Bivariate\_hypothesis\_testing

Oleg Arnaut

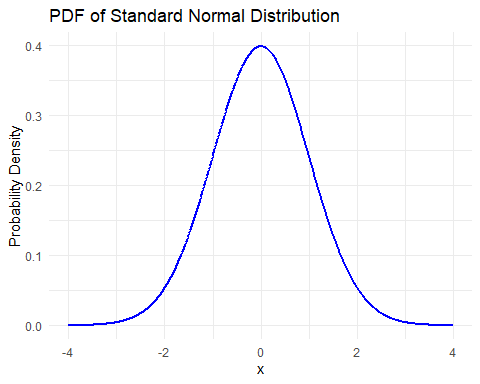
2023-11-02

#Распределения

## Normal Distribution

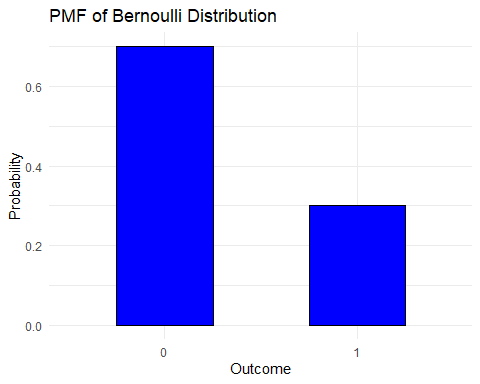
# Load necessary libraries  
library(ggplot2)  
  
# Create a sequence of x values  
x\_values <- seq(-4, 4, by = 0.01)  
  
# Calculate the PDF values for the standard normal distribution  
pdf\_normal <- dnorm(x\_values, mean = 0, sd = 1)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_normal)  
  
# Create a PDF plot to visualize the standard normal distribution  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "blue") +  
 labs(title = "PDF of Standard Normal Distribution", x = "x", y = "Probability Density") +  
 theme\_minimal()

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



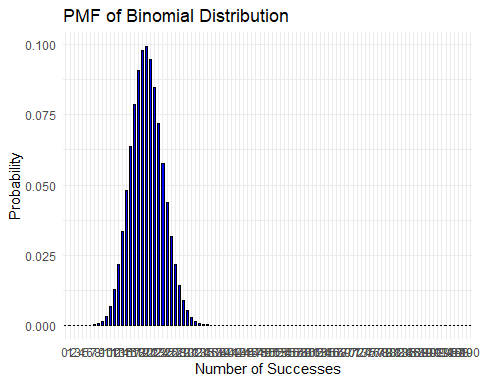
## Bernoulli Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the probability of success for the Bernoulli distribution  
p <- 0.3  
  
# Create a sequence of possible values (0 and 1)  
x\_values <- c(0, 1)  
  
# Calculate the PMF values for the Bernoulli distribution  
pmf\_bernoulli <- c(1 - p, p)  
  
# Create a data frame for plotting  
pmf\_data <- data.frame(x = x\_values, Probability = pmf\_bernoulli)  
  
# Create a PMF plot to visualize the Bernoulli distribution  
ggplot(pmf\_data, aes(x = factor(x), y = Probability)) +  
 geom\_bar(stat = "identity", fill = "blue", color = "black", width = 0.5) +  
 labs(title = "PMF of Bernoulli Distribution", x = "Outcome", y = "Probability") +  
 theme\_minimal()



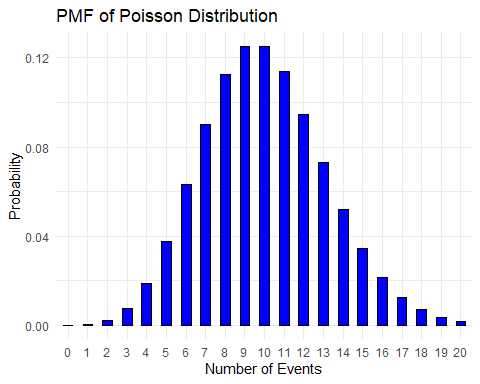
## Binomial Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the parameters of the binomial distribution  
n <- 100 # Number of trials  
p <- 0.2 # Probability of success in each trial  
  
# Create a sequence of possible values for the number of successes  
x\_values <- 0:n  
  
# Calculate the PMF values for the binomial distribution  
pmf\_binomial <- dbinom(x\_values, size = n, prob = p)  
  
# Create a data frame for plotting  
pmf\_data <- data.frame(x = x\_values, Probability = pmf\_binomial)  
  
# Create a PMF plot to visualize the binomial distribution  
ggplot(pmf\_data, aes(x = factor(x), y = Probability)) +  
 geom\_bar(stat = "identity", fill = "blue", color = "black", width = 0.5) +  
 labs(title = "PMF of Binomial Distribution", x = "Number of Successes", y = "Probability") +  
 theme\_minimal()



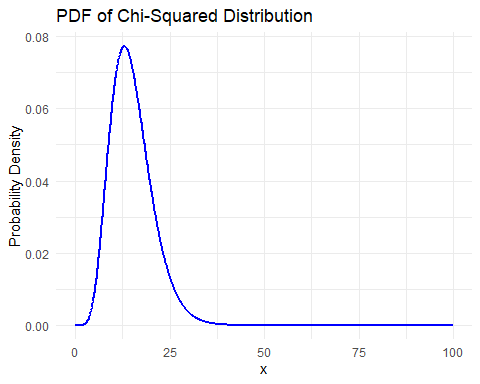
## Poisson Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the average rate of events for the Poisson distribution  
lambda <- 10 # Average rate of events  
  
# Create a sequence of possible values for the number of events  
x\_values <- 0:20  
  
# Calculate the PMF values for the Poisson distribution  
pmf\_poisson <- dpois(x\_values, lambda)  
  
# Create a data frame for plotting  
pmf\_data <- data.frame(x = x\_values, Probability = pmf\_poisson)  
  
# Create a PMF plot to visualize the Poisson distribution  
ggplot(pmf\_data, aes(x = factor(x), y = Probability)) +  
 geom\_bar(stat = "identity", fill = "blue", color = "black", width = 0.5) +  
 labs(title = "PMF of Poisson Distribution", x = "Number of Events", y = "Probability") +  
 theme\_minimal()



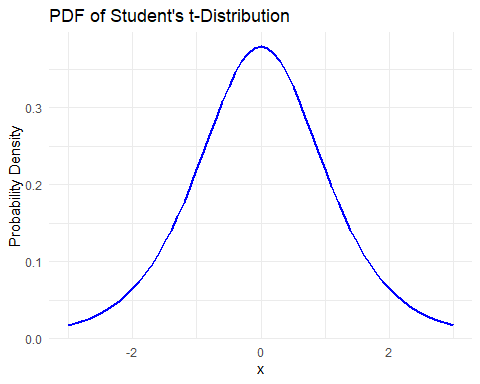
## Chi-Squared Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the degrees of freedom for the chi-squared distribution  
df <- 15 # Degrees of freedom  
  
# Create a sequence of x values  
x\_values <- seq(0, 100, by = 0.1) # Adjust the range and step size as needed  
  
# Calculate the PDF values for the chi-squared distribution  
pdf\_chi\_squared <- dchisq(x\_values, df)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_chi\_squared)  
  
# Create a PDF plot to visualize the chi-squared distribution  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "blue") +  
 labs(title = "PDF of Chi-Squared Distribution", x = "x", y = "Probability Density") +  
 theme\_minimal()



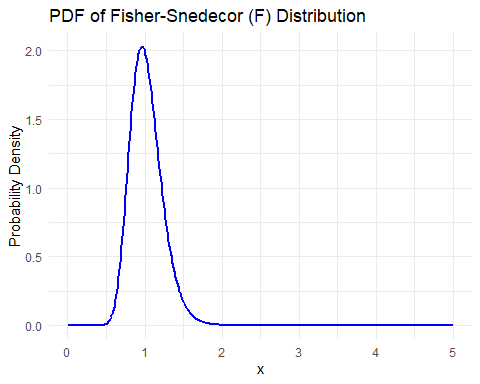
## Student’s Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the degrees of freedom for the t-distribution  
df <- 5 # Degrees of freedom  
  
# Create a sequence of x values  
x\_values <- seq(-3, 3, by = 0.1) # Adjust the range and step size as needed  
  
# Calculate the PDF values for the t-distribution  
pdf\_t <- dt(x\_values, df)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_t)  
  
# Create a PDF plot to visualize the t-distribution  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "blue") +  
 labs(title = "PDF of Student's t-Distribution", x = "x", y = "Probability Density") +  
 theme\_minimal()



## Fisher-Snedecor Distribution

# Load necessary libraries  
library(ggplot2)  
  
# Define the degrees of freedom parameters for the F-distribution  
df1 <- 100 # Degrees of freedom numerator  
df2 <- 100 # Degrees of freedom denominator  
  
# Create a sequence of x values  
x\_values <- seq(0.01, 5, by = 0.01) # Adjust the range and step size as needed  
  
# Calculate the PDF values for the F-distribution  
pdf\_fisher <- df(x\_values, df1, df2)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_fisher)  
  
# Create a PDF plot to visualize the F-distribution  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "blue") +  
 labs(title = "PDF of Fisher-Snedecor (F) Distribution", x = "x", y = "Probability Density") +  
 theme\_minimal()

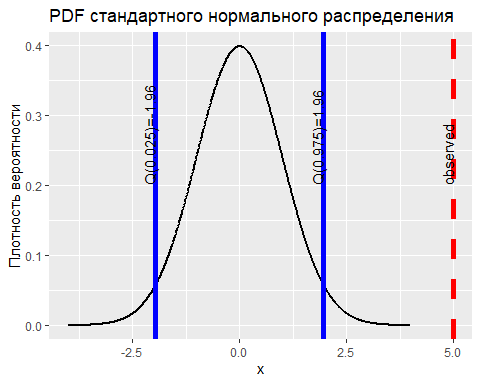


# Параметрические критерии

## Гипотезы о средних

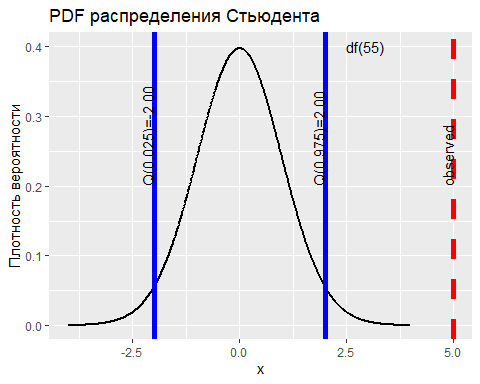
### Z-критерий для среднего

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=5  
  
# Создание последовательности значений x  
x\_values <- seq(-4, 4, by = 0.01)  
  
# Расчет значений Плотности Вероятности (PDF) для стандартного нормального распределения  
pdf\_normal <- dnorm(x\_values, mean = 0, sd = 1)  
  
# Квантили  
quantiles <- c(0.025, 0.975)  
  
quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
# Создание данных для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_normal)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = quantile\_values, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = quantile\_values[1]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], quantile\_values[1]), hjust = 0, angle = 90) +  
 annotate("text", x = quantile\_values[2]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], quantile\_values[2]), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 labs(title = "PDF стандартного нормального распределения", x = "x", y = "Плотность вероятности")



### t-критерий для среднего

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=5  
  
# Define the degrees of freedom for the t-distribution  
df <- 55 # Degrees of freedom  
  
# Create a sequence of x values  
x\_values <- seq(-4, 4, by = 0.01) # Adjust the range and step size as needed  
  
# Calculate the PDF values for the t-distribution  
pdf\_t <- dt(x\_values, df)  
  
# Квантили  
quantiles <- c(0.025, 0.975)  
  
#quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
t\_critical\_low <- qt(quantiles[1], df = df)  
t\_critical\_up <- qt(quantiles[2], df = df)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_t)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = t\_critical\_low, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = t\_critical\_up, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = t\_critical\_low-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], t\_critical\_low), hjust = 0, angle = 90) +  
 annotate("text", x = t\_critical\_up-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], t\_critical\_up), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 annotate("text", x = 2.5, y = 0.4, label = sprintf("df(%.0f)", df), hjust = 0)+  
 labs(title = "PDF распределения Стьюдента", x = "x", y = "Плотность вероятности")



### Реализация в R

library(BSDA)  
  
# Тест Z для одной выборки  
  
# Данные (замените их своими данными)  
sample\_data\_z <- c(23, 25, 27, 21, 24, 26, 22, 28, 25, 27)  
  
# Гипотеза о среднем значении в генеральной совокупности  
population\_mean\_z <- 25  
  
# Выполнить тест Z  
z\_test\_result <- BSDA::z.test(sample\_data\_z, mu = population\_mean\_z, sigma.x =sd(sample\_data\_z))  
  
# Вывести результат  
print("Тест Z для одной выборки:")

## [1] "Тест Z для одной выборки:"

print(z\_test\_result)

##   
## One-sample z-Test  
##   
## data: sample\_data\_z  
## z = -0.27501, p-value = 0.7833  
## alternative hypothesis: true mean is not equal to 25  
## 95 percent confidence interval:  
## 23.37462 26.22538  
## sample estimates:  
## mean of x   
## 24.8

# Тест t для одной выборки  
  
# Данные (замените их своими данными)  
sample\_data\_t <- c(23, 25, 27, 21, 24, 26, 22, 28, 25, 27)  
  
# Гипотеза о среднем значении в генеральной совокупности  
population\_mean\_t <- 25  
  
# Выполнить тест t  
t\_test\_result <- t.test(sample\_data\_t, mu = population\_mean\_t)  
  
# Вывести результат  
print("\nТест t для одной выборки:")

## [1] "\nТест t для одной выборки:"

print(t\_test\_result)

##   
## One Sample t-test  
##   
## data: sample\_data\_t  
## t = -0.27501, df = 9, p-value = 0.7895  
## alternative hypothesis: true mean is not equal to 25  
## 95 percent confidence interval:  
## 23.15485 26.44515  
## sample estimates:  
## mean of x   
## 24.8

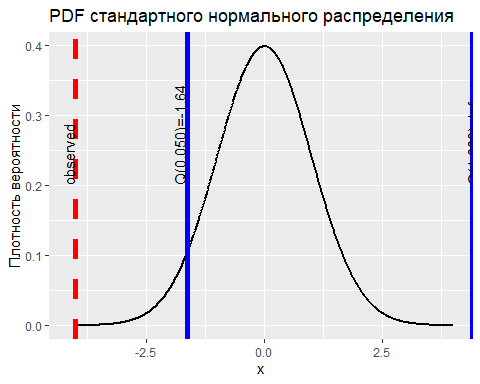
?t.test

## starting httpd help server ... done

## Гипотезы о разнице средних

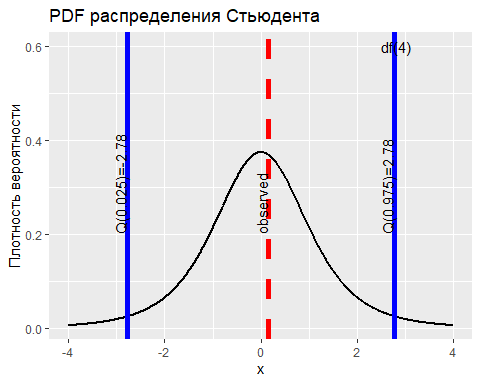
### Z-критерий для разности средних. Выборки независимые

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=-4  
  
# Создание последовательности значений x  
x\_values <- seq(-4, 4, by = 0.01)  
  
# Расчет значений Плотности Вероятности (PDF) для стандартного нормального распределения  
pdf\_normal <- dnorm(x\_values, mean = 0, sd = 1)  
  
# Квантили  
quantiles <- c(0.05, 1)  
  
quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
# Создание данных для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_normal)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = quantile\_values, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = quantile\_values[1]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], quantile\_values[1]), hjust = 0, angle = 90) +  
 annotate("text", x = quantile\_values[2]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], quantile\_values[2]), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 labs(title = "PDF стандартного нормального распределения", x = "x", y = "Плотность вероятности")



### t-критерий для разности средних. Выборки независимые

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=0.15  
  
# Define the degrees of freedom for the t-distribution  
df <- 4 # Degrees of freedom  
  
# Create a sequence of x values  
x\_values <- seq(-4, 4, by = 0.01) # Adjust the range and step size as needed  
  
# Calculate the PDF values for the t-distribution  
pdf\_t <- dt(x\_values, df)  
  
# Квантили  
quantiles <- c(0.025, 0.975)  
  
#quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
t\_critical\_low <- qt(quantiles[1], df = df)  
t\_critical\_up <- qt(quantiles[2], df = df)  
  
# Create a data frame for plotting  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_t)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = t\_critical\_low, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = t\_critical\_up, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = t\_critical\_low-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], t\_critical\_low), hjust = 0, angle = 90) +  
 annotate("text", x = t\_critical\_up-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], t\_critical\_up), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 annotate("text", x = 2.5, y = 0.6, label = sprintf("df(%.0f)", df), hjust = 0)+  
 labs(title = "PDF распределения Стьюдента", x = "x", y = "Плотность вероятности")



?z.test  
?var()

## Степени свободы для теста Welch  
  
sd\_x = 6  
nx = 10  
sd\_y = 4  
ny = 10  
  
# Степени свободы (Welch-Satterthwaite)  
numerator <- ((sd\_x^2 / nx) + (sd\_y^2 / ny))^2  
denominator <- (sd\_x^4 / (nx^2 \* (nx - 1))) + (sd\_y^4 / (ny^2 \* (ny - 1)))  
  
df <- numerator / denominator  
  
df

## [1] 15.68041

### Реализация в R

library(BSDA)  
  
  
# Sample data for group A and group B  
group\_A <- c(15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25)  
group\_A <- group\_A + 2  
group\_B <- c(10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30)  
  
  
z\_test <- BSDA::z.test(x = group\_A, y = group\_B, alternative = "two.sided", sigma.x =sd(group\_A), sigma.y = sd(group\_B))  
  
# Perform a t-test for the means of two independent samples (assuming equal variances)  
t\_test <- t.test(group\_B, group\_A, var.equal = TRUE)  
  
  
# Perform a t-test for the means of two independent samples (assuming unequal variances)  
t\_test\_Welch <- t.test(group\_B, group\_A, var.equal = FALSE)  
  
  
print(t\_test\_Welch)

##   
## Welch Two Sample t-test  
##   
## data: group\_B and group\_A  
## t = -0.89443, df = 14.706, p-value = 0.3855  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -6.774382 2.774382  
## sample estimates:  
## mean of x mean of y   
## 20 22

print(t\_test)

##   
## Two Sample t-test  
##   
## data: group\_B and group\_A  
## t = -0.89443, df = 20, p-value = 0.3817  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -6.664356 2.664356  
## sample estimates:  
## mean of x mean of y   
## 20 22

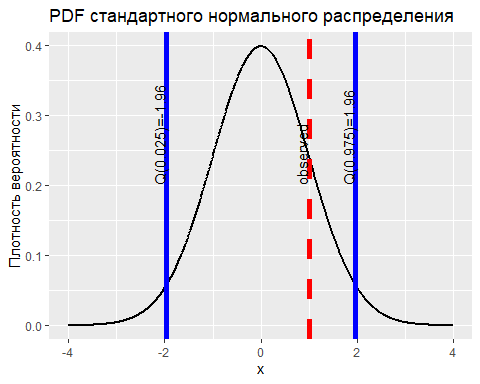
print(z\_test)

##   
## Two-sample z-Test  
##   
## data: group\_A and group\_B  
## z = 0.89443, p-value = 0.3711  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.382613 6.382613  
## sample estimates:  
## mean of x mean of y   
## 22 20

## Гипотезы о долях

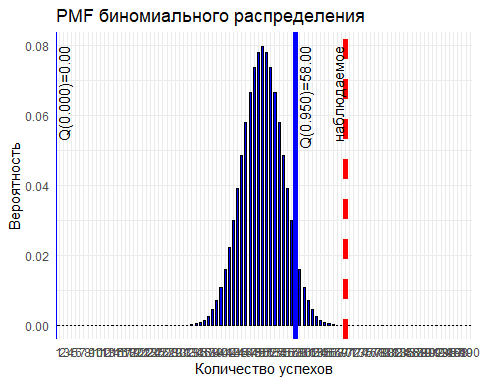
### Z-критерий для доли

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=1  
  
# Создание последовательности значений x  
x\_values <- seq(-4, 4, by = 0.01)  
  
# Расчет значений Плотности Вероятности (PDF) для стандартного нормального распределения  
pdf\_normal <- dnorm(x\_values, mean = 0, sd = 1)  
  
# Квантили  
quantiles <- c(0.025, 0.975)  
  
quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
# Создание данных для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_normal)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = quantile\_values, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = quantile\_values[1]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], quantile\_values[1]), hjust = 0, angle = 90) +  
 annotate("text", x = quantile\_values[2]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], quantile\_values[2]), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 labs(title = "PDF стандартного нормального распределения", x = "x", y = "Плотность вероятности")



### Точный критерий для доли

# Загрузка необходимых библиотек  
library(ggplot2)  
  
# Определение параметров биномиального распределения  
n <- 100 # Количество испытаний  
p <- 0.5 # Вероятность успеха в каждом испытании (истина)  
  
наблюдаемые = 70  
  
# Создание последовательности возможных значений числа успехов  
x\_values <- 1:n  
  
# Расчет значений ПФР (вероятности массовой функции) для биномиального распределения  
pmf\_binomial <- dbinom(x\_values, size = n, prob = p)  
  
# Создание фрейма данных для построения  
pmf\_data <- data.frame(x = x\_values, Вероятность = pmf\_binomial)  
  
# Квантили  
квантили <- c(0, 0.95)  
  
# Расчет критических значений (квантилей)  
critical\_value\_left <- qbinom(квантили[1], size = n, prob = p)  
critical\_value\_right <- qbinom(квантили[2], size = n, prob = p)  
  
# Создание графика ПФР для визуализации биномиального распределения  
ggplot(pmf\_data, aes(x = factor(x), y = Вероятность)) +  
 geom\_bar(stat = "identity", fill = "blue", color = "black", width = 0.5) +  
 labs(title = "PMF биномиального распределения", x = "Количество успехов", y = "Вероятность") +  
 theme\_minimal() +  
 geom\_vline(xintercept = critical\_value\_left, linetype = "solid", color = "blue", size = 2) +  
 geom\_vline(xintercept = critical\_value\_right, linetype = "solid", color = "blue", size = 2) +  
 geom\_vline(xintercept = наблюдаемые, size = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = critical\_value\_left + n/50, y = max(pmf\_binomial), label = sprintf("Q(%.3f)=%.2f", квантили[1], critical\_value\_left), color = "black", hjust = 1, angle = 90) +  
 annotate("text", x = critical\_value\_right + n/50, y = max(pmf\_binomial), label = sprintf("Q(%.3f)=%.2f", квантили[2], critical\_value\_right), color = "black", hjust = 1, angle = 90) +  
 annotate("text", x = наблюдаемые - n/50, y = max(pmf\_binomial), label = 'наблюдаемое', hjust = 1, angle = 90)



### Реализация в R

library(BSDA)  
  
# Aсимптотический тест  
# Sample data  
n <- 100 # Total number of trials  
x <- 70 # Number of successes  
  
# Null hypothesis value (usually 0.5 for a two-sided test)  
p0 <- 0.5  
  
# Create the binary sample  
sample <- c(rep(1, x), rep(0, n - x))  
  
# Shuffle the sample to randomize the order  
  
sample <- sample(sample)  
  
stdev <- sqrt(p0\*(1-p0))  
  
# Perform the z-test  
z\_test\_result <- BSDA::z.test(sample, mu = p0, alternative = "two.sided", sigma.x=stdev)  
  
# Print the test result  
print(z\_test\_result)

##   
## One-sample z-Test  
##   
## data: sample  
## z = 4, p-value = 6.334e-05  
## alternative hypothesis: true mean is not equal to 0.5  
## 95 percent confidence interval:  
## 0.6020018 0.7979982  
## sample estimates:  
## mean of x   
## 0.7

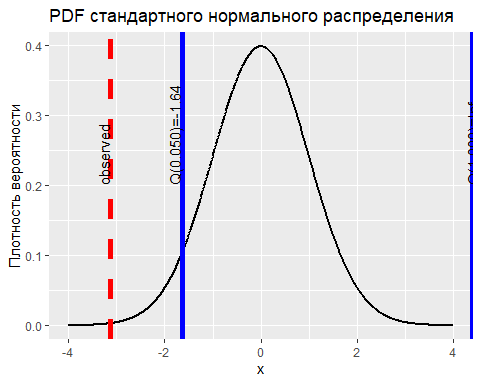
# Точный тест  
?binom.test()  
  
# Number of trials (sample size)  
n <- 100  
  
# Expected proportion  
expected\_proportion <- 0.5  
  
# Number of successes (observed)  
observed\_successes <- 70  
  
# Perform the binomial test  
binom\_test\_result <- binom.test(observed\_successes, n, p = expected\_proportion, alternative = "two.sided")  
  
# Print the test result  
print(binom\_test\_result)

##   
## Exact binomial test  
##   
## data: observed\_successes and n  
## number of successes = 70, number of trials = 100, p-value = 7.85e-05  
## alternative hypothesis: true probability of success is not equal to 0.5  
## 95 percent confidence interval:  
## 0.6001853 0.7875936  
## sample estimates:  
## probability of success   
## 0.7

## Гипотезы о разнице долей

### Z-критерий для разности долей для независимых выборок и зависимых выборок

# Загрузка необходимого пакета ggplot2  
library(ggplot2)  
  
#?quantile  
  
observed=-3.125  
  
# Создание последовательности значений x  
x\_values <- seq(-4, 4, by = 0.01)  
  
# Расчет значений Плотности Вероятности (PDF) для стандартного нормального распределения  
pdf\_normal <- dnorm(x\_values, mean = 0, sd = 1)  
  
# Квантили  
quantiles <- c(0.05, 1)  
  
quantile\_values <- qnorm(quantiles, mean = 0, sd = 1)  
  
# Создание данных для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_normal)  
  
# Создание графика PDF для визуализации стандартного нормального распределения  
ggplot(pdf\_data, aes(x = x)) +  
 geom\_line(aes(y = Probability), linewidth = 1, color = "black", linetype = "solid") +  
 geom\_vline(xintercept = quantile\_values, linewidth = 2, linetype = "solid", color = "blue") +  
 geom\_vline(xintercept = observed, linewidth = 2, linetype = "dashed", color = "red") +  
 annotate("text", x = quantile\_values[1]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], quantile\_values[1]), hjust = 0, angle = 90) +  
 annotate("text", x = quantile\_values[2]-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], quantile\_values[2]), hjust = 0, angle = 90) +  
 annotate("text", x = observed-0.15, y = 0.2, label = 'observed', hjust = 0, angle = 90)+  
 labs(title = "PDF стандартного нормального распределения", x = "x", y = "Плотность вероятности")



### Реализация в R

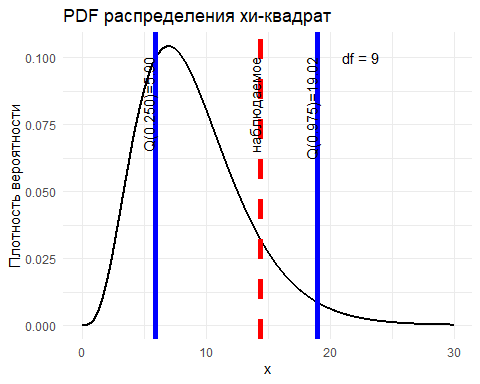
# Aсимптотический тест для независимых выборок  
  
# # Исходные данные для выборки 1  
# n1 <- 100 # Общее количество попыток  
# x <- 20 # Количество успешных исходов  
#   
# # Исходные данные для выборки 2  
# n2 <- 100 # Общее количество попыток  
# y <- 40 # Количество успешных исходов  
#   
# # Расчет доли успешных исходов в выборке 1  
# p1 <- x / n1  
#   
# # Расчет доли успешных исходов в выборке 2  
# p2 <- y / n2  
#   
# # Расчет объединенной доли успешных исходов  
# P <- (x + y) / (n1 + n2)  
#   
# # Расчет стандартной ошибки  
# SQ <- P \* (1 - P) \* (1/n1 + 1/n2)  
#   
# # Выполнение z-теста  
# z\_test\_result <- (p1 - p2) / sqrt(SQ)  
#   
# # Вывод результата теста  
# print(z\_test\_result)  
  
##################################################################  
  
  
# Функция для проведения z-теста для двух долей  
performTwoProportionZTest <- function(n1, x, n2, y) {  
 # Расчет долей успешных исходов в выборках  
 p1 <- x / n1  
 p2 <- y / n2  
  
 # Расчет объединенной доли успешных исходов  
 P <- (x + y) / (n1 + n2)  
  
 # Расчет стандартной ошибки  
 SQ <- P \* (1 - P) \* (1/n1 + 1/n2)  
  
 # Выполнение z-теста  
 z\_test\_result <- (p1 - p2) / sqrt(SQ)  
  
 return(z\_test\_result)  
}  
  
# Использование функции:  
n1 <- 100 # Общее количество попыток для выборки 1  
x <- 20 # Количество успешных исходов для выборки 1  
  
n2 <- 100 # Общее количество попыток для выборки 2  
y <- 40 # Количество успешных исходов для выборки 2  
  
z\_test\_result <- performTwoProportionZTest(n1, x, n2, y)  
print(z\_test\_result)

## [1] -3.086067

# Гипотезы о дисперсии

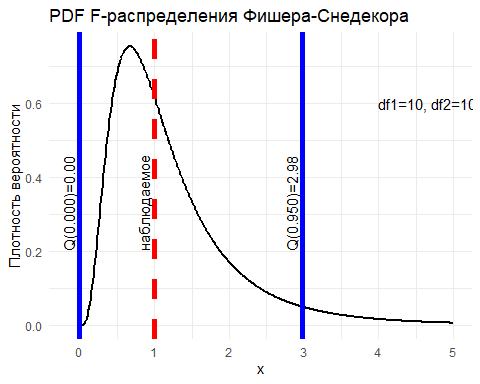
### Chi squared – критерий для дисперсии (точный)

# Загрузка необходимых библиотек  
library(ggplot2)  
  
# Определение степеней свободы для распределения хи-квадрат  
df <- 9 # Подстраивайте степени свободы по мере необходимости  
  
# Создание последовательности значений x  
x\_values <- seq(0.01, 30, by = 0.01) # Подстраивайте диапазон и шаг по мере необходимости  
  
# Расчет значений PDF для распределения хи-квадрат  
pdf\_chi\_squared <- dchisq(x\_values, df)  
  
# Квантили  
quantiles <- c(0.25, 0.975)  
  
# Расчет критических значений для распределения хи-квадрат  
critical\_value\_lower <- qchisq(quantiles[1], df)  
critical\_value\_upper <- qchisq(quantiles[2], df)  
  
# Расчет наблюдаемого значения хи-квадрат (замените на ваше фактическое наблюдаемое значение)  
observed\_chi\_squared <- 14.4 # Замените на ваше реальное наблюдаемое значение  
  
# Создание data frame для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_chi\_squared)  
  
# Создание графика PDF для визуализации распределения хи-квадрат  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "black") +  
 labs(title = "PDF распределения хи-квадрат", x = "x", y = "Плотность вероятности") +  
 theme\_minimal() +  
 geom\_vline(xintercept = c(critical\_value\_lower, critical\_value\_upper, observed\_chi\_squared),  
 linetype = c("solid", "solid", "dashed"),  
 color = c("blue", "blue", "red"),  
 size = c(2, 2, 2)) +  
 annotate("text", x = critical\_value\_lower - 0.5, y = 0.1, label = sprintf("Q(%.3f)=%.2f", quantiles[1], critical\_value\_lower), hjust = 1, angle = 90) +  
 annotate("text", x = critical\_value\_upper - 0.5, y = 0.1, label = sprintf("Q(%.3f)=%.2f", quantiles[2], critical\_value\_upper), hjust = 1, angle = 90) +  
 annotate("text", x = observed\_chi\_squared - 0.5, y = 0.1, label = 'наблюдаемое', hjust = 1, angle = 90) +  
 annotate("text", x = critical\_value\_upper + 5, y = 0.1, label = sprintf("df = %.0f", df), hjust = 1)



### F критерий для разницы дисперсий (точный)

# Загрузка необходимых библиотек  
library(ggplot2)  
  
# Определение параметров степеней свободы для F-распределения  
df1 <- 10 # Степени свободы числителя  
df2 <- 10 # Степени свободы знаменателя  
  
# Создание последовательности значений x  
x\_values <- seq(0.01, 5, by = 0.01) # Подстраивайте диапазон и шаг по мере необходимости  
  
# Расчет значений PDF для F-распределения  
pdf\_fisher <- df(x\_values, df1, df2)  
  
# Определение уровня значимости (альфа) для критических значений  
quantiles <- c(0, 0.95) # Подстраивайте по мере необходимости  
  
# Расчет критических значений для F-распределения  
critical\_value\_lower <- qf(quantiles[1], df1, df2)  
critical\_value\_upper <- qf(quantiles[2], df1, df2)  
  
# Расчет наблюдаемого значения F (замените на ваше фактическое наблюдаемое значение)  
observed\_f\_value <- 1 # Замените на ваше реальное наблюдаемое значение  
  
# Создание data frame для построения графика  
pdf\_data <- data.frame(x = x\_values, Probability = pdf\_fisher)  
  
# Создание графика PDF для визуализации F-распределения  
ggplot(pdf\_data, aes(x = x, y = Probability)) +  
 geom\_line(size = 1, color = "black") +  
 labs(title = "PDF F-распределения Фишера-Снедекора", x = "x", y = "Плотность вероятности") +  
 theme\_minimal() +  
 geom\_vline(xintercept = c(critical\_value\_lower, critical\_value\_upper, observed\_f\_value),  
 linetype = c("solid", "solid", "dashed"),  
 color = c("blue", "blue", "red"),  
 linewidth = c(2, 2, 2)) +  
 annotate("text", x = critical\_value\_lower-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[1], critical\_value\_lower), hjust = 0, angle = 90) +  
 annotate("text", x = critical\_value\_upper-0.15, y = 0.2, label = sprintf("Q(%.3f)=%.2f", quantiles[2], critical\_value\_upper), hjust = 0, angle = 90) +  
 annotate("text", x = observed\_f\_value-0.15, y = 0.2, label = 'наблюдаемое', hjust = 0, angle = 90) +  
 annotate("text", x = 4, y = 0.6, label = sprintf("df1=%.0f, df2=%.0f", df1, df2), hjust = 0)



### Тест Фишера для отношения дисперсий vs t-test

# Sample data for group A and group B  
group\_A <- c(15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25)  
group\_B <- c(10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30)  
  
?var.test  
  
  
# Perform an F-test for equality of variances  
f\_test\_result <- var.test(group\_B, group\_A)  
  
# Perform a t-test for the means of two independent samples (assuming unequal variances)  
t\_test\_result <- t.test(group\_B, group\_A, var.equal = FALSE)  
  
# Print the F-test and t-test results  
cat("F-Test Result (Equality of Variances):\n")

## F-Test Result (Equality of Variances):

print(f\_test\_result)

##   
## F test to compare two variances  
##   
## data: group\_B and group\_A  
## F = 4, num df = 10, denom df = 10, p-value = 0.03916  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 1.076197 14.867167  
## sample estimates:  
## ratio of variances   
## 4

cat("\nT-Test Result (Equality of Means):\n")

##   
## T-Test Result (Equality of Means):

print(t\_test\_result)

##   
## Welch Two Sample t-test  
##   
## data: group\_B and group\_A  
## t = 0, df = 14.706, p-value = 1  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -4.774382 4.774382  
## sample estimates:  
## mean of x mean of y   
## 20 20

# Check if the variances are significantly different  
if (f\_test\_result$p.value < 0.05) {  
 cat("The variances are significantly different, reject H0 for variances.\n")  
} else {  
 cat("There is no significant difference in variances, fail to reject H0 for variances.\n")  
}

## The variances are significantly different, reject H0 for variances.

# Check if the means are significantly different  
if (t\_test\_result$p.value < 0.05) {  
 cat("The means are significantly different, reject H0 for means.\n")  
} else {  
 cat("There is no significant difference in means, fail to reject H0 for means.\n")  
}

## There is no significant difference in means, fail to reject H0 for means.