Laboratorium 4

Martyna Konopacka

Zadanie 1

W tym zadaniu sprawdzamy, jak duża jest różnica, gdy konstruując przedział dla nieznanego σ użyjemy kwantyli rozkładu normalnego, zamiast t-Studenta. Przeprowadziłam najpierw eksperymenty z liczbą iteracji 200, jednak kiedy powtórzyłam je kilkukrotnie, zauważyłam duże wahania w wynikach wynikające z losowości i powtórzyłam eksperyment dla 1000 losowań. Sprawdziłam dodatkowo też próby rozmiaru $n = 50$.

```
glob_iter = 200
# Funkcja zwracające prawy przedziały ufności oparte na: 1.rozkładzie normalnym z szacowaniem sigma 2.r
interval1 <- function(X, opt = 1, lvl = 0.95){
  alpha <- 1 - lvl
  X_ <- mean(X)
  n <- length(X)
  s <- sd(X)
  if (opt == 1){a <- qnorm(1 - alpha/2)}
  if (opt == 2){a <- qt(1 - alpha/2, n-1)}
  expr <- a*s/sqrt(n)
  return (c(X_ - expr, X_ + expr))
}

# Funkcja losująca próbę rozmiaru n z rozkładu normalnego N(mu,si) iter razy i sprawdzająca jak często
experiment1 <- function(n = 10, opt = 1, iter = glob_iter, lvl = 0.95, mu = 0, si = 1){
  inside <- 0
  for (i in 1:iter){
    X <- rnorm(n, mu, si)
    interval <- interval1(X, opt, lvl)
    if (between(mu, interval[1], interval[2])){inside <- inside + 1}
  }
  return(data.frame(
    mu = mu,
    si = si,
    n = n,
    opt = opt,
    inside = inside/iter,
    lvl = lvl,
    iter = iter
  ))
}
```

Table 1: Podsumowanie wyników w zadaniu 1

mu	si	n	opt	inside	lvl	iter
0	1	10	1	0.935	0.95	200
0	1	10	2	0.945	0.95	200

mu	si	n	opt	inside	lvl	iter
0	1	50	1	0.950	0.95	200
0	1	50	2	0.915	0.95	200
0	1	100	1	0.915	0.95	200
0	1	100	2	0.950	0.95	200
0	1	10	1	0.909	0.95	1000
0	1	10	2	0.942	0.95	1000
0	1	50	1	0.945	0.95	1000
0	1	50	2	0.951	0.95	1000
0	1	100	1	0.940	0.95	1000
0	1	100	2	0.947	0.95	1000

Można zauważyć, że gdy n jest małe, przybliżenie rozkładem normalnym jest dużo gorsze niż rozkładem Studenta. W miarę ze wzrostem liczebności próby rozkłady coraz bardziej zbliżają się do siebie, więc błąd przybliżenia maleje.

Zadanie 2

W tym zadaniu zakładamy, że dane pochodzą z pewnej populacji - nie znamy więc parametru σ i używamy szacowania z użyciem rozkładu Studenta z zadania 1.

Table 2: Wyniki dla zmiennej IQ

lvl	low	high	width
0.99	104.9842	112.8619	7.877736
0.95	105.9535	111.8927	5.939191
0.90	106.4402	111.4060	4.965748
0.80	106.9953	110.8508	3.855486
0.70	107.3669	110.4792	3.112265
0.60	107.6610	110.1852	2.524249
0.50	107.9124	109.9337	2.021302

Table 3: Wyniki dla zmiennej TestPsych

lvl	low	high	width
0.99	53.24958	60.67350	7.423922
0.95	54.16301	59.76006	5.597051
0.90	54.62170	59.30138	4.679685
0.80	55.14485	58.77823	3.633382
0.70	55.49505	58.42803	2.932976
0.60	55.77212	58.15096	2.378834
0.50	56.00911	57.91397	1.904860

Gdy zwiększamy poziom istotności, zmniejszamy α i maleje ryzyko popełnienia błędu typu I, a przedział ufności zawęża się.

Zadanie 3

W tym zadaniu porównujemy dwa sposoby konstrukcji przedziału ufności dla frakcji: metodę klasyczną (opisaną w poprzednim raporcie) i metodę Agrestiego-Coulla. Wg. wykładu klasyczny przedział ufności może zawodzić, gdy liczba sukcesów jest bliska 0 lub n - tzn. gdy prawdziwy parametr frakcji p jest blisko 0 lub 1.

Przedział ufności Agrestiego-Coulla

Wprowadzamy następujące modyfikacje:

1. za środek przedziału zamiast $\frac{Y}{n}$ przyjmujemy $\tilde{p} = \frac{Y+0.5(u_{1-\frac{\alpha}{2}})^2}{n+(u_{1-\frac{\alpha}{2}})^2}$, gdzie Y oznacza liczbę sukcesów.
2. zamiast szacowanego jak poprzednio błędu standardowego użyjemy teraz $\sqrt{\frac{\tilde{p}(1-\tilde{p})}{n+(u_{1-\frac{\alpha}{2}})^2}}$

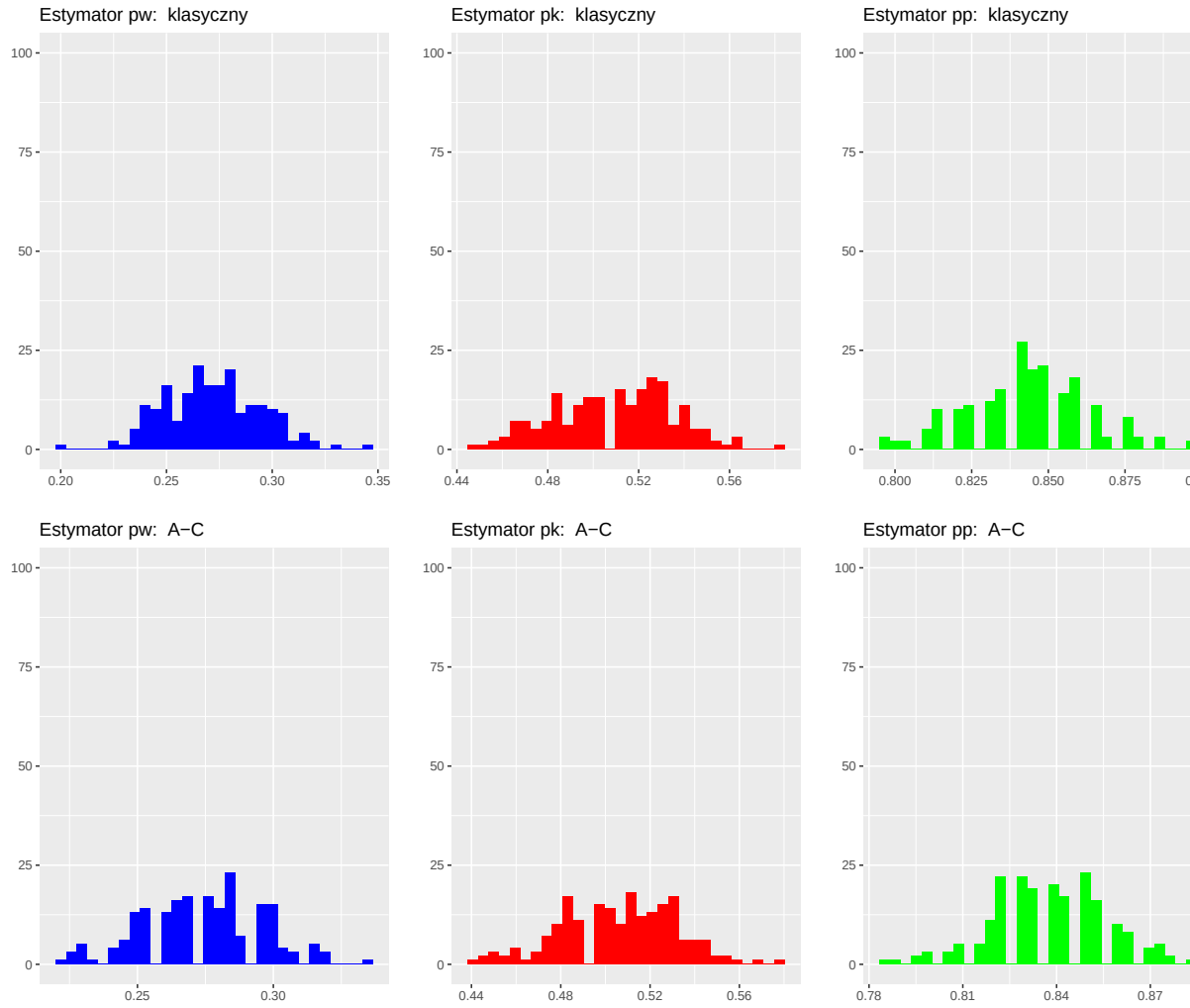
```
# Funkcja zwraca lewy i prawy brzeg przedziału ufności dla p; V powinien być wektorem TRUE / FALSE;
# opt: 1 - klasyczny, 2 - Agrestiego-Coulle
przedzial_p <- function(V, opt = 1, poziom = 0.95){
  n <- length(V)
  u <- qnorm(p = 1 - (1-poziom)/2)
  if (opt == 1){
    p <- sum(V)/length(V) # środek przedziału
    SE <- sqrt((p*(1-p)/n)) # błąd standardowy
  }
  if (opt == 2){
    Y <- sum(V)
    p <- (Y + 0.5 * u^2)/(n + u^2) # środek przedziału
    SE <- sqrt((p*(1-p)/(n+ 0.5 * u^2))) # błąd standardowy
  }
  return(c(p - u*SE, p+ u*SE))
}

# Funkcja zwraca estymator punktowy p w oparciu o wektor V (TRUE/FALSE); opt: 1 - klasycznie, 2 - Agres
estim_p <- function(V, opt = 1, lvl = 0.95){
  n = length(V)
  if (opt == 1){
    return(sum(V)/n)
    }
  else if (opt == 2){
    u = qnorm(1 - (1-lvl)/2)
    return ( (sum(V) + 0.5 * u^2)/(n + u^2) )
}
}
```

Table 4: Porównanie wyników w zadaniu 3

estymator	n	iter	pw_width	pk_width	pp_width	pw_percent	pk_percent	pp_percent
klasyczny	200	200	24.62719	27.67577	20.09938	0.99	0.995	0.990
A-C	200	200	24.56195	27.55005	20.30382	1.00	0.995	0.995

Oba estymatory dają porównywalne wyniki. Poniżej histogramy estymatorów punktowych otrzymanych



dwoma sposobami.

Dodatkowe eksperymenty do zadania 3

Przeprowadziłam dodatkowe eksperymenty w celu sprawdzenia, czy różnica pomiędzy metodami będzie widoczna w testach na próbie, dla której liczba sukcesów jest bliska 0 lub n.

```
experiment2b <- function(real_p = 0.5, opt = 1, n = 200, iter = 1000, lvl = 0.95){
  name <- ifelse(opt == 1, 'klasyczny', 'A-C')
  # Funkcja zwraca listę z histogramem i dataframe wyników jak w zadaniu 3 ale dla prób z rozkładu Bern
  # wektor na estymatory punktowe
  p <- numeric(iter)

  # liczniki jak często przedział zawiera prawdziwy parametr
  p_count <- 0

  # skumulowane szerokości przedziałów
  p_w <- 0

  for (i in 1:iter){
    X <- rbernoulli(n, real_p)
```

```
    # zapisanie estymatora punktowego
    p[i] <- estim_p(X, opt, lvl)

    # wyznaczanie przedziałów
    I <- przedzial_p(X, opt, lvl)

    # dodanie ich szerokości
    p_w <- p_w + I[2] - I[1]

    # sprawdzenie czy zawierają prawdziwy parametr (frakcję w całym zbiorze income)
    if (between(real_p, I[1], I[2])){p_count <- p_count + 1}

  }
  # tworzenie histogramu
  hist <- qplot(p, xlab = '', geom = 'histogram', main = paste('Estymator p: ', name), fill = I('Blue'))

  # tworzenie podsumowania
  summar <- data.frame(
    p = real_p,
    lvl = lvl,
    estymator = name,
    n = n,
    iter = iter,
    mean_width = mean(p_w),
    p_percent = sum(p_count)/iter
  )
  return (summar = summar)
}

results_extr <- rbind(experiment2b(0.005), experiment2b(0.005, opt = 2), experiment2b(0.01), experiment2
                      experiment2b(0.01, opt = 2), experiment2b(0.02, opt = 2), experiment2b(0.1, opt =

results_extr[order(results_extr$p), ] %>% mutate(rel_difference = (lvl - p_percent)/lvl) %>% knitr::kabl
```

Table 5: Dodatkowe eksperymenty w zadaniu 3 - wyniki

p	lvl	estymator	n	iter	mean_width	p_percent	rel_difference
0.005	0.95	klasyczny	200	1000	15.35290	0.642	0.3242105
0.005	0.95	A-C	200	1000	32.26147	0.987	-0.0389474
0.010	0.95	klasyczny	200	1000	25.03143	0.873	0.0810526
0.010	0.95	A-C	200	1000	37.33599	0.982	-0.0336842
0.020	0.95	klasyczny	200	1000	37.38021	0.924	0.0273684
0.020	0.95	A-C	200	1000	45.85342	0.976	-0.0273684
0.100	0.95	klasyczny	200	1000	82.22072	0.920	0.0315789
0.100	0.95	A-C	200	1000	84.70513	0.966	-0.0168421
0.900	0.95	klasyczny	200	1000	82.54917	0.925	0.0263158
0.900	0.95	A-C	200	1000	84.58342	0.958	-0.0084211
0.950	0.95	klasyczny	200	1000	59.28326	0.926	0.0252632
0.950	0.95	A-C	200	1000	64.33240	0.965	-0.0157895
0.990	0.95	klasyczny	200	1000	24.78035	0.865	0.0894737
0.990	0.95	A-C	200	1000	37.35281	0.976	-0.0273684

W ostatniej kolumnie tabeli widać błąd względny pomiędzy odsetkiem parametrów w przedziale teoretycznym, a wyznaczonym. Dla skrajnych wartości p (tj. mniejszych od 0.01 lub większych od 0.99) klasyczna metoda rzeczywiście daje wyniki wyraźnie gorsze niż metoda Agrestiego-Coull'a. Dodatkowo można zauważyć, że wszystkie ujemne wartości w tej kolumnie pochodzą z estymacji metodą A-C (są to przypadki, gdzie oszacowany przedział zawiera więcej obserwacji prawdziwego parametru niż powinien, tzn. jest za szeroki.)