

1 Laboratorinis darbas

5 grupė: Arnas Kazanzvičius, Arnas Usonis, Lukas Janušauskas, Simonas Lapinskas

Skirstinys: χ^2

1 užduotis

Chi kvadrato skirstinys (žymimas $\chi^2(k)$) yra tikimybinis skirstinys, kuris atsiranda sumuojant k Nepriklausomų normaliųjų kintamųjų kvadratai (kai vidurkis 0, o dispersija 1).

Matematiškai, jei Z_1, Z_2, \dots, Z_k yra nepriklausomi standartiniai normalieji atsitiktiniai dydžiai (tai tokie tokie dydžiai kurių vidurkis 0 ir dispersija 1): $Z_i \sim N(0, 1)$ tada jų kvadratų suma seka chi kvadrato skirstinį su k laisvės laipsniais:

$$X = \sum_{i=1}^k Z_i^2 \sim \chi^2(k) \quad (1)$$

Taikymas

Chi kvadrato skirstinys yra labai svarbus statistikoje, ypač hipotezių tikrinimui ir dispersijos analizei.

1. Chi kvadrato nepriklausomumo testas

Naudojamas nustatyti, ar duomenų lentelės kintamieji yra nepriklausomi.

Pvz., ar žmonių rūkymo įpročiai priklauso nuo jų lyties?

2. Gerumo įvertinimo testas

Tikrina, ar stebėti duomenys atitinka teorinį pasiskirstymą.

Pvz., ar kauliuko metimo rezultatai atitinka tolygų pasiskirstymą?

3. Dispersijos analizė

Naudojamas pasikliautiniams intervalams populiacijos dispersijai įvertinti.

Pvz., ar skirtingose mokyklose mokinių pažymių dispersija yra vienoda?

2 užduotis

Tikimybės tankio funkcija:

$$f(x; k) = \begin{cases} \frac{x^{k/2-1} e^{-x/2}}{2^{k/2} \Gamma\left(\frac{k}{2}\right)}, & x > 0; \\ 0, & \text{kitais atvejais.} \end{cases} \quad (2)$$

Kur $\Gamma(k/2)$ reiškia gamma funkciją, kuri turi uždaras formas (užbaigtas išraiškas) reikšmėms sveikaisiais k .

Kumuliatyvi pasiskirstymo funkcija:

$$F(x; k) = \frac{\gamma\left(\frac{k}{2}, \frac{x}{2}\right)}{\Gamma\left(\frac{k}{2}\right)} = P\left(\frac{k}{2}, \frac{x}{2}\right) \quad (3)$$

Kur $\gamma(s, t)$ yra žemesnė nepilna gamma funkcija, o $P(s, t)$ yra normalizuota nepilna gamma funkcija.

3 užduotis

$$\text{Vidurkis: } \mu = k \quad (4)$$

$$\text{Dispersija: } \sigma^2 = 2k \quad (5)$$

$$\text{Asimetrijos koeficientas: } \gamma_1 = \frac{2}{\sqrt{k}} \quad (6)$$

$$\text{Eksceso koeficientas: } \gamma_2 = \frac{6}{k} \quad (7)$$

4 užduotis

Brėžiame tankio funkcijų grafiką.

```
library(ggplot2)

x <- seq(0, 50, 0.1)

y1 <- dchisq(x, df = 4)
y2 <- dchisq(x, df = 8)
y3 <- dchisq(x, df = 16)

df1 <- data.frame(x = x, y = y1, df = 4)
df2 <- data.frame(x = x, y = y2, df = 8)
df3 <- data.frame(x = x, y = y3, df = 16)

df <- rbind(df1, df2, df3)

ggplot(data=df, aes(x = x, y = y, color = as.factor(df)))+
  geom_line(linewidth=1)+
  scale_color_discrete(name = "Laisvės laipsnis")+
  labs(y = "Tikimybių tankis",
       x = "Reikšmės")+
  theme_minimal()

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybių tankis' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybių tankis' in 'mbcsToSbcs': dot substituted for
## <b3>
```

```

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <a1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <97>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <a1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <97>

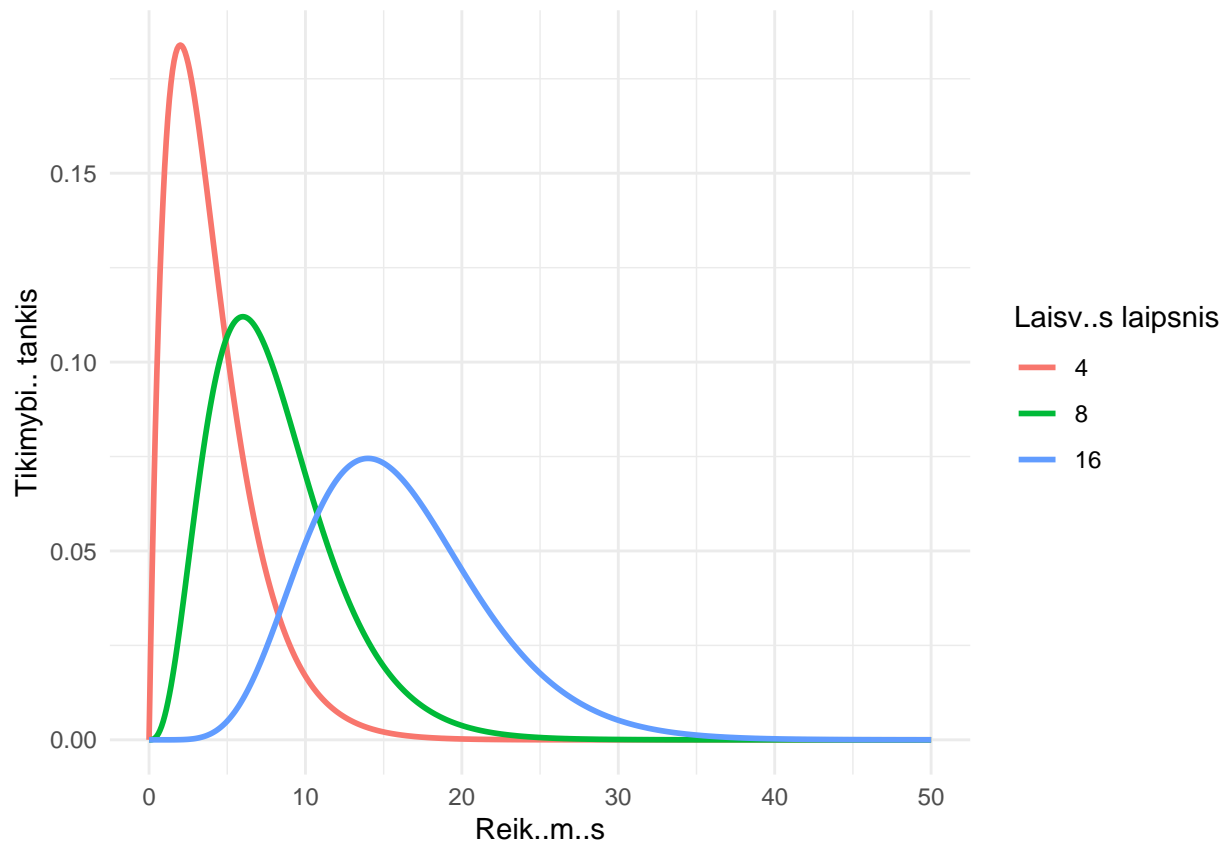
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybių tankis' in 'mbcsToSbcs': dot substituted for
## <c5>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybių tankis' in 'mbcsToSbcs': dot substituted for
## <b3>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>

```



Brėžiame pasiskirstymo funkcijų grafiką.

```
y1 <- pchisq(x, df = 4)
y2 <- pchisq(x, df = 8)
y3 <- pchisq(x, df = 16)

df1 <- data.frame(x = x, y = y1, df = 4)
df2 <- data.frame(x = x, y = y2, df = 8)
df3 <- data.frame(x = x, y = y3, df = 16)

df <- rbind(df1, df2, df3)

ggplot(data=df, aes(x = x, y = y, color = as.factor(df)))+
  geom_line(linewidth=1)+
  scale_color_discrete(name = "Laisvės laipsnis")+
  labs(y = "komuliatyvi tikimybė",
       x = "Reikšmės")+
  theme_minimal()
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <c4>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
```

```

## <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'komuliatyvi tikimybė' in 'mbcsToSbcs': dot substituted
## for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'komuliatyvi tikimybė' in 'mbcsToSbcs': dot substituted
## for <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <a1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <97>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <a1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Reikšmės' in 'mbcsToSbcs': dot substituted for <97>

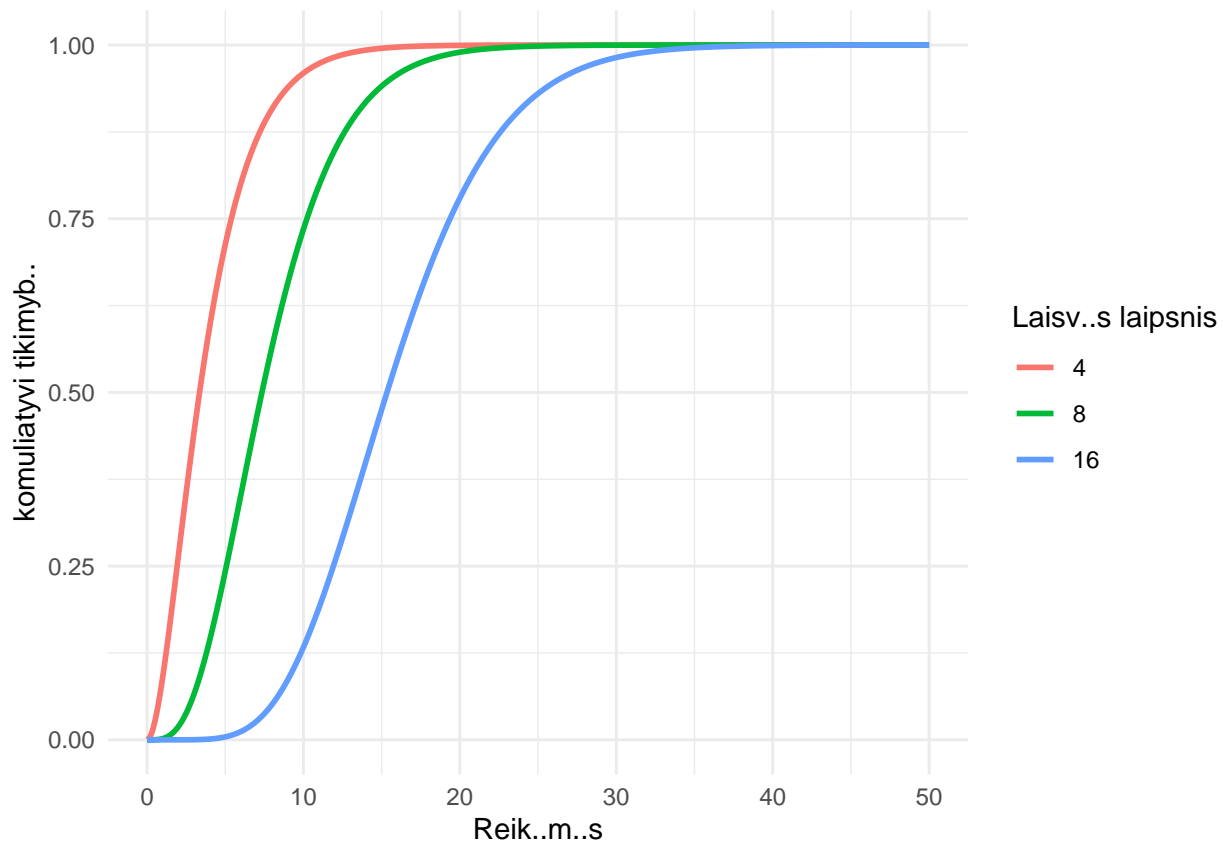
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'komuliatyvi tikimybė' in 'mbcsToSbcs': dot substituted
## for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'komuliatyvi tikimybė' in 'mbcsToSbcs': dot substituted
## for <97>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsnis' in 'mbcsToSbcs': dot substituted for
## <97>

```



Didėjant laisvės laipsniui, chi kvadrato skirstinys vis labiau primena normaliąjį skirstinį. Esant mažesniai laisvės laipsnių skaičiui, skirstinys yra labiau asimetriškas.

5 užduotis

Kvantilių funkcija

$$Q(p) = F^{-1}(p) \quad (8)$$

čia:

$Q(p)$ – kvantilis tam tikram tikimybės lygiui p , $F^{-1}(p)$ – atvirkštinė chi kvadrato pasiskirstymo funkcija.

Grafikas

```
p_values <- seq(0.01, 0.99, 0.01)

df_values <- c(4, 8, 16)

df_list <- lapply(df_values, function(df) {
  data.frame(p = p_values, q = qchisq(p_values, df), df = as.factor(df))
})

df <- do.call(rbind, df_list)

ggplot(df, aes(x = p, y = q, color = df)) +
  geom_line(linewidth = 1) +
  scale_color_discrete(name = "Laisvės laipsniai") +
```

```
labs(title = "Chi kvadrato kvantilių funkcija ",
      x = "Tikimybė (p)",
      y = "Kvantilis Q(p)") +
theme_minimal()
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Chi kvadrato kvantilių funkcija ' in 'mbcsToSbcs': dot
## substituted for <c5>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Chi kvadrato kvantilių funkcija ' in 'mbcsToSbcs': dot
## substituted for <b3>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybė (p)' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybė (p)' in 'mbcsToSbcs': dot substituted for <97>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybė (p)' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Tikimybė (p)' in 'mbcsToSbcs': dot substituted for <97>

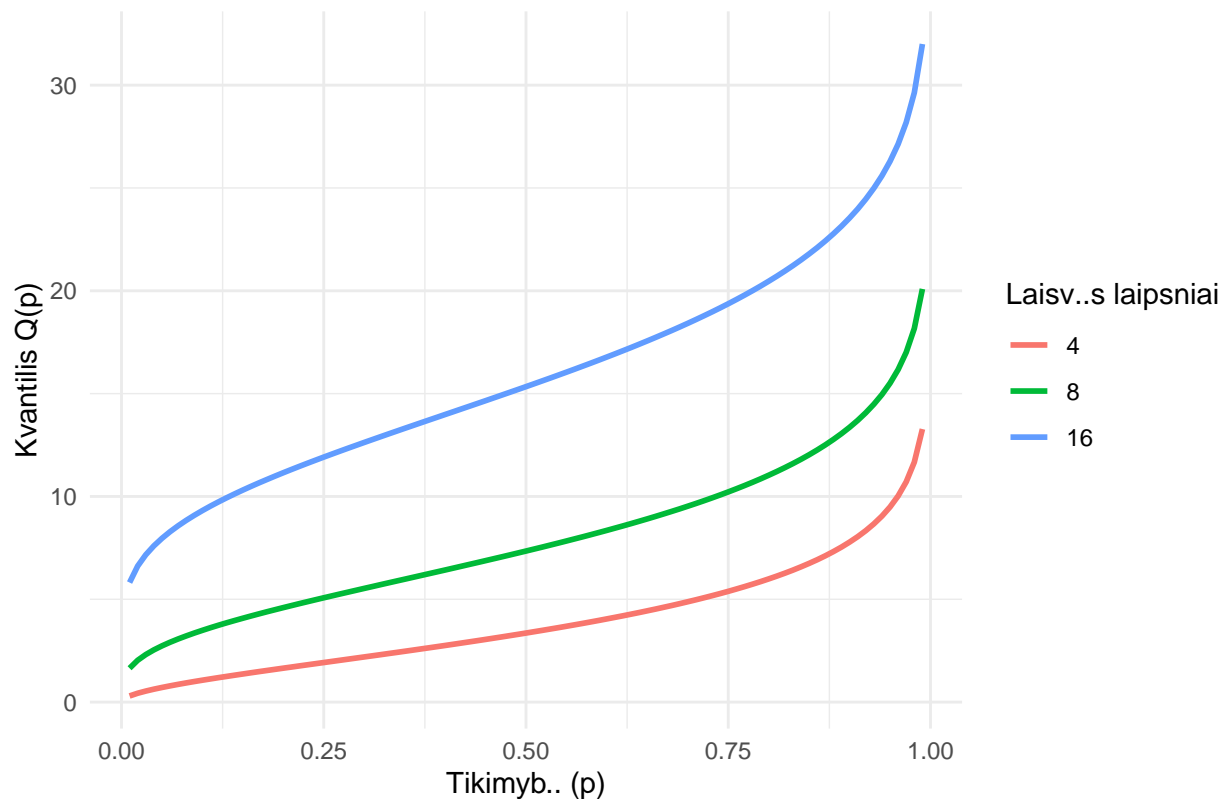
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Laisvės laipsniai' in 'mbcsToSbcs': dot substituted for
## <97>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Chi kvadrato kvantilių funkcija ' in 'mbcsToSbcs': dot
## substituted for <c5>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Chi kvadrato kvantilių funkcija ' in 'mbcsToSbcs': dot
## substituted for <b3>
```

Chi kvadrato kvantili.. funkcija



Chi kvadrato pasiskirstymas su didesniu laisvės laipsniu tampa vis panašesnis į normalųjį pasiskirstymą, kuomet didesnis laisvės laipsnis, tuo kvantiliai lėčiau auga ir įgauna vis simetriškesnę formą.

6. Fiksavome, pasirinktą a.d. parametrų rinkinį ($k=5$). Sugeneravome χ_5^2 duomenų rinkinius su 20, 50, 200, 1000 imčių dydžiais.

```
k <- 5
n <- c(20, 50, 200, 1000)

set.seed(42)

imtis1 <- rchisq(n[1], k)
imtis2 <- rchisq(n[2], k)
imtis3 <- rchisq(n[3], k)
imtis4 <- rchisq(n[4], k)
```

6a) Nubrėžėme histogramas

```
# Apibrėžiame pagalbinę funkciją, kadangi grafikai labai panašūs
plot_chisq_sample <- function(sample) {
  # sample - imtis, kurią įdedame į funkciją

  # Sudarome pavadinimą, į kurį įeis imties dydis
  imties_dydis <- length(sample)
  pavadinimas <- paste0("Imties dydis: ", imties_dydis)

  # Nubrėžiame histogramą
  hist(sample, main=pavadinimas,
        xlab = "X", ylab = "",
```



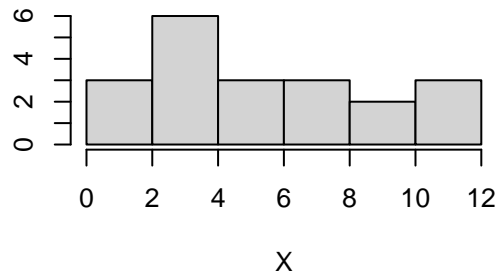
```

    freq = TRUE)
}

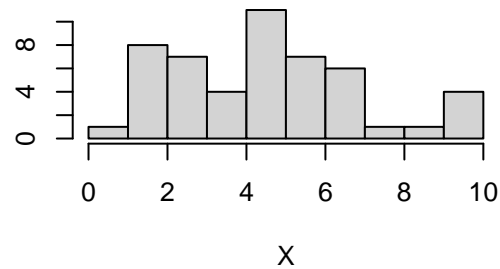
par(mfrow=c(2,2))
plot_chisq_sample(imtis1)
plot_chisq_sample(imtis2)
plot_chisq_sample(imtis3)
plot_chisq_sample(imtis4)

```

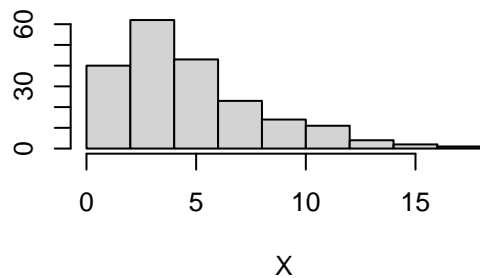
Imties dydis: 20



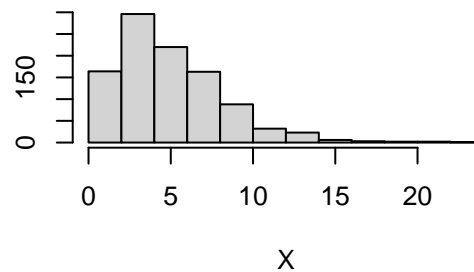
Imties dydis: 50



Imties dydis: 200



Imties dydis: 1000



6b) Empirinės pasiskirstymo funkcijos mūsų darbe imčių pasiskirstymo funkcijos pateikiamos su teorine pasiskirstymo funkcija.

```

nubreztai_chisq_empirini <- function(sample, df=5) {
  # sample - imtis.
  # df - chi kvadratu parametras

  # Sudarome pavadinimą
  imties_dydis <- length(sample)
  pavadinimas <- paste0("Imties dydis: ", imties_dydis)

  # Nubrėžiame empirinę pasiskirstymo funkciją
  plot(ecdf(sample), main=pavadinimas)

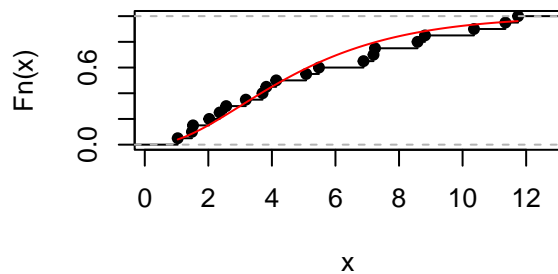
  # Nubrėžiame teorinę pasiskirstymo funkciją
  x <- seq(min(sample), max(sample), by=0.01)
  lines(x, pchisq(x, df), col = 'red')
}

par(mfrow=c(2,2))

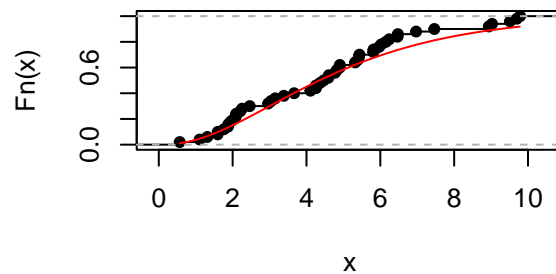
```

```
suppressWarnings({
  nubrežti_chisq_empirini(imtis1)
  nubrežti_chisq_empirini(imtis2)
  nubrežti_chisq_empirini(imtis3)
  nubrežti_chisq_empirini(imtis4)
})
```

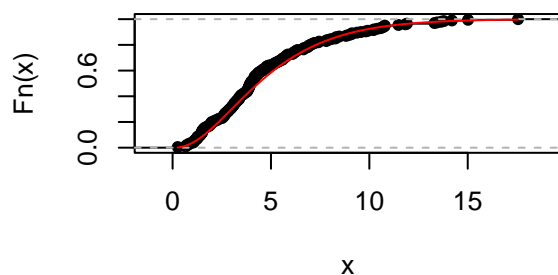
Imties dydis: 20



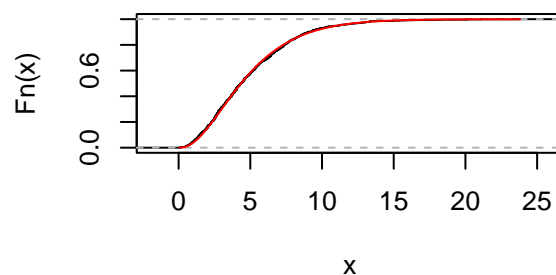
Imties dydis: 50



Imties dydis: 200



Imties dydis: 1000



Nubrėžę empirines pasiskirstymo funkcijas pastebime, kad didesnių imčių empirinės pasiskirstymo funkcijos geriau aproksimuoja tikrąją p.f.

7. Momentų metodu išvestas parametro k įvertinys:

Remiantis 3. punktu prisimename, kad $EX = k$. Pirmasis žingsnis, sudarant įvėčius momentų metodu yra momentų prilyginimas empiriniams momentams. Taigi EX prilyginame \bar{X} . Gauname parametro k įvertinį \tilde{k} :

$$\tilde{k} = \bar{X} \quad (9)$$

```
imtys <- list(imtis1, imtis2, imtis3, imtis4)

sapply(imtys, function(x)
  paste(length(x), "dydžio imties parametro įvertinys", mean(x)))
```

```
## [1] "20 dydžio imties parametro įvertinys 5.42851954560543"
## [2] "50 dydžio imties parametro įvertinys 4.49953597150517"
## [3] "200 dydžio imties parametro įvertinys 4.82725161603656"
## [4] "1000 dydžio imties parametro įvertinys 4.99404810200757"
```

Vėl, kuo didesnė imtis, tuo geriau aproksimuojame skirstinio parametą k .

8. Didžiausio tikėtinumo metodu taip pat gavome įvėčius. Tačiau, kadangi skaičiavimai per daug komplikuoti, naudojome optim funkcija.

```

# install.packages('likelihoodExplore')
library(likelihoodExplore)

pakoreguota_tiketinumo <- function(x, par) {
  # Pakoreguojame tikėtinumo funkciją, kad tikėtų optim funkcijai
  return( -1 * likchisq(x=x, df=par) )
}

mle_chisq_ivertis <- function(imtis) {
  res <- optim(par=c(1),                # Pradedame nuo 1
               fn=pakoreguota_tiketinumo, # Pakoreguojame tikėtinumo funkciją
               method="L-BFGS-B",        # Naudojame L-BFGS optimizatorių
               x=imtis)
  return ( res$par )
}

sapply(imtys, function(x)
  paste(length(x), "dydžio imties parametro įvertinys", mle_chisq_ivertis(x)))

## [1] "20 dydžio imties parametro įvertinys 5.30031169143056"
## [2] "50 dydžio imties parametro įvertinys 4.76282529559125"
## [3] "200 dydžio imties parametro įvertinys 4.79013509203223"
## [4] "1000 dydžio imties parametro įvertinys 4.93233518543483"

```

9. Palyginkime 7 ir 8 punktuose gautus rezultatus

Nors, davus mažą imtį, gauti geresni rezultatai, panaudojus didžiausio tikėtinumo metodą, tačiau momentų metodo įverčiai, iš rezultatų atrodo, greičiau konverguoja link tikrojo parametro(5). Be abejo, abejais atvejais kuo didesnė imtis, tuo geresnė parametro aproksimacija.

10. Parametrų patikimumo intervalai

Iš χ^2 apibrėžimo žinome, kad a.d. $X \sim \chi_k^2$, jei $X = \sum_{i=1}^k N_i$, kur $N_i \sim N(0, 1)$. Todėl, n dydžio imties empirinio vidurkio skirstinį nustatome štai taip:

$$n\bar{X} = \sum_{j=1}^n X_j = \sum_{j=1}^n \sum_{i=1}^k N_{ji} \quad (10)$$

Kadangi kiekvienas N_{ji} yra pasiskirstęs pagal standartinį normalųjį skirstinį, tai vėl gauname χ^2 skirstinį:

$$n\bar{X} \sim \chi_{nk}^2 \quad (11)$$

Taigi, pasikliautinio intervalo skaičiavimas (čia x_p - p-tasis χ_{nk}^2 kvartilis):

$$P(x_{(1-\alpha)/2} < n\hat{k} < x_{(1+\alpha)/2}) \quad (12)$$

$$P\left(\frac{x_{(1-\alpha)/2}}{n} < \hat{k} < \frac{x_{(1+\alpha)/2}}{n}\right) \quad (13)$$

```

confidence_interval_chisq <- function(sample, alpha) {
  lower <- (1 - alpha) / 2
  upper <- (1 + alpha) / 2

  lower_b <- qchisq(lower, sum(sample)) / length(sample)

```

```

upper_b <- qchisq(upper, sum(sample)) / length(sample)

return( c(lower_b, upper_b) )
}

lapply(imtys, function(x)
  paste(length(x), "dydžio imties parametro pasikliautinio intervalo rėžiai",
    confidence_interval_chisq(x, 0.9)))

## [[1]]
## [1] "20 dydžio imties parametro pasikliautinio intervalo rėžiai 4.27607029946644"
## [2] "20 dydžio imties parametro pasikliautinio intervalo rėžiai 6.69455758007594"
##
## [[2]]
## [1] "50 dydžio imties parametro pasikliautinio intervalo rėžiai 3.8251888171799"
## [2] "50 dydžio imties parametro pasikliautinio intervalo rėžiai 5.21934234370739"
##
## [[3]]
## [1] "200 dydžio imties parametro pasikliautinio intervalo rėžiai 4.47163450665412"
## [2] "200 dydžio imties parametro pasikliautinio intervalo rėžiai 5.19423774046643"
##
## [[4]]
## [1] "1000 dydžio imties parametro pasikliautinio intervalo rėžiai 4.83080552588767"
## [2] "1000 dydžio imties parametro pasikliautinio intervalo rėžiai 5.15956468681696"

```

Matome, kad 50 objektų imtyje patikimumo intervalas pasislinko į kairę. Nors didesnės imties patikimumo intervalo apatinis rėžis yra toliau, turime turėti omenyje, kad vidurkis yra mažesnis, todėl ne patikimumo intervalas yra platesnis, bet. Vietoj to reikėtų pažiūrėti į aplitiudę intervalo.

11. Apskaičiavome 6 dalyje gautų duomenų rinkinių su 20, 50, 200, 1000 imčių dydžiais kvartilius.

```

imtis1_kvartiliai <- quantile(imtis1, c(0.25, 0.5, 0.75))
imtis2_kvartiliai <- quantile(imtis2, c(0.25, 0.5, 0.75))
imtis3_kvartiliai <- quantile(imtis3, c(0.25, 0.5, 0.75))
imtis4_kvartiliai <- quantile(imtis4, c(0.25, 0.5, 0.75))

print(list(imtis1_kvartiliai, imtis2_kvartiliai, imtis3_kvartiliai, imtis4_kvartiliai))

## [[1]]
##      25%      50%      75%
## 2.510790 4.607263 7.579767
##
## [[2]]
##      25%      50%      75%
## 2.219394 4.528022 5.926666
##
## [[3]]
##      25%      50%      75%
## 2.638602 3.988445 6.260006
##
## [[4]]
##      25%      50%      75%
## 2.608217 4.332031 6.746946

```

12. Prie 6 punkte gautų duomenų rinkinių pridėjome po 5 išskirtis.

Visų pirma paskaičiuokime apskaičiuokime išskirčių klasifikavimo ribas, remiantis kvartilų metodu:

```
imtis1_ribos <- c(imtis1_kvartiliai[1] - 3 * IQR(imtis1),
                 imtis1_kvartiliai[3] + 3 * IQR(imtis1))

imtis2_ribos <- c(imtis2_kvartiliai[1] - 3 * IQR(imtis2),
                 imtis2_kvartiliai[3] + 3 * IQR(imtis2))

imtis3_ribos <- c(imtis3_kvartiliai[1] - 3 * IQR(imtis3),
                 imtis3_kvartiliai[3] + 3 * IQR(imtis3))

imtis4_ribos <- c(imtis4_kvartiliai[1] - 3 * IQR(imtis4),
                 imtis4_kvartiliai[3] + 3 * IQR(imtis4))

print(list(imtis1_ribos, imtis2_ribos, imtis3_ribos, imtis4_ribos))
```

```
## [[1]]
##      25%      75%
## -12.69614  22.78670
##
## [[2]]
##      25%      75%
## -8.902421  17.048482
##
## [[3]]
##      25%      75%
## -8.22561  17.12422
##
## [[4]]
##      25%      75%
## -9.807969  19.163133
```

Kad galėtume atpažinti vėliau, patikrinkime, duomenų amplitudes

```
sapply(imtys,
       function(x) paste(length(x),
                          "imties dyio amplitudė:",
                          min(x), "-", max(x))
       )
```

```
## [1] "20 imties dyio amplitudė: 1.03617574402698 - 11.7459815259314"
## [2] "50 imties dyio amplitudė: 0.567393736980694 - 9.7863797767188"
## [3] "200 imties dyio amplitudė: 0.260054426489983 - 17.5581190135009"
## [4] "1000 imties dyio amplitudė: 0.0798479131768038 - 23.814071833844"
```

```
imtis1_isskirtys <- c(imtis1, 25, 30, 35, 40, 43)
imtis2_isskirtys <- c(imtis2, 18, 20, 21, 23, 24)
```

```
# Kadangi jau pačiose imtyse yra išskirčių(matome iš amplitudžių ir kvartilų testo)
# Paimame didesnes išskirtis, kad lengviau išskirtume, ką pridėjome, o kas jau buvo
imtis3_isskirtys <- c(imtis3, 30, 34, 39, 42, 43)
imtis4_isskirtys <- c(imtis4, 35, 39, 42, 46, 48)
```

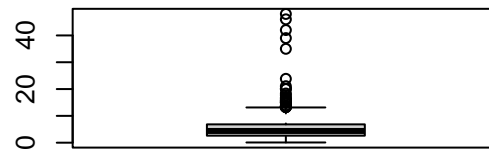
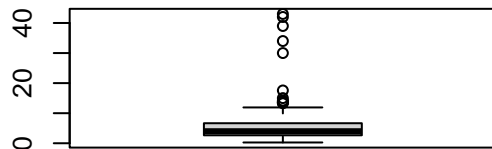
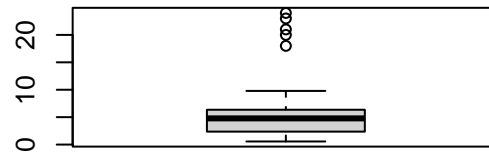
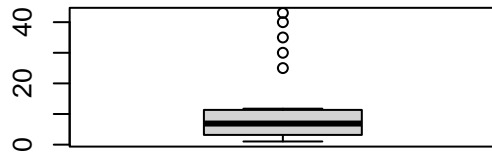
13. Nubraižėme po stačiakampę diagramą kiekvienam 12 dalies duomenų rinkiniui

```
par(mfrow=c(2,2))
```

```

boxplot(imtis1_isskirtys)
boxplot(imtis2_isskirtys)
boxplot(imtis3_isskirtys)
boxplot(imtis4_isskirtys)

```



akampės diagramos identifikuoja įrašytas išskirtis.

14. Grafiškai palyginome 6 punkte gautas histogramas su normaliojo skirstinio tankiu.

Kad galėtume grafiškai palyginti, mums reikia x ašių. Jas gauti pasitelkėme funkciją `seq`.

```

imtis1_seq <- seq(min(imtis1), max(imtis1))
imtis2_seq <- seq(min(imtis2), max(imtis2))
imtis3_seq <- seq(min(imtis3), max(imtis3))
imtis4_seq <- seq(min(imtis4), max(imtis4))

```

```

par(mfrow=c(2,2))
hist(imtis1, freq = FALSE)
lines(dnorm(imtis1_seq, mean = mean(imtis1),
            sd = sd(imtis1)), type = "l", col = "red")

hist(imtis2, freq = FALSE)
lines(dnorm(imtis2_seq, mean = mean(imtis2),
            sd = sd(imtis2)), type = "l", col = "red")

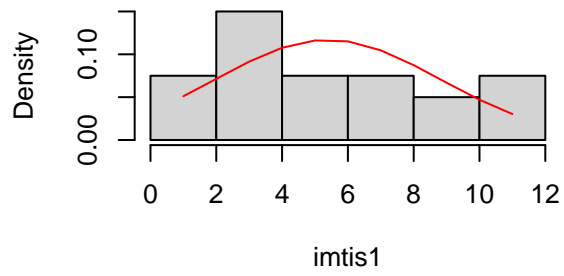
hist(imtis3, freq = FALSE)
lines(dnorm(imtis3_seq, mean = mean(imtis3),
            sd = sd(imtis3)), type = "l", col = "red")

hist(imtis4, freq = FALSE)
lines(dnorm(imtis4_seq, mean = mean(imtis4),
            sd = sd(imtis4)), type = "l", col = "red")

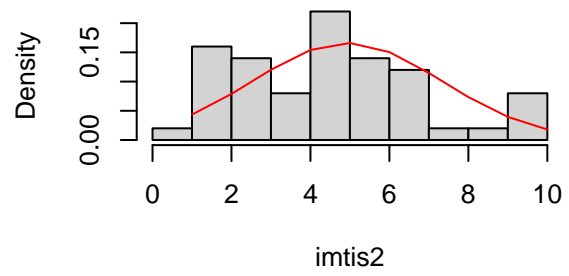
```

Stači-

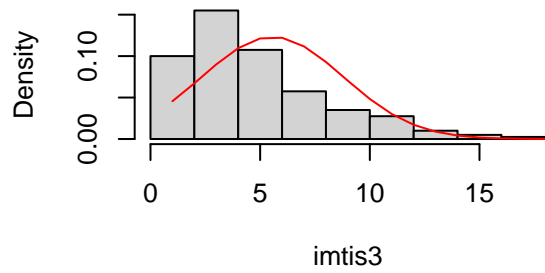
Histogram of imtis1



Histogram of imtis2



Histogram of imtis3



Histogram of imtis4

