# Test chi-kwadrat

## Test $\chi^2$

Przypomnijmy sobie na początek podstawowy schemat statystycznego testowania hipotez:

### Testowanie hipotez (przypomnienie)

0. Wymyślamy naszą hipotezę alternatywną (ta nas interesuje!)
1. Przyjmujemy hipotezę zerową (i ew. dodatkowe założenia).
2. Decydujemy się na określony poziom istotności $\alpha$.
3. Konstruujemy rozkład interesującej nas statystyki z próby (przy założeniu $H_0$ i ew. innych założeniach) (dziś będzie to statystyka testowa $\chi^2$).
4. Określamy region krytyczny (biorąc pod uwagę $\alpha$ i ewentualnie kierunkowość hipotezy) (przy teście $\chi^2$ kierunkowość nie ma znaczenia - test ten jest zawsze jednostronny).
5. Losujemy próbę, zbieramy dane, liczymy statystykę.
6. Odrzucamy, bądź nie $H_0$

### Statystyka testowa $\chi^2$

Poniżej podany jest wzór na statystykę testową $\chi^2$:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

gdzie:

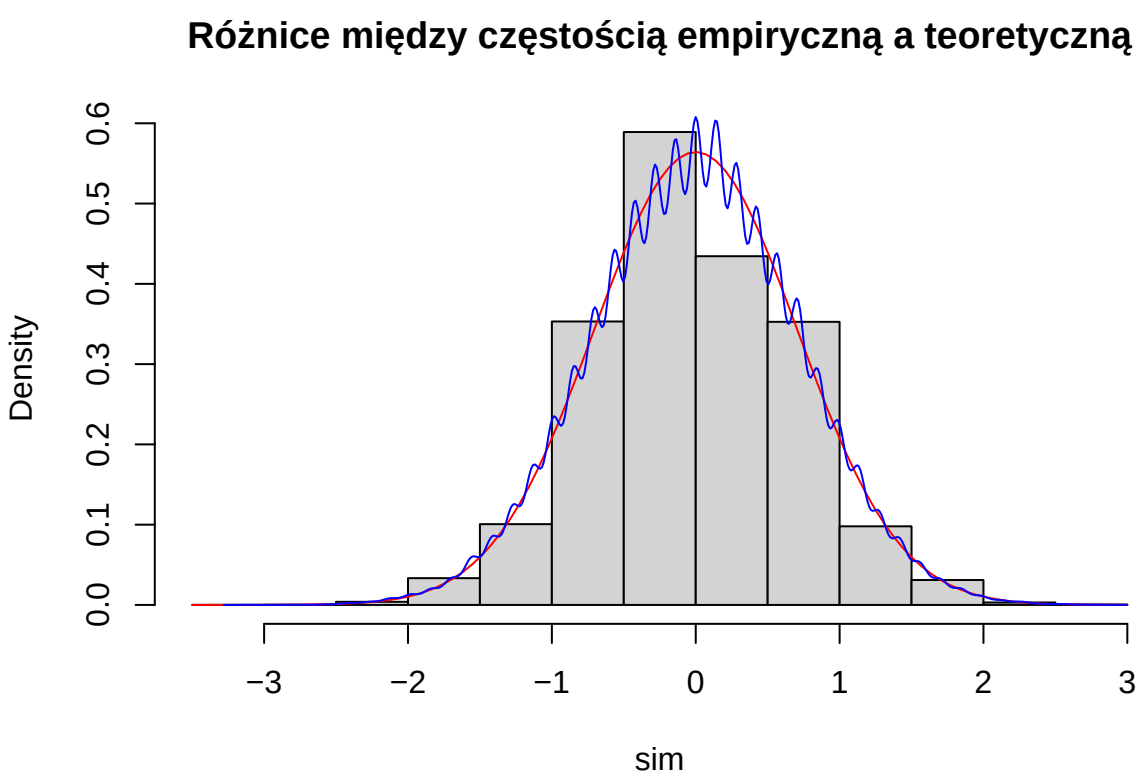
- $O$ - zaobserwowana liczba wystąpień określonego zdarzenia
- $E$ - oczekiwana/teoretyczna liczba wystąpień określonego zdarzenia

Kształt rozkładu teoretycznego $\chi^2$ zależy *wyłącznie od liczby stopni swobody*. Rozkład ten będzie dla nas bardzo przydatny w testowaniu hipotez ze względu na to, że rozkład $\chi^2$ to rozkład zmiennej losowej, która jest sumą $k$ kwadratów niezależnych zmiennych losowych o standardowym rozkładzie normalnym. W tej chwili może wydawać się to niezbyt przydatna informacja, jak jednak zobaczymy, jest ona kluczowa dla logiki działania testów $\chi^2$.

Wrócimy do tego jeszcze za chwilę. Teraz wyobraźmy sobie, że rzucamy 100 razy (uczciwą) monetą. Spodziewamy się, że orzeł wypadnie około 50 razy. Jest to teoretyczna/oczekiwana liczba orłów w naszym eksperymencie. Obliczmy różnicę pomiędzy faktyczną liczbą orłów a teoretyczną liczbą orłów i podzielmy ją na pierwiastek z teoretycznej liczbę orłów. Można dowieść z Centralnego Twierdzenia Granicznego (w bardziej "zaawansowanym" sformułowaniu, niż omawialiśmy), że takie zmienna wraz ze wzrostem liczebności próby będzie dążyć do rozkładu normalnego $N(0, 1-p)$. My zamiast tego dowodzić, pokażemy, że ta zależność działa, korzystając z prostej symulacji.

```
sim <- replicate(100000, { # powtórzmy nasze doświadczenie 100 000 razy
    # losujemy 100 wartości z rozkładu dwupunktowego o P=0.5
    k <- sum(sample(0:1, 100, replace = TRUE))
    # liczymy różnicę między częstością obserwowaną a teoretyczną
    # następnie dzielimy przez pierwiastek z oczekiwanej liczby wystąpień
    (k-50)/sqrt(100*0.5)
})
```

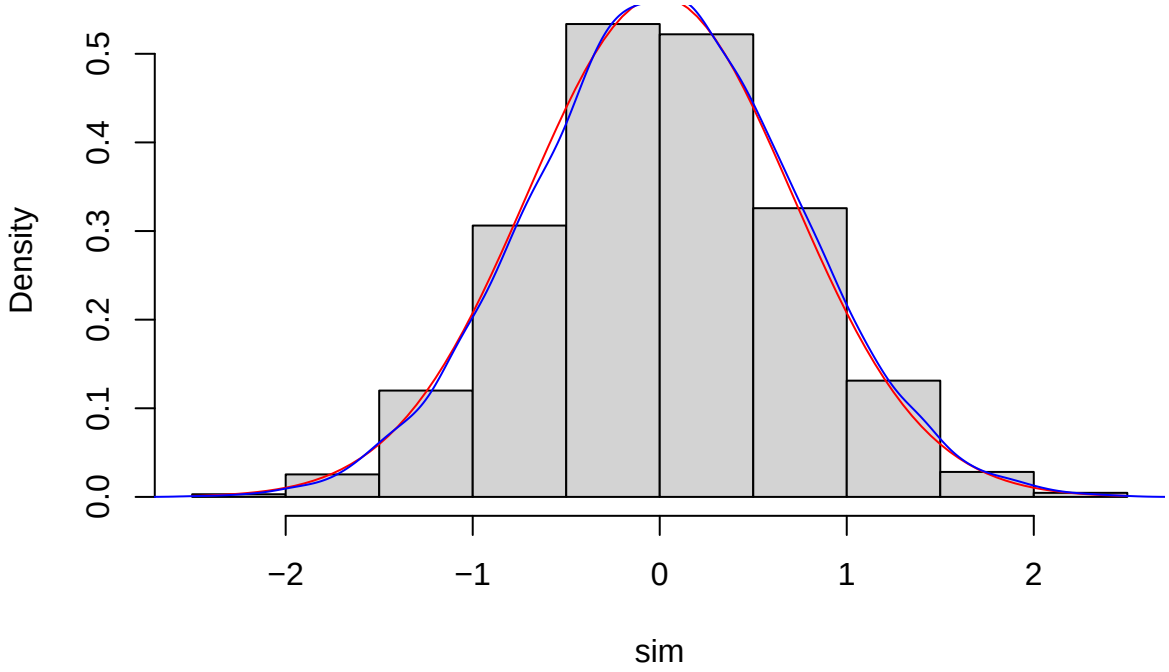
```r
# Następnie stwórzmy histogram z naszych wyników
hist(sim, freq = FALSE,
     main = "Różnice między częstością empiryczną a teoretyczną")

# Na stworzony histogram nałożymy krzywa gęstości prawdopodobieństwa N(0, 1-p)
curve(dnorm(x, sd = sqrt(1-0.5)), add = TRUE, col='red')

# Gęstość "empiryczna" z jądrowego estymatora gęstości
points(density(sim), type = 'l', col = 'blue')
```




To "pofalowanie" krzywej gęstości naszych symulowanych danych wynika oczywiście z faktu, że nasza zmienna nie jest tak naprawdę ciągła - może przyjmową pewne wartości (liczby naturalne od $-50$ do 50 podzielone przez pierwiastek z 50), a pewnych nie może (np. 124.34). Można łatwo stwierdzić to, patrząc na wzór - jeśli O może być tylko liczbą całkowitą liczbą całkowitą, to $\frac{(O-E)^2}{E}$ nie będzie mogło przyjąć wszystkich wartości ze zbioru dodatnich liczb rzeczywistych. Wraz ze wzrostem liczebności próby może przyjmować coraz więcej wartości więc krzywa w sposób naturalny "wygładzi" się.

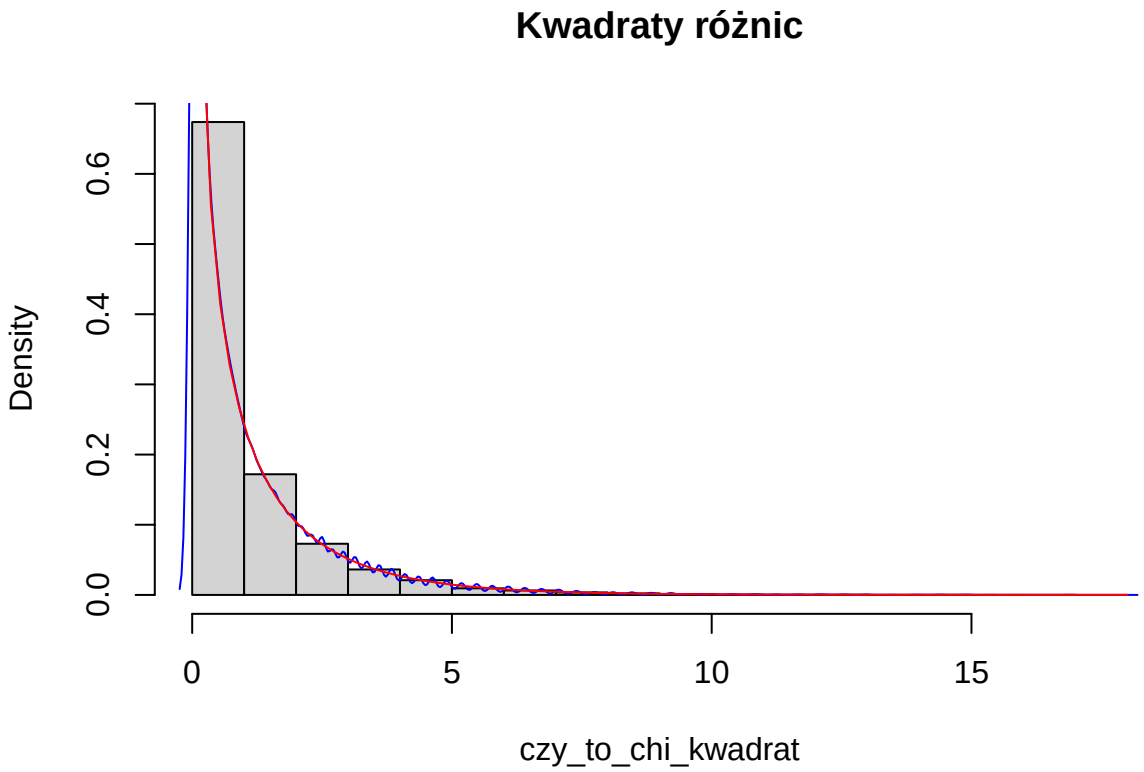
Przy $N = 10000$ już wygląda zupełnie jak rozkład normalny.

```r
sim <- replicate(10000, {
    k = sum(sample(0:1, 10000, replace = TRUE))
    (k-5000)/sqrt(10000*0.5)
})

hist(sim, freq = FALSE,
     main = "Różnice między częstością teoretyczną a empiryczną")
curve(dnorm(x, sd = sqrt(1-0.5)), add = TRUE, col='red')
points(density(sim), type = 'l', col = 'blue')
```



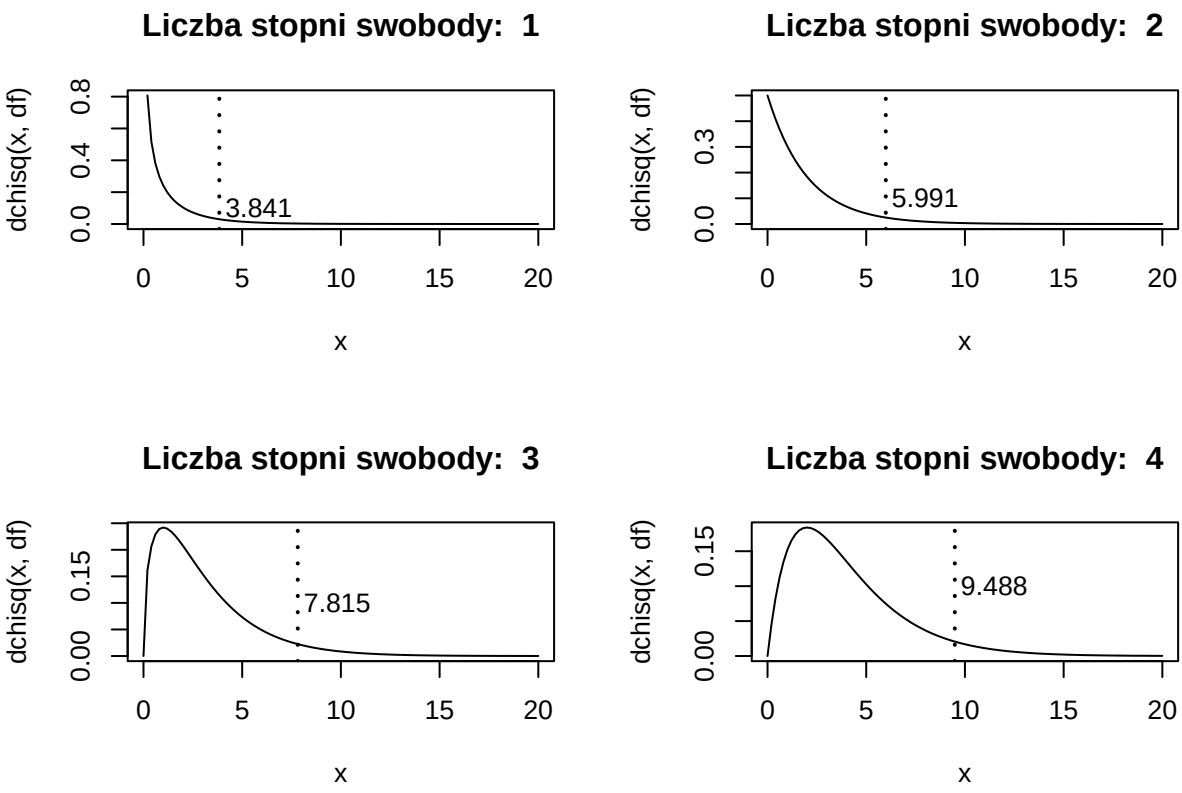

W testach opartych na statystyce testowej $\chi^2$ będzie nas interesował rozkład sumy kwadratów kilku takich zmiennych. Można dowieść, że dla $r$ takich zmiennych rozkład taki ma postać $\chi^2(r-1)$ to znaczy jest to rozkład $\chi^2$ o $r-1$ stopniach swobody. Nie będziemy tego dowodzić, dowód mogą Państwo znaleźć np. tu: https://ocw.mit.edu/courses/mathematics/18-443-statistics-for-applications-fall-2003/lecture-notes/lec23.pdf (prawdopodobnie nigdy nie będą Państwo tego dowodzić). Przetestujmy tę "prawdę objawioną" na przykładzie dwóch zmiennych. Jak widzimy nasz rozkład empiryczny $\chi^2$, który otrzymaliśmy za pomocą symulacji (histogram i niebieska krzywa) odpowiada rozkładowi teoretycznemu (czerwona krzywa).

```
czy_to_chi_kwadrat <- replicate(100000, {
    k = sum(sample(0:1, 1000, replace = TRUE))
    ((k-500)**2)/(1000*0.5) + # jedna zmienna (liczba orłów)
    ((1000-k-500)**2)/(1000*0.5) # druga zmienna (liczba reszek)
})


hist(czy_to_chi_kwadrat, freq = FALSE, main = "Kwadraty różnic")
points(density(czy_to_chi_kwadrat), type = 'l', col = 'blue')
curve(dchisq(x, 1), from = 0, add = TRUE, col = 'red')
```




Na poniższej ilustracji przedstawione mamy wykresy rozkładu $\chi^2$ dla różnej liczby stopni swobody. Na każdy z wykresów nałożona została również pionowa linia odcinająca górne 5% rozkładu. Tam będą wartości statystyki testowej uprawniające nas do odrzucenia hipotezy zerowej (oczywiście tylko wtedy, gdy przyjmiemy $\alpha = 0.05$)!

```
par(mfrow = c(2,2))
for (df in 1:4) {
    curve(dchisq(x, df), 0, 20, main = paste('Liczba stopni swobody: ', df))
    wk = qchisq(0.95, df) # wartość krytyczna
    abline(v=wk, lwd = 2, lty = 3) # linia wyznaczająca obszar krytyczny
    text(x = wk+2, y = 0.1, labels = as.character(round(wk,3))) # wartość krytyczna
}
```




## Testowanie hipotez za pomocą testu $\chi^2$

Na tych zajęciach będziemy zajmować się dwoma testami bazującymi na statystyce testowej $\chi^2$. Pierwszy z nich to test zgodności $\chi^2$ (*goodness-of-fit test*), drugi to test niezależności $\chi^2$ (*test of independence*). Z obliczeniowego punktu widzenia resty te działają dość podobnie, są miedzy nimi jednak istotne różnice i stosuje się je w nieco innych sytuacjach.

### Przykład I (test zgodności)

Korzystając z naszej teoretycznej wiedzy możemy wykorzystać rozkład $\chi^2$ do statystycznego testowania hipotez.

Prześledźmy przykład zaczerpnięty z podręcznika Davida Howella.

> Praktycy Leczniczego Dotkniecia twierdzą, że potrafią wyleczyć różne dolegliwości zdrowotne używając swoich dłoni do manipulacji "polem energetycznym człowieka". Emily Rosa zarekrutowała praktyków Leczniczego Dotknięcia do eksperymentu. Polegał ona na tym, że mając zawiązane oczy mieli oni wykryć, nad którą z ich dłoni znajduje się dłoń Emily (a zawsze była tylko nad jedną). Okazało sie że w 280 próbach tylko 123 razy udało im się wskazać właściwą dłoń.

Chcemy dowiedzieć się, czy rezultaty osiągane przez praktyków DT różnią się od rezultatów, których spodziewalibyśmy się, gdyby zgadywali. Nie interesuje nas w tym wypadku kierunek różnicy, tzn. to, czy odpowiadali lepiej czy gorzej, niż gdyby odpowiadali całkowicie losowo.

Wiemy, że w rzeczywistości 123 razy odpowiedzieli poprawnie a w 157 niepoprawnie. Ponadto wiemy, że gdyby zgadywali, to wartościami oczekiwanymi byłoby 140 i 140.

Nasza hipoteza głosi wiec, że rozkład empiryczny (ten, który otrzymaliśmy w eksperymencie) różni się od rozkładu teoretycznego (w tym przypadku - losowego wskazywania).

$$H_A : \neg(X \sim F)$$

a zatem hipotezą zerową (którą będziemy chcieli obalić!) jest:

$$H_0 : X \sim F$$

Innymi słowy, będziemy starać się wykazywać, że nasze dane nie pochodzą z określonego rozkładu teoretycznego.

Obliczmy statystykę testową $\chi^2$ według wzoru:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

- $\sum$ - suma
- $O$ - zaobserwowana liczba wystąpień danego zdarzenia
- $E$ - oczekiwana liczba wystąpień danego zdarzenia

Następnie używając teoretycznego rozkładu $\chi^2$ sprawdźmy, jakie jest prawdopodobieństwo uzyskania takiego lub bardziej skrajnego wyniku przy założeniu prawdziwości hipotezy zerowej.

Liczba stopni swobody jaka nas interesuje (tzn. 1) dana jest przez wzór:

$$df = r - 1$$

- $df$ - liczba stopni swobody
- $r$ - liczba zmiennych losowych

Generalnie zasadą jest to, że liczba stopni swobody równa jest liczbie kategorii, którą zmniejszamy o liczbę stopni swobody, które "utraciliśmy" dopasowując rozkład do danych.

Obrazowo - wyobraźmy sobie, że mamy 3 kategorie i nie wiemy nic o tym, ile jest zliczeń w każdej z nich, wiemy tylko ile jest obserwacji w sumie. Dokładnie taki wynik możemy uzyskać przy różnych liczbach zliczeń, możemy je modyfikować ALE jeśli ustalimy już wartości w dwóch komórkach, to wartość trzeciej komórki jest zdeterminowana.

W innych kategoriach można myśleć o tym tak: wyobraźmy sobie że rzucamy monetą 100 razy. Czy liczba orłów i liczba reszek są zmiennymi niezależnymi? Oczywiście nie - jeśli znamy liczbę orłów to wiemy już, jaka jest liczba reszek. Stąd nasza redukcja stopni swobody.

Liczba stopni swobody przy teście zgodności jest równa liczbie zmiennych, które możemy swobodnie zmieniać zachowując taką samą sumę wszystkich zliczeń.

(W przypadku tabeli 2x2 i testu niezależności, który omówimy nieco później, sprawa wygląda troszkę inaczej. Miejsce naszej sumy wszystkich zliczeń zajmują sumy brzegowe. Jak łatwo zauważyć, w tej sytuacji możemy swobodnie zmienić wartość tylko jednej zmiennej. Jesli już ją ustalimy to wartości sum brzegowych determinują już liczby w pozostałych komórkach. Tracimy po jednym stopniu swobody z "wiersza" i z "kolumny". Jest to prawdą także dla tabeli 2x3 - jeżeli ustalimy wartości w dwóch komórkach, to sumy brzegowe determinują wartości w pozostałych. Tak samo jak w przypadku testu zgodności struktura zależności między zmiennymi sprawia, że musimy zredukować liczbę stopni swobody. Zobaczymy, jak to wszystko działa, kiedy omówimy drugi typ testu $\chi^2$. Teraz wróćmy jednak do naszego przykładu.)

Obliczmy więc statystykę testową:

```
zaobserwowane <- c(123,157)
oczekiwane <- c(140,140)

# Obliczamy statystykę testową chi kwadrat
chi_kw <- sum((zaobserwowane - oczekiwane)**2/oczekiwane)

# Za pomocą dystrybuanty możemy wyznaczyć prawdopodobieństwo uzyskania takiego
```

```r
# lub bardziej skrajnego wyniku
# Uwaga! Interesuje nas zawsze górna część rozkładu, stąd `lower.tail = FALSE`
pchisq(chi_kw, 1, lower.tail = FALSE)
```

```
## [1] 0.04216493
```

```r
# za pomocą funkcji kwantylowej (=odwrotnej dystrybuanty) możemy obliczyć wartość krytyczną
qchisq(0.95,1)
```

```
## [1] 3.841459
```

```r
# to samo możemy łatwiej uzyskać za pomocą znajdującej
# się w bibliotece standardowej funkcji `chisq.test`
chisq.test(c(123,157))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(123, 157)
## X-squared = 4.1286, df = 1, p-value = 0.04216
```

Okazuje się, że dla poziomu istotności statystycznej $\alpha = 0.05$ hipoteza zerowa została obalona. Mamy prawo przyjąć hipotezę alternatywną, głoszącą, że rozkład odpowiedzi różni się od rozkładu teoretycznego. Problemem dla praktyków Leczniczego Dotknięcia jest to, że różni się na ich niekorzyść :).

Skądinąd wiadomo, że ze względu na własności rozkładu $\chi^2$ nasze postępowanie jest równoważne użyciu rozkładu normalnego $N(0, 1-p)$ i dwustronnej hipotezy. Wynik ten dokładnie pokrywa się z wynikiem, które otrzymalibyśmy stosując test dwumianowy korzystając z aproksymacji z rozkładu normalnego.

```r
pnorm((123-140)/sqrt(280*0.5), sd = sqrt(1-0.5))*2
```

```
## [1] 0.04216493
```

```r
pnorm((123-140)/sqrt(280*0.5*0.5))*2
```

```
## [1] 0.04216493
```

Rozkład teoretyczny $\chi^2$ nie zawsze dokładnie odpowiada teoretycznemu rozkładowi statystyki $\chi^2$ z próby (bo jej rozkład jest np. "pofalowany" - patrz wykres na górze). Możemy czasami uzyskać dokładniejszy wynik za pomocą symulacji.

```r
# możemy użyć do tego funkcji `chisq.test` wywołanej z
# argumentem `simulate.p.value`
chisq.test(c(123,157), simulate.p.value = TRUE, B = 100000)
```

```
##
##  Chi-squared test for given probabilities with simulated p-value (based
##  on 1e+05 replicates)
##
## data:  c(123, 157)
## X-squared = 4.1286, df = NA, p-value = 0.04759
```

```r
# dla ilustracji lepiej zrobić to samemu w tym celu
# wylosujemy 1kk prób z rozkładu dwupunktowego
# dla każdej obliczymy statystykę testową chi-kwadrat
# i zobaczymy ile z nich było równe lub większe od tej
# którą uzyskalismy w naszym eksperymencie

q = replicate(100000, { # 100 000 razy replikujemy eksperyment
  x = sample(c('C', 'I'), size = 280, replace = TRUE) # polegający na losowaniu ze zwracaniem 280 warto
```

```
                                                    # odpowiadających sukcesowi i porażce
                                                    # przy założeniu, że ich prawdopodobieństwa są ró
  sum(((table(x)[1] - 140)**2)/140, ((table(x)[2] - 140)**2)/140) # obliczamy statystykę testową chi^2
})

length(q[q>=chi_kw])/length(q) # sprawdzamy ile było wyników takich lub bardziej skrajnych jak ten otrz
```

```
## [1] 0.048
```

```
                                # w prawdziwym eksperymencie
```

### Przykład II (test zgodności)

Wyobraźmy sobie, że przeprowadziliśmy eksperyment na grupie 75 dzieci grających w Papier-Kamień-Nożyce i zapisywaliśmy, którego ze znaków użyło dziecko w swojej rozgrywce. Okazało się, że 30 razy padł Kamień, 21 Papier i 24 Nożyce. Czy mamy prawo powątpiewać w to, że dzieci wybierały znak losowo?

```
zaobserwowane <- c(30,21,24)
oczekiwane <- c(25,25,25)

chi_kw <- sum((zaobserwowane - oczekiwane)**2/oczekiwane)

pchisq(chi_kw, 2, lower.tail = FALSE)
```

```
## [1] 0.4317105
```

```
qchisq(0.95,2)
```

```
## [1] 5.991465
```

```
chisq.test(c(30,21,24))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  c(30, 21, 24)
## X-squared = 1.68, df = 2, p-value = 0.4317
```

Ponownie możemy obliczyć wartość $p$ za pomocą metody Monte Carlo (czyli symulacji). Zasadniczo nie będziemy robić tego w praktyce, ale warto zrobić to kilka razy żeby uzmysłowić sobie, że rozkład teoretyczny $\chi^2$ rozkład statystyki testowej $\chi^2$ nie zawsze są dokładnie tym samym rozkładem.

```
chisq.test(c(30,21,24), simulate.p.value = TRUE, B = 100000)
```

```
##
##  Chi-squared test for given probabilities with simulated p-value (based
##  on 1e+05 replicates)
##
## data:  c(30, 21, 24)
## X-squared = 1.68, df = NA, p-value = 0.4503
```

```
q = replicate(100000, {
  x = sample(c('R', 'P', 'S'), size = 75, replace = TRUE) # losujemy z 3 wartości
                                                          # nie chodzi o Rychu Peja Saluffka tylko Rock Paper
  sum(((table(x)[1] - 25)**2)/25, # obliczamy statystykę testową chi^2
      ((table(x)[2] - 25)**2)/25,
      ((table(x)[3] - 25)**2)/25)
})
```

```
length(q[q>=chi_kw])/length(q) # sprawdzamy ile było wyników takich lub bardziej skrajnych jak ten otrz
```

```
## [1] 0.45124
```

```
                                # w prawdziwym eksperymencie
```

Warto zwrócić uwagę na to, że nie zawsze teoretyczne prawdopodobieństwa będą równe i wobec tego nie zawsze będą równe częstości teoretyczne. W zadaniach domowych spotkają Państwo tego typu sytuacje - w R obsłużyć można je za pomocą argumentu `p`(rawdopodobieństwa) funkcji `chisq.test`.

### Przykład III (test niezależności)

W przeważającej większości przypadków nasze dane będą kategoryzowane nie we względu na jedną klasyfikację (jak w poprzednich przykładach) ale na dwie (lub więcej).

Bardzo często interesuje nas pytanie, czy jedna zmienna jest zależna od drugiej. W tym celu musimy skonstruować tabelę krzyżową (zwaną także tabelą kontyngencji lub rozdzielczą).

Następnie musimy obliczyć oczekiwane częstości w każdej komórce takiej tabeli. Po zrobieniu tego możemy zastosować test niezależności $\chi^2$. Jak łatwo się domyslić naszą hipotezą zerową będzie to, że dwie zmienne są od siebie niezależne (bo hipotezą alternatywną jest, że są zależne).

Wartości oczekiwane obliczamy według następującego wzoru:

$$E_{ij} = \frac{R_i C_j}{N}$$

gdzie $R_i$ i $C_j$ to sumy brzegowe (*marginal totals*) czyli w naszym przypadku suma wszystkich obserwacji z odpowiednio - wiersza i kolumny, w którym znajduje się nasza komórka w tabeli.

Zajmiemy się teraz danymi dotyczącymi kary śmierci w północnej karolinie zebranymi przez Unah i Borgera, które przywołuje Howell w swoim podręczniku.

```
<td colspan = 2> Kara śmierci </td>
<td></td>
```

Rasa oskarżonego

Tak

Nie

Suma

Nie-biały

33

251

284

Biały

33

508

541

Suma

66

759

825

Chcemy sprawdzić, czy rodzaj wyroku jest zależny od rasy oskrazonego.

Zestaw naszych hipotez to:

$H_A$ : Cechy $X$ i $Y$ są zależne.

Uważamy bowiem, że wyniki te świadczą o tym, że skazanie na karę śmierci ($Y$) jest zależne od rasy oskarżonego ($X$). Będziemy zatem obalać następującą hipotezę:

$H_0$ : Cechy $X$ i $Y$ są niezależne

Rozpoczniemy od obliczenia wartości oczekiwanych. Metod na zrobienie tego jest kilka, oto dwie z nich:

```
library(DescTools)
c_table = matrix(c(33,251,33,508),2,2, byrow = TRUE)

# za pomocą margin.table możemy wyliczyć sumy brzegowe

n = margin.table(c_table)
k = margin.table(c_table, 2)
w = margin.table(c_table, 1)

g = expand.grid(w,k) # proszę zajrzeć do dokumentacji tej funkcji

# To są nasze liczebności teoretyczne

exp_table = matrix((g$Var1 * g$Var2)/n,2,2)

# Można sobie pomóc funkcją ExpFreq z pakietu DescTool

oczekiwane = ExpFreq(c_table)

print('Zaobserwowane liczebności')
```

```
## [1] "Zaobserwowane liczebności"
```

```
c_table
```

```
##      [,1] [,2]
## [1,]   33  251
## [2,]   33  508
```

```
print('Liczeności teoretyczne')
```

```
## [1] "Liczeności teoretyczne"
```

```
exp_table # metodą expand.grid
```

```
##       [,1]   [,2]
## [1,] 22.72 261.28
## [2,] 43.28 497.72
```

```
oczekiwane # metodą mnożenia element-po-elemencie
```

```
##       [,1]   [,2]
## [1,] 22.72 261.28
## [2,] 43.28 497.72
```

Kiedy mamy już liczebności oczekiwane oraz liczebności rzeczywiście zaobserwowane, możemy obliczyć statystykę testową $\chi^2$, korzystając ze znanego już nam wzoru:

```
chi_kw = sum(((c_table - exp_table)**2)/exp_table)
```

Sam test możemy przeprowadzić albo wyznaczając wartość krytyczną i region odrzucenia dla określonego przez nas poziomu $\alpha$, albo obliczając wartość $p$ i sprawdzając, czy jest niższa niż próg istotności.

```
print('Obszar krytyczny')
```

```
## [1] "Obszar krytyczny"
```

```
qchisq(0.95,1)
```

```
## [1] 3.841459
```

```
print('Chi-kwadrat')
```

```
## [1] "Chi-kwadrat"
```

```
chi_kw
```

```
## [1] 7.709865
```

```
print('Wartość p')
```

```
## [1] "Wartość p"
```

```
pchisq(chi_kw, 1, lower.tail = FALSE)
```

```
## [1] 0.005491987
```

Do przeprowadzania testu $\chi^2$ możemy wykorzystać wbudowaną funkcję `chisq.test`. Musimy jednak zdawać sobie sprawę, że domyślnie aplikuje ona poprawkę na ciągłość dla tabel 2x2.

```
# Standardowo funkcja `chisq.test` aplikuje poprawkę
# na ciągłość dla tabel 2x2
print('Z poprawką')
```

```
## [1] "Z poprawką"
```

```
chisq.test(c_table)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  c_table
## X-squared = 6.9781, df = 1, p-value = 0.008251
```

```
# Kiedy ją wyłączymy, to zwrócony wyniki będzie identyczny
print('Bez poprawki')
```

```
## [1] "Bez poprawki"
```

```
# jak nasz
chisq.test(c_table, correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  c_table
## X-squared = 7.7099, df = 1, p-value = 0.005492
```

```r
# Ale możemy równie dobrze nie robić żadnych poprawek
# i wyliczyć wartość p metodą Monte Carlo
# Ciekawe czy bliższa będzie wartości bez poprawki czy z poprawką na ciągłość?

print('Zasymulowana wartość p ')
```

```
## [1] "Zasymulowana wartość p "
```

```r
chisq.test(c_table, simulate.p.value = TRUE, B = 10000000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 1e+07
##  replicates)
##
## data:  c_table
## X-squared = 7.7099, df = NA, p-value = 0.006811
```

### Przykład IV (test niezależności)

Zwykle nie będziemy mieli naszych danych w postaci tabeli krzyżowej, ale ramki danych. Często spotykamy się również z sytuacją, w której jakaś funkcja potrzebuje do działania ramki danych. Spróbujmy stworzyć sobie "udawaną" ramkę danych a nastepnie zobaczmy funkcję `CrossTable` z pakietu `gmodels`. Pozwala tworzyć "wydruki", na których znajduje się cała masa przydantych informacji dotyczących danych z tabel krzyżowych - zliczenia, wartości oczekiwane, sumy brzegowe, rozkłady brzegowe, różne statystyki testowe, itp (zachęcam do zajrzenia do dokumentacji).

Kolejny przykład dotyczy terapii antydepresantami w leczeniu anoreksji. Od dawna przypuszczana, że depresja jest jednym z powodów, przez które dziewczęta leczone z anoreksji wpadają w nią ponownie. Zwyczajnym sposobem postępowania jest przepisywanie antydepresantów. Walsh zbadał 93 pacjentki, którym udało się powrócić do prawidłowej wagi. 49 z nich został przepisany Prozac, 44 przyjmowało placebo. Eksperyment miał sprawdzić wpływ przyjmowania antydepresantów na nawrót choroby.

| Terapia | Sukces | Nawrót |
|---|---|---|
| Prozac | 13 | 36 |
| Placebo | 14 | 30 |

Czy mamy powody by sądzić, że terapia jest skuteczna w powstrzymywaniu nawrotów? Innymi słowy - czy przyjmowanie depresantów i nawrót choroby są zmiennymi zależnymi? Aby się tego dowiedzieć, przyjmiemy, że są niezależne (hipoteza zerowa) i będziemy próbowali to twierdzenie obalić.

Najpierw stworzymy sobie ramkę danych...

```r
countsToCases <- function(x, countcol = "Freq") {
    # Get the row indices to pull from x
    idx <- rep.int(seq_len(nrow(x)), x[[countcol]])

    # Drop count column
    x[[countcol]] <- NULL

    # Get the rows from x
    x[idx, ]
}

c_table <- as.table(matrix(c(13,14,36,30), 2,2))
df <- countsToCases(as.data.frame(c_table))
```

```
colnames(df) = c('Outcome', 'Treatment')
levels(df$Outcome) = c('Success', 'Relapse')
levels(df$Treatment) = c('Drug', 'Placebo')
head(df)
```

```
##     Outcome Treatment
## 1   Success      Drug
## 1.1 Success      Drug
## 1.2 Success      Drug
## 1.3 Success      Drug
## 1.4 Success      Drug
## 1.5 Success      Drug
```

Oto nasza ramka! Teraz, żeby przeprowadzić test $\chi^2$ musimy zbudować tabelę krzyżową. Alternatywnie możemy użyć funkcji z pakietu `gmodels`, która domyślnie zwraca dużo więcej informacji (ale ktoś mógłby powiedzieć - za dużo!). Z ramki danych można zrobić tabelę krzyżową tworzymy przy użyciu funkcji `table`.

```
#install.packages('gmodels')
library(gmodels)
```

```
## Registered S3 method overwritten by 'gdata':
##   method         from
##   reorder.factor DescTools
```

```
table(df$Outcome, df$Treatment)
```

```
##
##           Drug Placebo
##   Success   13      36
##   Relapse   14      30
```

```
chisq.test(table(df$Outcome, df$Treatment), correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$Outcome, df$Treatment)
## X-squared = 0.31458, df = 1, p-value = 0.5749
```

```
# Jesli ktoś lubi, to można na bogato użyć wspomnianej funkcji `CrossTable`
CrossTable(df$Treatment, df$Outcome, expected = TRUE, fisher = TRUE) # jak ktoś lubi...
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |              Expected N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  93
##
```

```
##
##              | df$Outcome
## df$Treatment |   Success |   Relapse | Row Total |
## -------------|-----------|-----------|-----------|
##         Drug |        13 |        14 |        27 |
##              |    14.226 |    12.774 |           |
##              |     0.106 |     0.118 |           |
##              |     0.481 |     0.519 |     0.290 |
##              |     0.265 |     0.318 |           |
##              |     0.140 |     0.151 |           |
## -------------|-----------|-----------|-----------|
##      Placebo |        36 |        30 |        66 |
##              |    34.774 |    31.226 |           |
##              |     0.043 |     0.048 |           |
##              |     0.545 |     0.455 |     0.710 |
##              |     0.735 |     0.682 |           |
##              |     0.387 |     0.323 |           |
## -------------|-----------|-----------|-----------|
## Column Total |        49 |        44 |        93 |
##              |     0.527 |     0.473 |           |
## -------------|-----------|-----------|-----------|
##
##
## Statistics for All Table Factors
##
##
## Pearson's Chi-squared test
## ------------------------------------------------------------
## Chi^2 =  0.3145837     d.f. =  1     p =  0.574881
##
## Pearson's Chi-squared test with Yates' continuity correction
## ------------------------------------------------------------
## Chi^2 =  0.1102895     d.f. =  1     p =  0.7398148
##
##
## Fisher's Exact Test for Count Data
## ------------------------------------------------------------
## Sample estimate odds ratio:  0.7759623
##
## Alternative hypothesis: true odds ratio is not equal to 1
## p =  0.6501028
## 95% confidence interval:  0.2859687 2.089782
##
## Alternative hypothesis: true odds ratio is less than 1
## p =  0.3695202
## 95% confidence interval:  0 1.809645
##
## Alternative hypothesis: true odds ratio is greater than 1
## p =  0.785158
## 95% confidence interval:  0.3309917 Inf
##
##
##
```

**Przykład V (test niezależności)**

Jankowski, Leitenberg, Henning i Coffey zbadali związek między molestowaniem seksualnym w dzieciństwie i molestowaniem seksualnym w dorosłym życiu. Wyniki ich badań na próbie 934 kobiet przedstawiały się tak:

```
c_table = as.table(matrix(c(512,227,59,18,54,37,15,12), 4,2))
colnames(c_table) = c('No', 'Yes')
rownames(c_table) = c('0', '1', '2', '3-4')
names(attributes(c_table)$dimnames) <- c("Number of CAC checked","Abused as Adult")
c_table
```

```
##                       Abused as Adult
## Number of CAC checked  No Yes
##                   0   512  54
##                   1   227  37
##                   2    59  15
##                   3-4  18  12
```

Spróbujmy przeprowadzić test niezależności $\chi^2$. Standardowo, najpierw zrobimy to ręcznie (obliczając wartości oczekiwane i statystykę testową ze wzoru). Potem zaś zobaczymy, czy nasz wynik zgadza się z wynikiem zwracanym przez funkcję `chisq.test`.

```
n <- margin.table(c_table)
k <- margin.table(c_table, 2)
w <- margin.table(c_table, 1)

g <- expand.grid(w,k)
exp_table <- matrix((g$Var1 * g$Var2)/n,4,2)

chi_kw <- sum(((c_table - exp_table)**2)/exp_table)

print('Wartości oczekiwane')
```

```
## [1] "Wartości oczekiwane"
```

```
exp_table
```

```
##           [,1]      [,2]
## [1,] 494.49251 71.507495
## [2,] 230.64668 33.353319
## [3,]  64.65096  9.349036
## [4,]  26.20985  3.790150
```

```
print('Wartość krytyczna')
```

```
## [1] "Wartość krytyczna"
```

```
qchisq(0.95,3)
```

```
## [1] 7.814728
```

```
print('Wartość p')
```

```
## [1] "Wartość p"
```

```
pchisq(chi_kw, 3, lower.tail = FALSE) # 3 stopnie swobody, ponieważ mamy (2-1) x (4-1) czyli 1 x 3 czyl
```

```
## [1] 1.653054e-06
```

```
chisq.test(c_table) # dostajemy warning z powodu małych oczekiwanych wartości
```

```
## Warning in chisq.test(c_table): Aproksymacja chi-kwadrat może być niepoprawna
```

```
##
##  Pearson's Chi-squared test
##
## data:  c_table
## X-squared = 29.627, df = 3, p-value = 1.653e-06
```

```
chisq.test(c_table, simulate.p.value = TRUE, B = 10000) # to świetna okazja, żeby zasymulować sobie!
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 10000
##  replicates)
##
## data:  c_table
## X-squared = 29.627, df = NA, p-value = 9.999e-05
```

### Wizualizacje danych z tabeli krzyżowej

Istnieją pewne bardziej wyrafinowane sposoby przedstawiania wyników testu niezależności $\chi^2$. Jeden z nich możemy stworzyć za pomocą funkcji `assoc` i `mosaic` z pakietu `vcd`. Mamy również bardzo podobne funkcję ze wbudowane w R - `assocplot` i `mosaicplot`.

```
# install.packages('vcd')
library(vcd)
```

```
## Ładowanie wymaganego pakietu: grid
```

```
assocstats(c_table)
```

```
##                     X^2 df   P(> X^2)
## Likelihood Ratio 23.265  3 3.5558e-05
## Pearson          29.627  3 1.6531e-06
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.175
## Cramer's V        : 0.178
```

```
# Przydatna funkcja
```

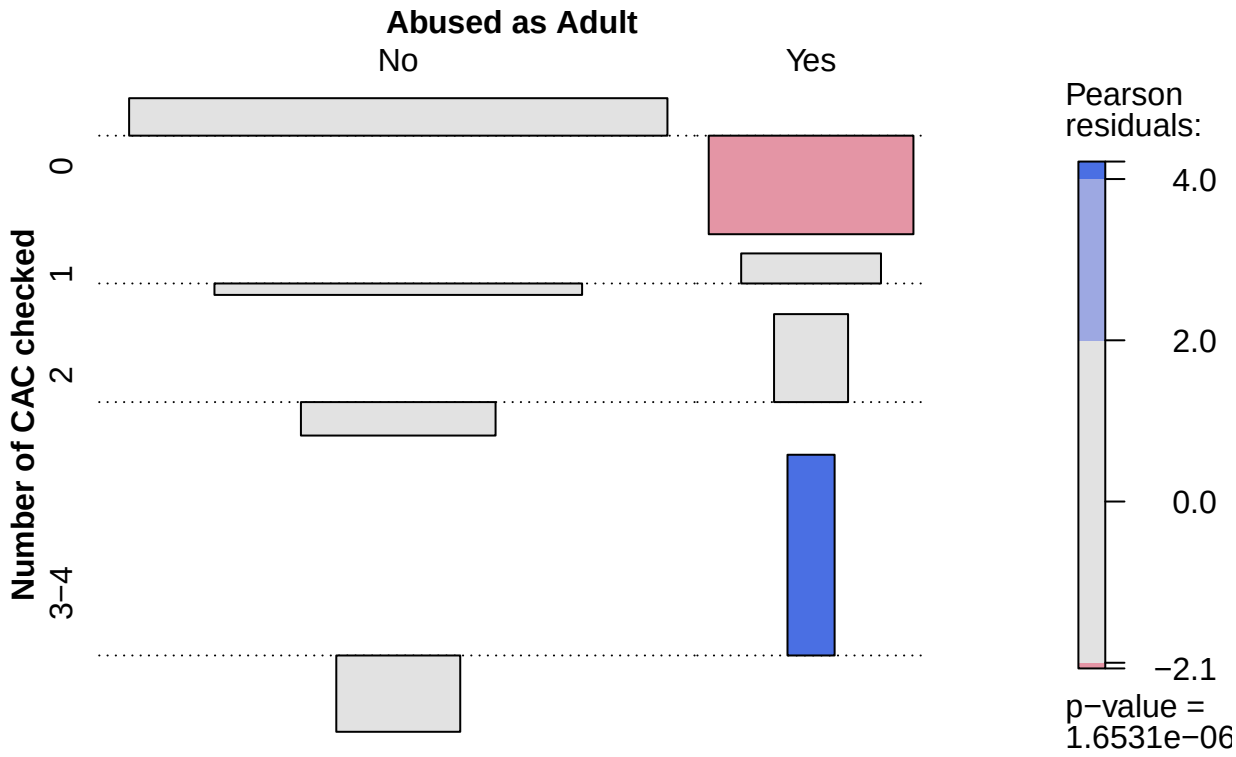
**`assoc` i `assocplot`** W tych funkcjach wysokość słupków odpowiada wystandaryzowanym rezuduom Paersona a więc $O - E/sqrt(E)$. Innymi słowy, im wyższy słupek, tym bardziej "zaskakujący" jest wynik dla danej kategorii (= proporcjonalnie bardziej odbiega od wartości oczekiwanej). Pozwala nam to ocenić "wkład" do statystyki testowej $\chi^2$ rozbieżności między wartością oczekiwaną i zaobserwowaną dla każdej kategorii.

Mówiąc prościej i na przykładzie - na podstawie tego wykresu mozemy stwierdzić, że za istotny wynik testu odpowiada przede wszystkim znacznie wyższa niż oczekiwana liczba wystąpień dla połączeni kategorii "Abused as Adult: yes" i "Number of CAC checked: 3-4".

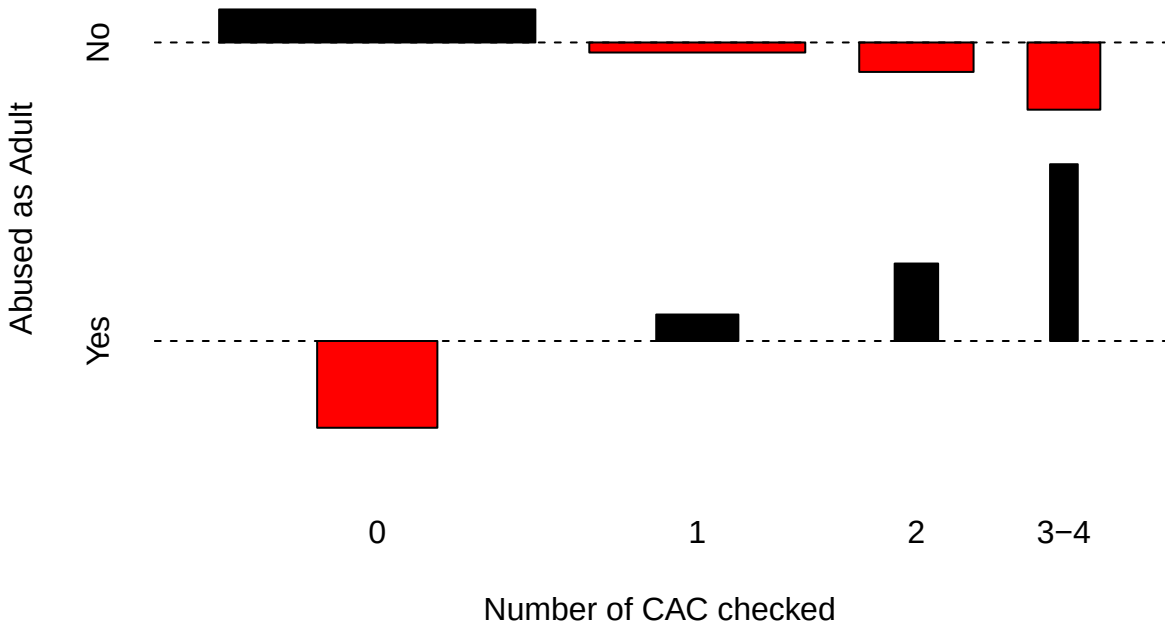
Szerokość słupka odpowiada pierwiastkowi z wartości oczekiwanej.

Powierzchnia każego prostokąta jest proporcjonalna do różnicy między wartościami teoretycznymi i empirycznymi

```
assoc(c_table, shade = TRUE, legend = TRUE)
```




```
assocplot(c_table)
```



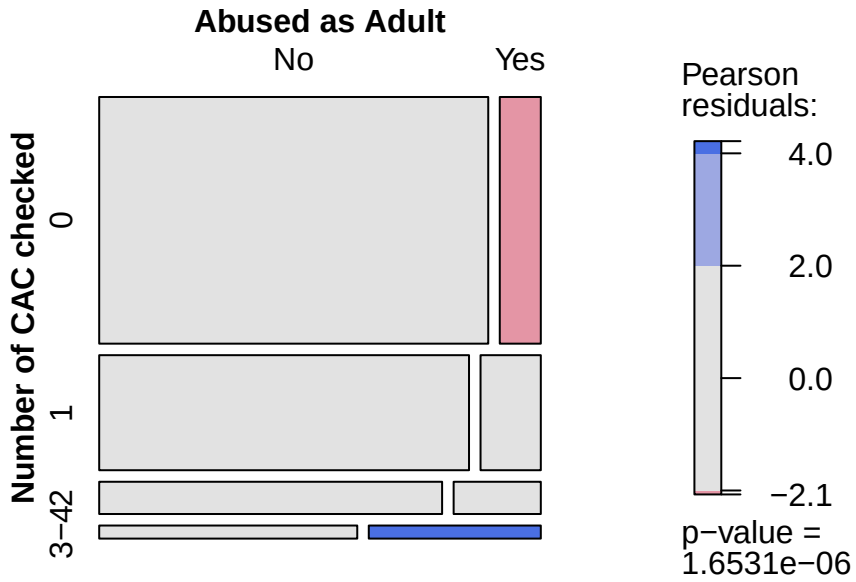

```
(c_table - exp_table)/sqrt(exp_table)
```

```
##                      Abused as Adult
## Number of CAC checked         No        Yes
##                   0    0.7873071 -2.0703712
##                   1   -0.2401177  0.6314344
##                   2   -0.7028053  1.8481579
##                   3-4 -1.6036255  4.2170333
```

```
# zobaczmy sobie te wystandaryzowane reszty Paersona
```

**`mosaic` i `mosaicplot`** Ten rodzaj wykresu jest koncepcyjnie znacznie prostszy. Pole prostokąta dla każdej kategorii odpowiada liczbie wystąpień w danej komórce tabeli. O tym wykresie można myśleć jako o standardowym procentowym wykresie słupkowym "na sterydach". W wersji z pakietu `vcd` domyślnie poszczególne prostokąty kolorowane są według standaryzowanych reszt Pearsona, we wbudowanej w R wersji domyślny schemat kolorowania jest nieco bardziej plebejski (ale też - mniej informatywny!).

```
mosaic(c_table, shade = TRUE, legend = TRUE)
```




```
mosaicplot(c_table, color = T)
```