

# 1 Laboratorinis darbas

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

Skirstinys:  $\chi^2$

## 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

**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  $k$ .

**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 uždutis

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

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

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

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

### 4 uždutis

Brėžiame tankio funkcijų grafiką.

```
library(ggplot2)
# install.packages("latex2exp")
library(latex2exp)

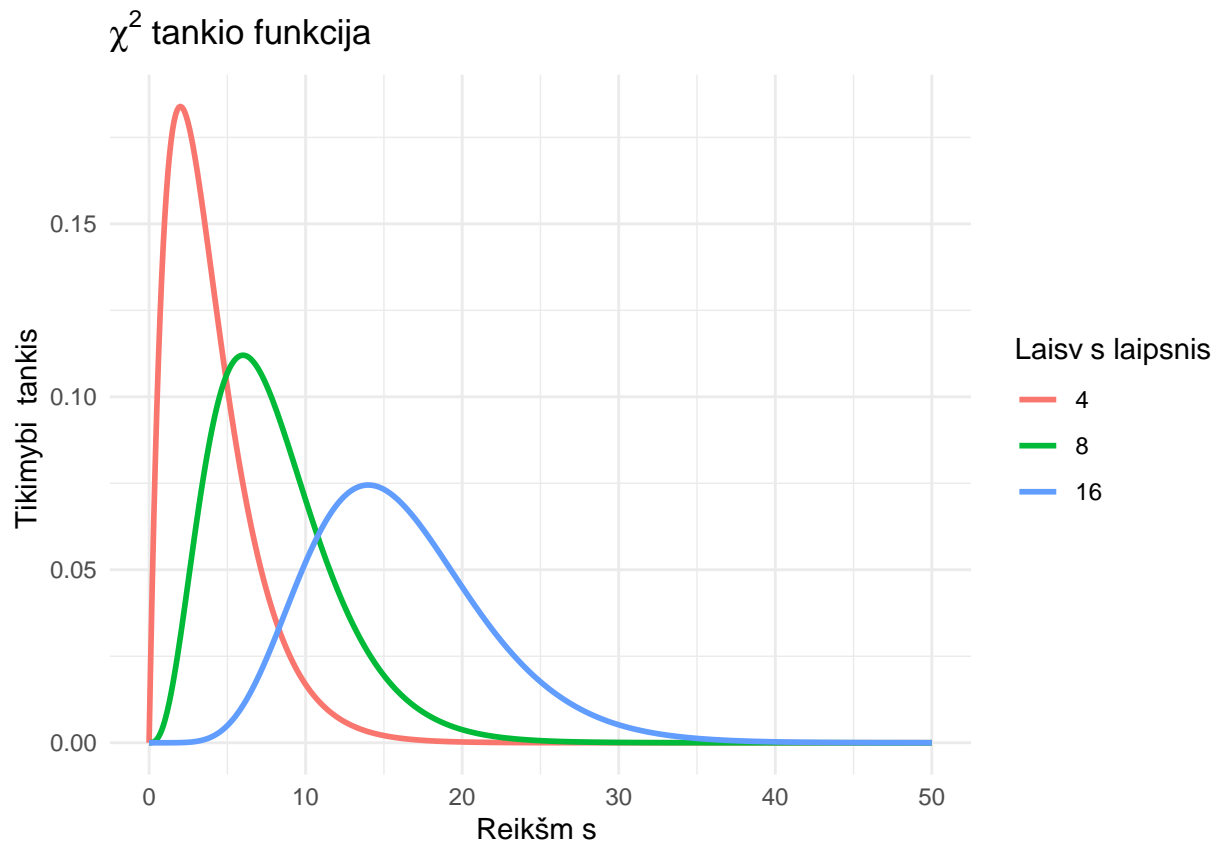
x <- seq(0, 50, 0.1)

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() +
  ggtitle(TeX("$\\chi^2$ tankio funkcija"))
```



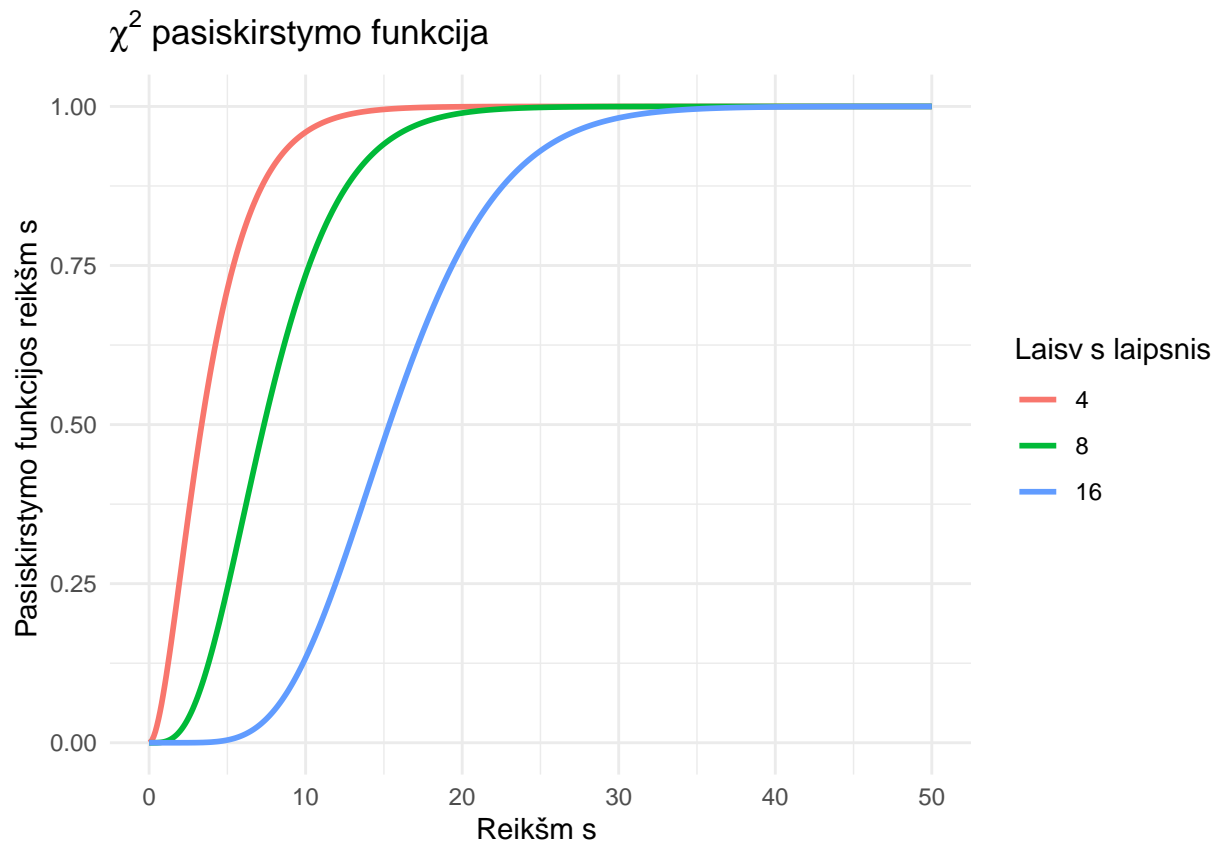
Brėžiame pasiskirstymo funkcijų grafiką.

```
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 = "Pasiskirstymo funkcijos reikšmės",
       x = "Reikšmės")+
  theme_minimal() +
  ggtitle(TeX("$\\chi^2$ pasiskirstymo funkcija"))
```



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

## 5 užduotis

### Kvantilių funkcija

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

čia:

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

### Grafikas

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

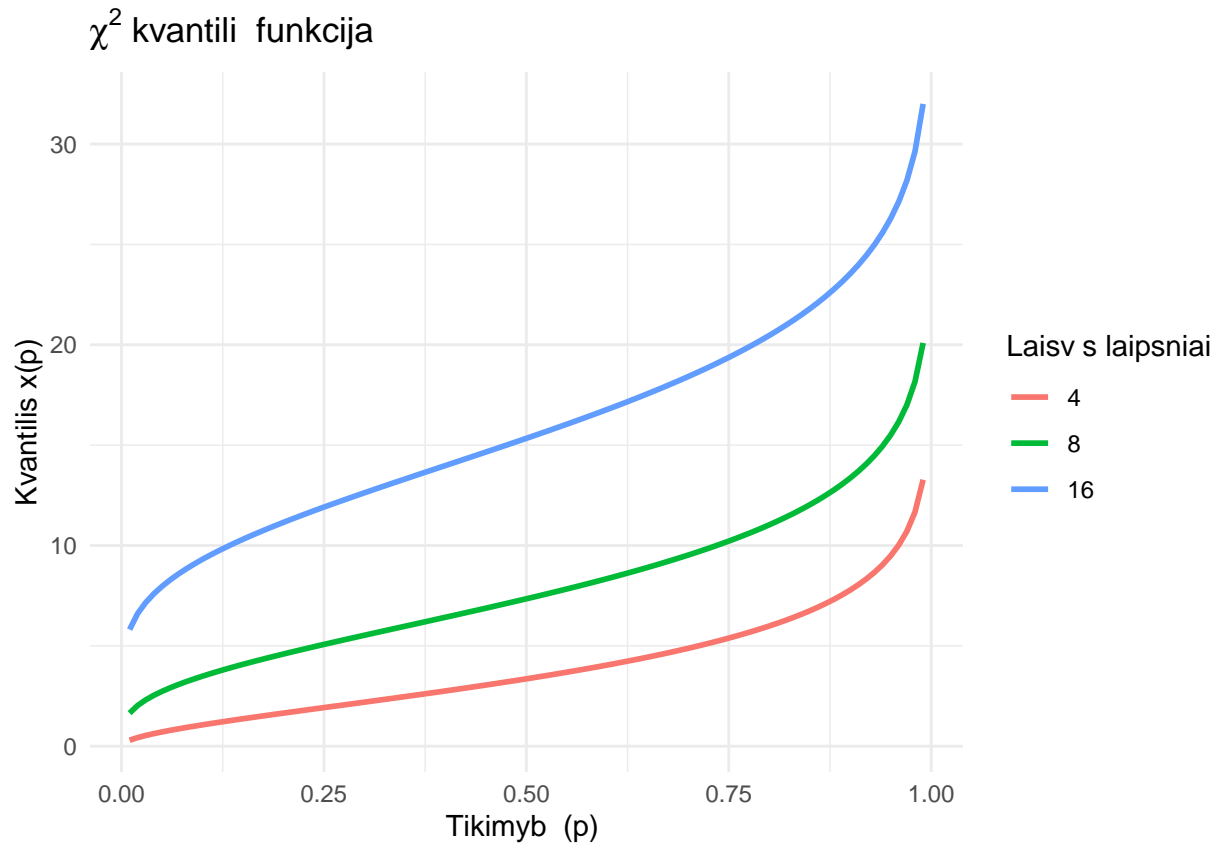
df_values <- c(4, 8, 16)

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 = TeX("$\\chi^2$ kvantilių funkcija "),
      x = "Tikimybė (p)",
      y = "Kvantilis x(p)" +
      theme_minimal()
```



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 užduotis

Fiksavome, pasirinktą a.d. parametrų rinkinį ( $k=5$ ). Sugeneravome  $\chi^2_5$  duomenų rinkinius su 20, 50, 200, 1000 imčių dydžiais.

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

set.seed(42)
imtis1 <- rchisq(n[1], df=k)

set.seed(42)
imtis2 <- rchisq(n[2], df=k)

set.seed(42)
imtis3 <- rchisq(n[3], df=k)

set.seed(42)
imtis4 <- rchisq(n[4], df=k)
```

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

## Nubrėžiame histogramas

```
# Apibrėžiame pagalbinę funkciją, kadangi grafikai labai panašūs
suppressWarnings({
  library(tidyr)
  library(dplyr)
})

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

plot_chisq_sample <- function(samples) {
  # samples - imčių vektorius

  # Imtis paverčiame duomenų lentelėmis ir jas sujungiame
  dt <- lapply(samples, function(x) {
    data.frame.out <- data.frame(x)
    data.frame.with.lengths <- data.frame.out %>%
      mutate("Imties.dydis" = length(x))
    return(data.frame.with.lengths)
  })

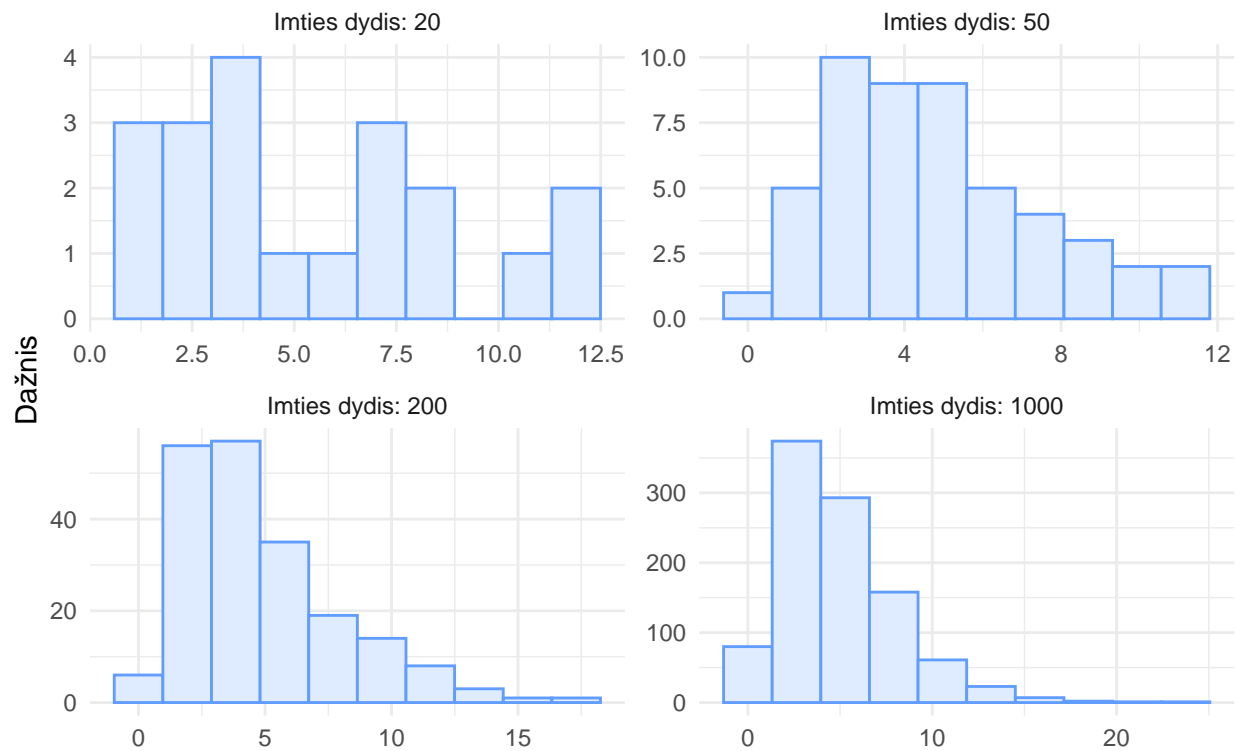
  df <- do.call(rbind, dt)

  # Dėl įskaitomumo pervardiname imties dydžio stulpelį
  labels.dydziu <- sapply(n, function(x) paste("Imties dydis:", x))
  df <- df %>%
    mutate("Imties.dydis" = factor(Imties.dydis, labels=labels.dydziu))

  # Nubrėžiame histogramas
  df %>%
    ggplot(aes(x=x)) +
      geom_histogram(bins = 10, color = "#619CFF", fill="#dfebff") +
      facet_wrap(~Imties.dydis, scales = "free") +
      theme_minimal() +
      xlab("") +
      ylab("Dažnis") +
      ggtitle("Histogramos visoms sugeneruotoms imtimis")
}

plot_chisq_sample(imtys)
```

## Histogramos visoms sugeneruotoms imtims



Empirinės pasiskirstymo funkcijos mūsų darbe imčių pasiskirstymo funkcijos pateikiamos su teorine pasiskirstymo funkcija.

```
nubrezti_chisq_empirini <- function(samples, k=5) {
  # samples - imtiys.
  # df - chi kvadratu parametras

  # samples - imčių vektorius

  # Imtis paverčiame duomenų lentelėmis ir jas sujungiame
  dt <- lapply(samples, function(x) {
    seq.x <- seq(min(x), max(x), length.out=1000)

    data.frame.out <- data.frame(seq.x,                # Įdedame seką
                                ecdf(x)(seq.x),        # Įdedame emp. p.f.
                                pchisq(seq.x, k))       # Įdedame p.f.

    data.frame.with.lengths <- data.frame.out %>%
      mutate("Imties.dydis" = length(x))

    return(data.frame.with.lengths)
  })

  df <- do.call(rbind, dt)

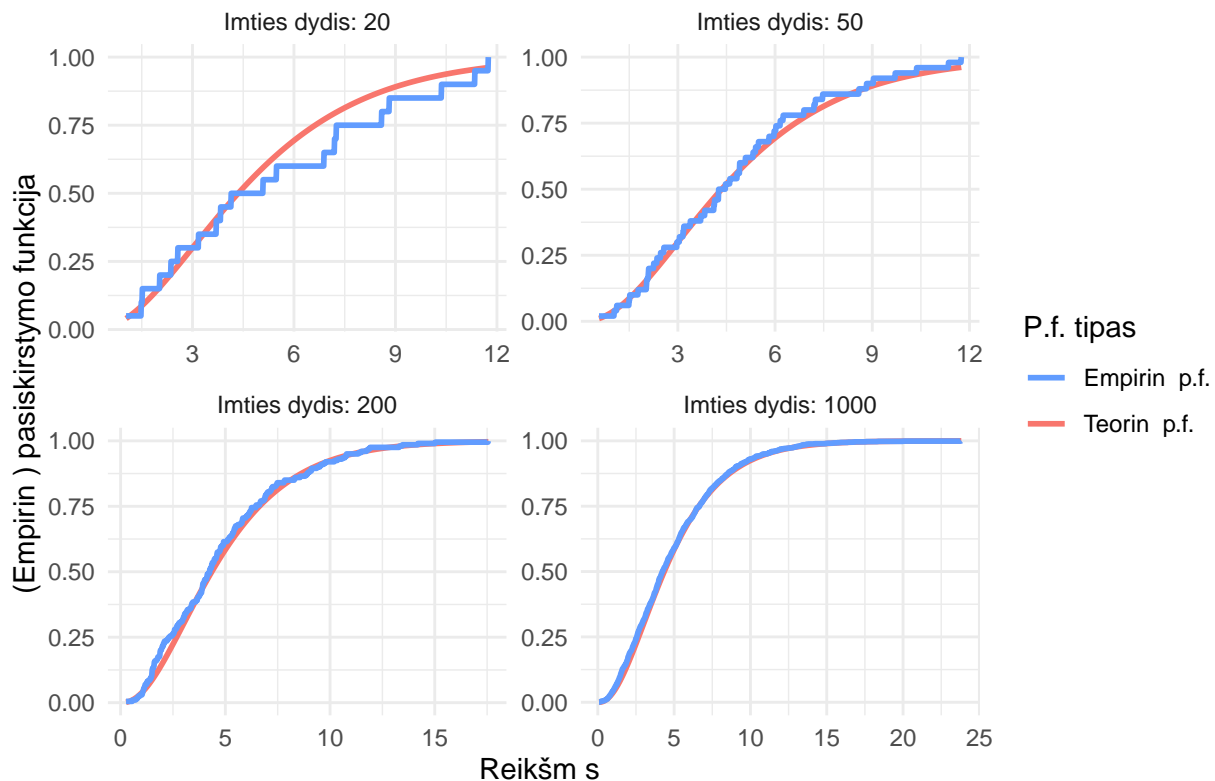
  # Dėl įskaitomumo pervardiname imties dydžio stulpelį
  labels.dydziu <- sapply(n, function(x) paste("Imties dydis:", x))
```

```
df <- df %>%
  mutate("Imties.dydis" = factor(Imties.dydis, labels=labels.dydziu))

# Nubrėžiame teorinę pasiskirstymo funkciją
df %>%
  ggplot(aes(x = seq.x)) +
    geom_line(aes(y = pchisq.seq.x..k., color="Teorinė p.f."),
              linewidth = 1) +
    geom_line(aes(y = ecdf.x..seq.x., color="Empirinė p.f."),
              linewidth = 1) +
    facet_wrap(~Imties.dydis, scales = "free") +
    theme_minimal() +
    labs(x = "Reikšmės",
         y = "(Empirinė) pasiskirstymo funkcija",
         title = "Empirinė ir teorinė pasiskirstymo funkcijos") +
    scale_color_manual("P.f. tipas",
                       breaks = c("Empirinė p.f.", "Teorinė p.f."),
                       values = c("#619CFF", "#F8766D"))
}
```

nubrezti\_chisq\_empirini(imtys)

## Empirin ir teorin pasiskirstymo funkcijos



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



## 7 užduotis

Remiantis 3. punktu prisimename, kad  $EX = k$ . Pirmasis žingsnis, sudarant įverčius 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 4.77809785556632"
## [3] "200 dydžio imties parametro įvertinys 4.83744392677588"
## [4] "1000 dydžio imties parametro įvertinys 4.90701265387103"
```

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

## 8 užduotis

```
# 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=imtys)
  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 4.92064761863795"
## [3] "200 dydžio imties parametro įvertinys 4.80092615216356"
## [4] "1000 dydžio imties parametro įvertinys 4.86620238891593"
```

## 9 užduotis

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 užduotis

Parametrų patikimumo intervalai

Pasikliautiniams intervalams nustatyti naudojome procentinį bootstrap metodą.

```
library(boot)

mean_mod <- function(data, idx) {
  # Modifikuojame vidurkio funkciją, kad galėtume gauti bootstrap imtis
  return(mean(data[idx]))
}

bootstrap_ci <- function(sample) {
  bootstrap <- boot(sample, mean_mod, R = 100)
  boot_ci_output <- boot.ci(boot.out=bootstrap, type="perc")

  print(paste("Imties ", length(sample), " pasikliautiniai intervalai"))
  print(boot_ci_output$perc[c(4,5)] )
}

bootstrap_ci(imtis1)

## [1] "Imties 20 pasikliautiniai intervalai"
## [1] 4.007542 7.014174

bootstrap_ci(imtis2)

## [1] "Imties 50 pasikliautiniai intervalai"
## [1] 3.952414 5.544095

bootstrap_ci(imtis3)

## [1] "Imties 200 pasikliautiniai intervalai"
## [1] 4.370672 5.336238

bootstrap_ci(imtis4)

## [1] "Imties 1000 pasikliautiniai intervalai"
## [1] 4.708274 5.143900
```

Pastebime, kad patikimumo intervalas vis arčiau prilimpa tikrojo parametro - 5. Matome, kad 50 objektų imtyje patikimumo intervalas pasislinko į kairę (palyginus su 20). Nors didesnės imties patikimumo intervalo apatinis rėžis yra toliau, turime turėti omenyje, kad vidurkis tiesiog gavosi mažesnis.

## 11 užduotis

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%
```

```
## 2.487877 4.362066 6.120290
##
## [[3]]
##      25%      50%      75%
## 2.340236 4.250821 6.469480
##
## [[4]]
##      25%      50%      75%
## 2.550964 4.246462 6.650125
```

## 12 užduotis

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%
## -8.40936 17.01753
##
## [[3]]
##      25%      75%
## -10.04749 18.85721
##
## [[4]]
##      25%      75%
## -9.746521 18.947610
```

Kad galėtume atpažinti vėliau, patikrinkime, duomenų amplitudes

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

```
## [1] "20 imties dyio amplitudė: 1.03617574402698 - 11.7459815259314"
## [2] "50 imties dyio amplitudė: 0.567393736980694 - 11.7459815259314"
## [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 kvartilių 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)

su.iskirtimis <- list(imtis1_isskirtys,
                     imtis2_isskirtys,
                     imtis3_isskirtys,
                     imtis4_isskirtys)
```

### 13 užduotis

Nubraižėme po stačiakampę diagramą kiekvienam 12 dalies duomenų rinkiniui

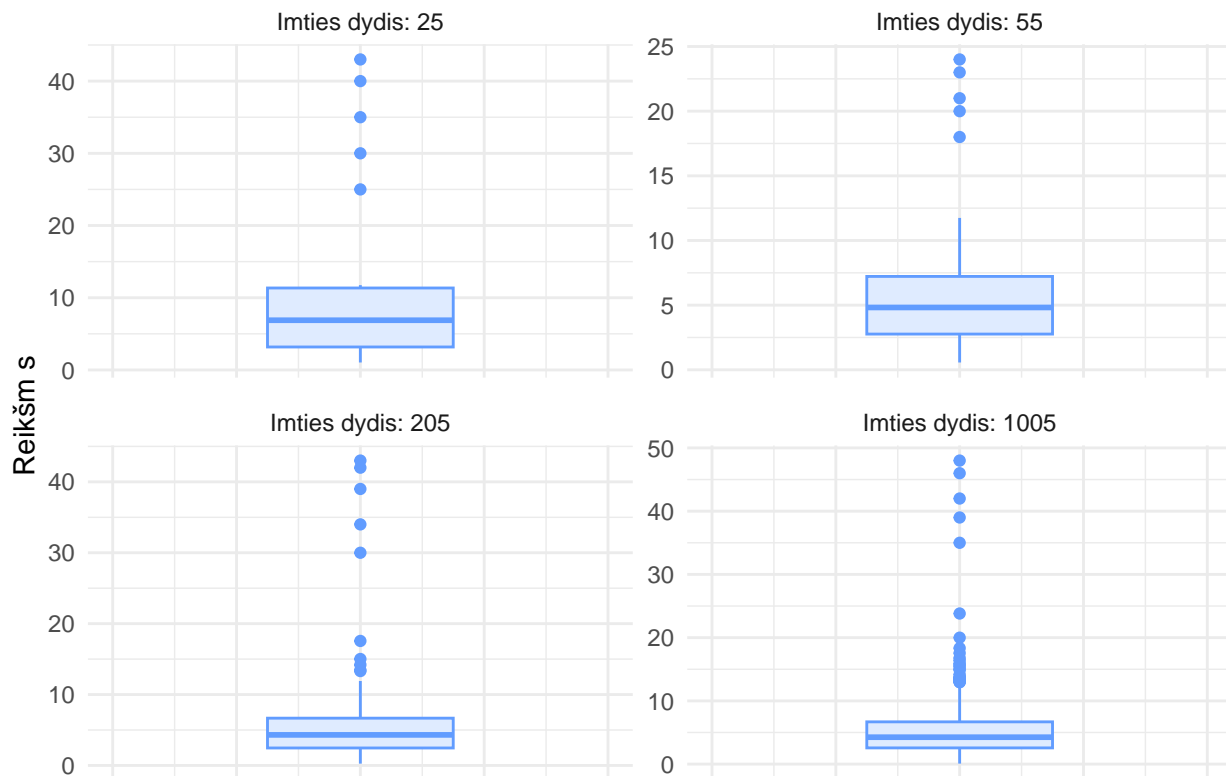
```
dt <- lapply(su.iskirtimis, function(x) {
  data.frame.out <- data.frame(x)
  data.frame.with.lengths <- data.frame.out %>%
    mutate("Imties.dydis" = length(x))
  return(data.frame.with.lengths)
})

df <- do.call(rbind, dt)

labels.dydziu <- sapply(n, function(x) paste("Imties dydis:", x+5))
df <- df %>%
  mutate("Imties.dydis" = factor(Imties.dydis, labels=labels.dydziu))

df %>%
  ggplot(aes(y = x)) +
  geom_boxplot(color = "#619CFF", fill = "#dfebff") +
  xlim(c(-1, 1)) +
  facet_wrap(~ Imties.dydis, scales = "free") +
  theme_minimal() +
  theme(
    axis.text.x = element_blank()) +
  labs(y = "Reikšmės",
       title = "Stačiakampės diagramos")
```

## Stačiakampės diagramos



Matome, kad stačiakampės diagramos identifikuoja įrašytas išskirtis.

## 14 užduotis

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))

append.normal <- function(sample) {
  seq.x <- seq(min(sample), max(sample),
               length.out=length(sample))

  norm.equiv <- dnorm(seq.x,
                      mean=mean(sample),
                      sd=sd(sample))

  return(
    data.frame(list(imtis = sample,
                    Imtis.dydis = rep(length(sample), length(sample)),
                    seq.x = seq.x,
                    norm.equiv = norm.equiv))
  )
}
```

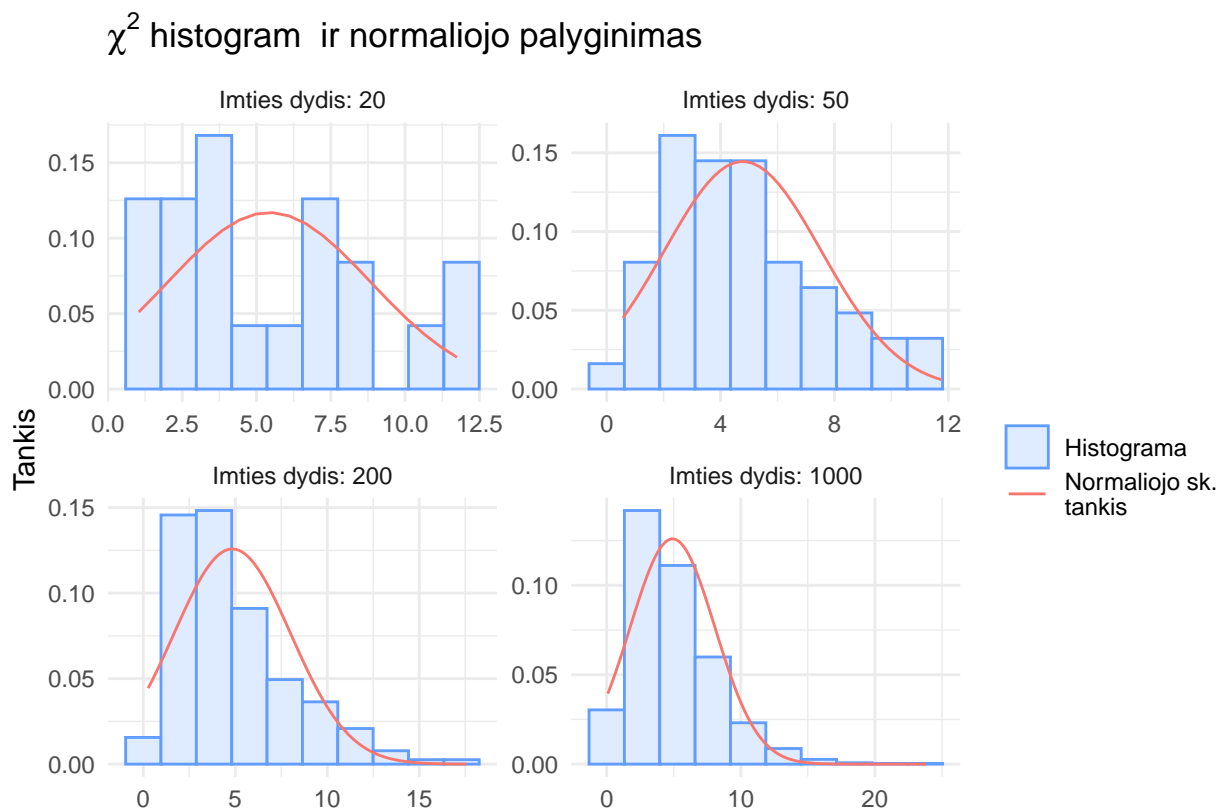
```

dt <- lapply(imtys, append.normal)
df <- do.call(rbind, dt)

labels.dydziu <- sapply(n, function(x) paste("Imties dydis:", x))
df <- df %>%
  mutate("Imtis.dydis" = factor(Imtis.dydis, labels=labels.dydziu))

df %>%
  ggplot(aes(x = imtis)) +
    geom_histogram(aes(y = after_stat(density), color = "Histograma"),
                  bins=10, fill="#dfebf5") +
    geom_line(aes(x = seq.x, y = norm.equiv,
                  color = "Normaliojo sk.\ntankis")) +
    facet_wrap(~ Imtis.dydis, scales="free") +
    scale_color_manual("",
                      breaks = c("Histograma", "Normaliojo sk.\ntankis"),
                      values = c("#619CFF", "#F8766D")) +
    theme_minimal() +
    labs(x="", y="Tankis",
         title=TeX("$\\chi^2$ histogramų ir normaliojo palyginimas"))

```



## 14 užduotis

Paprastai,  $\chi^2$  skistinis yra naudojamas  $\chi^2$  testui, tačiau taip pat normaliojo skirstinio imties dispersiją galime modeliuoti pagal  $\chi^2$  skirstinį:

$$\frac{(n-1)S^2}{\sigma^2} \sim \chi_{n-1}^2 \quad (10)$$

Pavyzdžiui paime žinomą duomenų rinkinį: vynu kokybės.

```
library(boot)
library(latex2exp)

wine.quality <- read.csv('WineQT.csv')

var_pop <- var(wine.quality$pH)
n <- 10
var_sample <- sapply(1:1000, function(x) var(sample(wine.quality$pH, n)))

chi_sq_example <- data.frame(var_sample / var_pop * (n-1))
colnames(chi_sq_example) <- c("Sample variance")

chi_sq_example %>%
  ggplot(aes(x = `Sample variance`)) +
  geom_histogram(aes(y=after_stat(density)),
    color='#619CFF', fill='#dfebf') +
  stat_function(fun = dchisq, args = c(n-1), color="#F8766D") +
  theme_minimal() +
  labs(y = "", x = "",
    title = TeX("$\\frac{S^2}{\\sigma^2} (n-1) \\sim \\chi^2_{n-1}$ pasiskirstymas"))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

$$\frac{S^2(n-1)}{\sigma^2} \text{ pasiskirstymas}$$

