p,p))
}
X[i] <- sum(Y)
}
return(X)
}

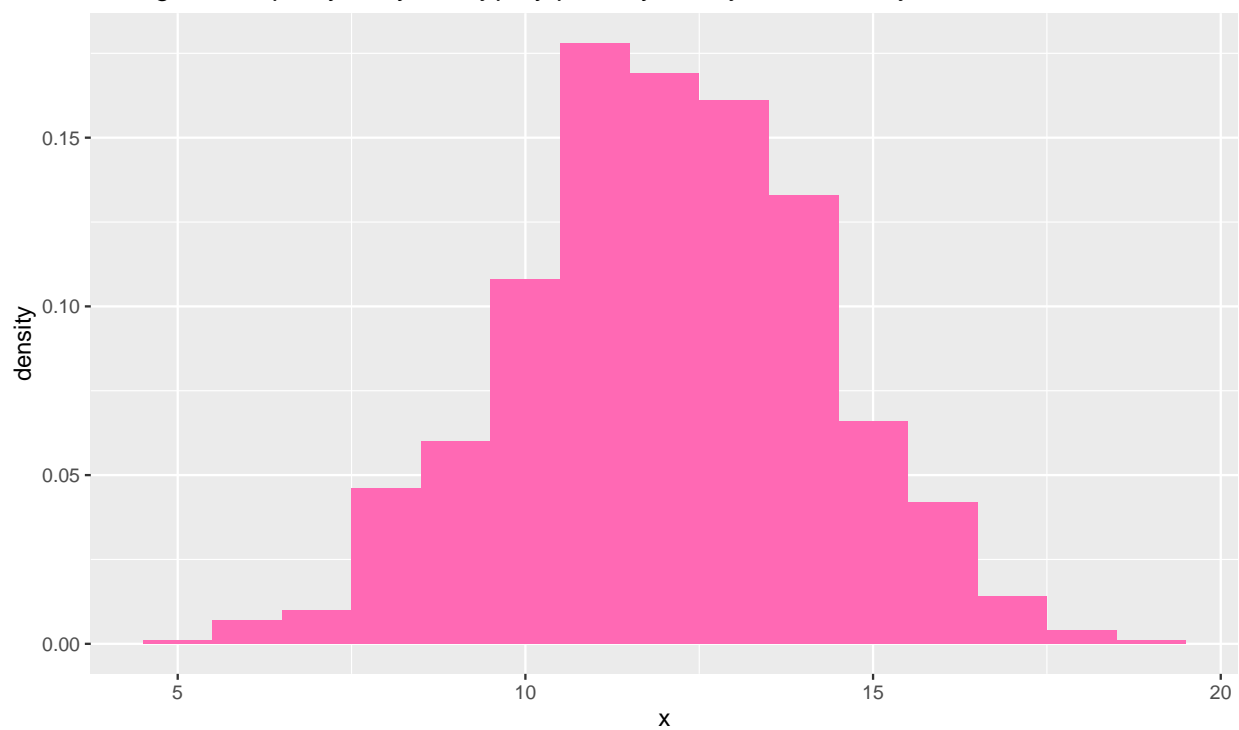
```

Porównanie funkcji

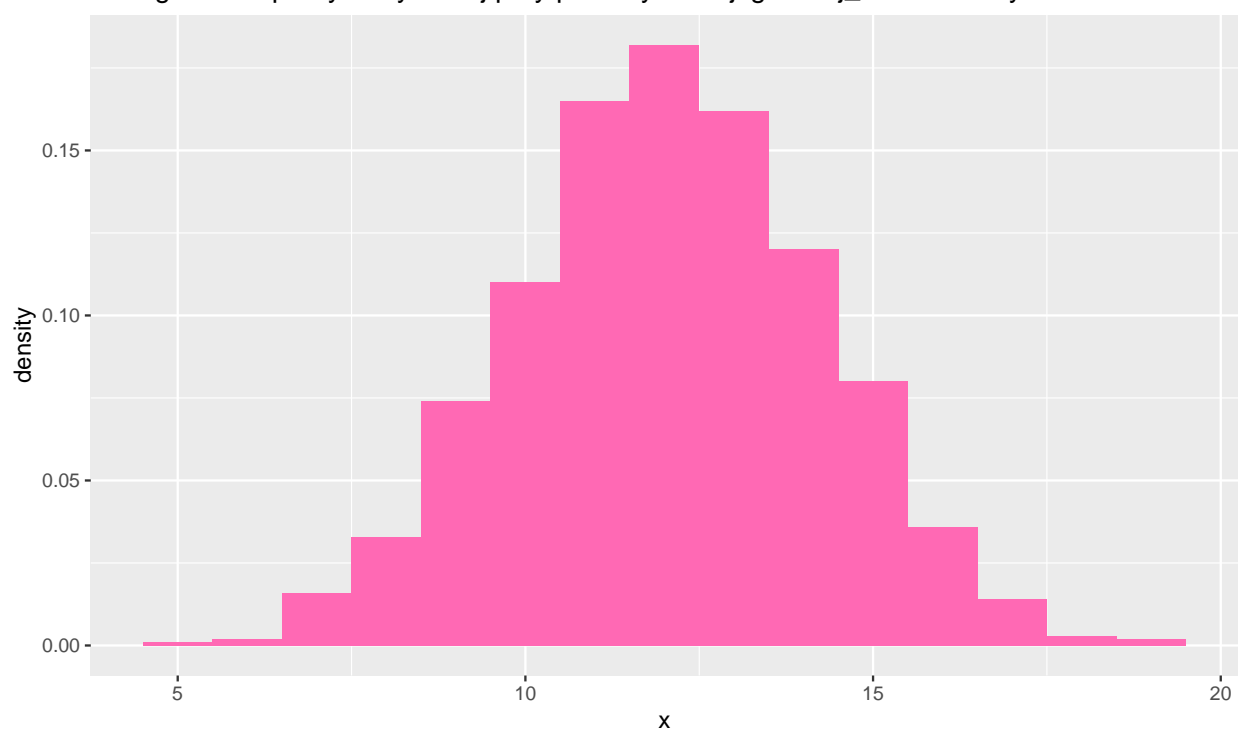
Tab. 24: Wartości statystyk

funkcja	średnia	wariancja
rbinom	12.037	4.942574
generuj_dwumianowy	12.050	4.816316

Histogram dla próby otrzymanej przy pomocy funkcji wbudowanej



Histogram dla próby otrzymanej przy pomocy funkcji generuj_dwumianowy



Prawdopodobieństwo pokrycia

Opis symulacji

Coś, że Monte Carlo itp.

Algorytm

1. Ustal n i p .
2. Generuj realizację zmiennej losowej z rozkładu $B(n, p)$.
3. Wyznacz przedział ufności dla parametru p wybraną metodą.
4. Sprawdź czy $p \in$ przedziału ufności. Wyznacz długość przedziału.
5. Powtórz 2-4 N razy.
6. Wyznacz procent pokrycia i średnią długość przedziału.

```
simulation <- function(n, method, name){
  N <- 1000
  p <- seq(0, 1, 0.01)
  for(j in 1:length(p)){
    len <- rep(NA, N)
    counter <- 0
    for(i in 1:N){
      x <- rbinom(1, n, p[j])
      interval <- binom.confint(x, n, conf.level = 0.95, methods = method)
      TL <- interval$lower
      TU <- interval$upper
      if(between(p[j], TL, TU)){
        counter <- counter + 1
      }
      len[i] <- TU - TL
    }
    p.cover <- counter/N
    p.len <- mean(len)
    df <- data.frame('p' = p[j], 'pokrycie' = p.cover, 'dlugosc' = p.len)
    write.table(df, name,
               append = TRUE,
               sep = ",",
               col.names = FALSE,
               row.names = FALSE,
               quote = FALSE)
  }
}
```

