automatization\_notebook\_03

2024-10-27

# Чтение данных

В вашем варианте нужно использовать датасет framingham.

data <- read\_csv("data/raw/framingham.csv")

## Rows: 4238 Columns: 16  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## dbl (16): male, age, education, currentSmoker, cigsPerDay, BPMeds, prevalent...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Выведите общее описание данных

summary(data)

## male age education currentSmoker   
## Min. :0.0000 Min. :32.00 Min. :1.000 Min. :0.0000   
## 1st Qu.:0.0000 1st Qu.:42.00 1st Qu.:1.000 1st Qu.:0.0000   
## Median :0.0000 Median :49.00 Median :2.000 Median :0.0000   
## Mean :0.4292 Mean :49.58 Mean :1.979 Mean :0.4941   
## 3rd Qu.:1.0000 3rd Qu.:56.00 3rd Qu.:3.000 3rd Qu.:1.0000   
## Max. :1.0000 Max. :70.00 Max. :4.000 Max. :1.0000   
## NA's :105   
## cigsPerDay BPMeds prevalentStroke prevalentHyp   
## Min. : 0.000 Min. :0.00000 Min. :0.000000 Min. :0.0000   
## 1st Qu.: 0.000 1st Qu.:0.00000 1st Qu.:0.000000 1st Qu.:0.0000   
## Median : 0.000 Median :0.00000 Median :0.000000 Median :0.0000   
## Mean : 9.003 Mean :0.02963 Mean :0.005899 Mean :0.3105   
## 3rd Qu.:20.000 3rd Qu.:0.00000 3rd Qu.:0.000000 3rd Qu.:1.0000   
## Max. :70.000 Max. :1.00000 Max. :1.000000 Max. :1.0000   
## NA's :29 NA's :53   
## diabetes totChol sysBP diaBP   
## Min. :0.00000 Min. :107.0 Min. : 83.5 Min. : 48.00   
## 1st Qu.:0.00000 1st Qu.:206.0 1st Qu.:117.0 1st Qu.: 75.00   
## Median :0.00000 Median :234.0 Median :128.0 Median : 82.00   
## Mean :0.02572 Mean :236.7 Mean :132.4 Mean : 82.89   
## 3rd Qu.:0.00000 3rd Qu.:263.0 3rd Qu.:144.0 3rd Qu.: 89.88   
## Max. :1.00000 Max. :696.0 Max. :295.0 Max. :142.50   
## NA's :50   
## BMI heartRate glucose TenYearCHD   
## Min. :15.54 Min. : 44.00 Min. : 40.00 Min. :0.000   
## 1st Qu.:23.07 1st Qu.: 68.00 1st Qu.: 71.00 1st Qu.:0.000   
## Median :25.40 Median : 75.00 Median : 78.00 Median :0.000   
## Mean :25.80 Mean : 75.88 Mean : 81.97 Mean :0.152   
## 3rd Qu.:28.04 3rd Qu.: 83.00 3rd Qu.: 87.00 3rd Qu.:0.000   
## Max. :56.80 Max. :143.00 Max. :394.00 Max. :1.000   
## NA's :19 NA's :1 NA's :388

# Очистка данных

1. Уберите переменные, в которых пропущенных значений больше 20% или уберите субъектов со слишком большим количеством пропущенных значений. Или совместите оба варианта. Напишите обоснование, почему вы выбрали тот или иной вариант:

**Обоснование**:

Я бы совместила оба подхода, если в данных есть как переменные, так и субъекты с большим количеством пропусков. Это поможет устранить наиболее проблемные данные (например, переменные с большим количеством пропусков и субъекты с чрезмерным числом пропусков), при этом сохранив достаточный объем информации как о переменных, так и о субъектах.

*Переменные с более чем 20% пропущенных значений:* Эти переменные вряд ли смогут дать надежную информацию, особенно если они важны для анализа, поэтому их удаление целесообразно.

*Субъекты с большим количеством пропусков:* Если субъект имеет много пропусков, то данные о нем могут быть недостаточно полными для анализа. Это особенно критично в тех случаях, когда важные переменные отсутствуют.

Однако, в моем датасете, после проведения проверки не обнаруженно переменных, в которых пропущенных значений более 20% или субъектов со слишком большим количеством пропущенных значений.

1. Переименуйте переменные в человекочитаемый вид (что делать с пробелами в названиях?); +
2. В соответствии с описанием данных приведите переменные к нужному типу (numeric или factor); +
3. Отсортируйте данные по возрасту по убыванию; +
4. Сохраните в файл outliers.csv субъектов, которые являются выбросами (например, по правилу трёх сигм) — это необязательное задание со звёздочкой; +
5. Присвойте получившийся датасет переменной “cleaned\_data”. +

# Удаляем переменные с более 20% пропущенных значений  
na\_percentage\_cols <- colSums(is.na(data)) / nrow(data)  
cleaned\_data <- data[, na\_percentage\_cols <= 0.2]  
  
# Удаляем субъектов с болле чем 20% пропущенных значений  
row\_na\_percentage <- rowSums(is.na(cleaned\_data)) / ncol(cleaned\_data)  
cleaned\_data <- cleaned\_data[row\_na\_percentage <= 0.2, ]  
  
# Переимнование переменных   
cleaned\_data <- cleaned\_data %>%  
 rename(`Пол` = male,  
 `Возраст` = age,  
 `Уровень\_образования` = education,  
 `Курильщик` = currentSmoker,  
 `Количество\_сигарет\_в\_день` = cigsPerDay,  
 `Прием\_лекарств` = BPMeds,  
 `Инсульт` = prevalentStroke,  
 `Гипертония` = prevalentHyp,  
 `Диабет` = diabetes,  
 `Общий\_холестерин` = totChol,  
 `САД` = sysBP,  
 `ДАД` = diaBP,  
 `ИМТ` = BMI,  
 `ЧСС` = heartRate,  
 `Глюкоза` = glucose,  
 `Риск\_СЗ\_на\_10\_лет` = TenYearCHD) %>%  
 mutate(`Пол` = as.factor(`Пол`),  
 `Уровень\_образования` = as.factor(`Уровень\_образования`),  
 `Курильщик` = as.factor(`Курильщик`),  
 `Прием\_лекарств` = as.factor(`Прием\_лекарств`),  
 `Инсульт` = as.factor(`Инсульт`),  
 `Гипертония` = as.factor(`Гипертония`),  
 `Диабет` = as.factor(`Диабет`),  
 `Риск\_СЗ\_на\_10\_лет` = as.factor(`Риск\_СЗ\_на\_10\_лет`)) %>%  
 arrange(`Возраст`, .desc = TRUE)  
  
# Идентифицируем количесвтенные переменные   
numeric\_vars <- sapply(cleaned\_data, is.numeric)  
  
# Применяем правило 3-сигма для определения выбросов  
outliers <- cleaned\_data %>%  
 filter(  
 if\_any(all\_of(names(cleaned\_data)[numeric\_vars]), ~ . < (mean(., na.rm = TRUE) - 3 \* sd(., na.rm = TRUE)) |  
 . > (mean(., na.rm = TRUE) + 3 \* sd(., na.rm = TRUE)))  
 )  
  
# Сохраняем субъектов с выбросами в файл outliers.csv  
write.csv(outliers, "outliers.csv", row.names = FALSE)

# Сколько осталось переменных?

Число переменных не изменилось.

print(ncol(cleaned\_data))

## [1] 16

# Сколько осталось случаев?

Число случаев не изменилось.

print(nrow(cleaned\_data))

## [1] 4238

# Есть ли в данных идентичные строки?

Идентичные строки отсутствуют.

duplicate\_rows <- duplicated(cleaned\_data)  
any(duplicate\_rows)

## [1] FALSE

# Сколько всего переменных с пропущенными значениями в данных и сколько пропущенных точек в каждой такой переменной?

# Подсчитываем пропущенные значения для каждой переменной  
missing\_values\_per\_var<- colSums(is.na(cleaned\_data))  
  
# Сортируем только переменные, которые содержат пропущенные значения  
variables\_with\_missing <- missing\_values\_per\_var[missing\_values\_per\_var > 0]  
variables\_with\_missing

## Уровень\_образования Количество\_сигарет\_в\_день Прием\_лекарств   
## 105 29 53   
## Общий\_холестерин ИМТ ЧСС   
## 50 19 1   
## Глюкоза   
## 388

# Описательные статистики

## Количественные переменные

1. Рассчитайте для всех количественных переменных для каждой группы (TenYearCHD):

1.1) Количество значений;

1.2) Количество пропущенных значений;

1.3) Среднее;

1.4) Медиану;

1.5) Стандартное отклонение;

1.6) 25% квантиль и 75% квантиль;

1.7) Интерквартильный размах;

1.8) Минимум;

1.9) Максимум;

1.10) 95% ДИ для среднего - задание со звёздочкой.

# Функция для расчета 95% ДИдля среднего  
ci\_95 <- function(x) {  
 n <- sum(!is.na(x))  
 if (n < 3) return("Н/П\*")  
 se <- sd(x, na.rm = TRUE) / sqrt(n) # Standard Error  
 mean\_x <- mean(x, na.rm = TRUE)  
 ci <- c(mean\_x - 1.96 \* se, mean\_x + 1.96 \* se)  
 paste0(round(ci[1], 2), " - ", round(ci[2], 2))  
}  
  
# Список описательных статистик  
statistics <- list(  
 `\_Количество значений` = ~as.character(sum(!is.na(.x))),  
 `\_Нет данных` = ~as.character(sum(is.na(.x))),  
 `\_Среднее` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", as.character(mean(.x, na.rm = TRUE) %>% round(2))),  
 `\_Медиана` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", as.character(median(.x, na.rm = TRUE) %>% round(2))),  
 `\_Станд. откл.` = ~ifelse(sum(!is.na(.x)) < 3, "Н/П\*", as.character(sd(.x, na.rm = TRUE) %>% round(2))),  
 `\_Q1 - Q3` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", paste0(as.character(quantile(.x, 0.25, na.rm = TRUE) %>% round(2)), " - ", as.character(quantile(.x, 0.75, na.rm = TRUE) %>% round(2)))),  
 `\_IQR` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", as.character(IQR(.x, na.rm = TRUE) %>% round(2))),  
 `\_мин.` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", as.character(min(.x, na.rm = TRUE) %>% round(2))),  
 `\_макс.` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", as.character(max(.x, na.rm = TRUE) %>% round(2))),  
 `\_95% ДИ для среднего` = ~ci\_95(.x)  
)  
  
cleaned\_data %>%  
 select(`Риск\_СЗ\_на\_10\_лет`, where(is.numeric)) %>%  
 group\_by(`Риск\_СЗ\_на\_10\_лет`) %>%  
 summarise(across(where(is.numeric), statistics)) %>%  
 pivot\_longer(!`Риск\_СЗ\_на\_10\_лет`, names\_sep = "\_\_", names\_to = c("Переменная", "Статистика")) %>%  
 rename(`Значение` = value) %>%  
 flextable() %>%  
 theme\_zebra() %>%  
 merge\_v(c("Риск\_СЗ\_на\_10\_лет", "Переменная")) %>%  
 autofit() %>%  
 set\_table\_properties(width = 0.9, layout = "autofit")

| **Риск\_СЗ\_на\_10\_лет** | **Переменная** | **Статистика** | **Значение** |
| --- | --- | --- | --- |
| 0 | Возраст | Количество значений | 3594 |
| Нет данных | 0 |
| Среднее | 48.77 |
| Медиана | 48 |
| Станд. откл. | 8.41 |
| Q1 - Q3 | 42 - 55 |
| IQR | 13 |
| мин. | 32 |
| макс. | 70 |
| 95% ДИ для среднего | 48.49 - 49.04 |
| Количество\_сигарет\_в\_день | Количество значений | 3567 |
| Нет данных | 27 |
| Среднее | 8.71 |
| Медиана | 0 |
| Станд. откл. | 11.69 |
| Q1 - Q3 | 0 - 20 |
| IQR | 20 |
| мин. | 0 |
| макс. | 70 |
| 95% ДИ для среднего | 8.33 - 9.09 |
| Общий\_холестерин | Количество значений | 3553 |
| Нет данных | 41 |
| Среднее | 235.17 |
| Медиана | 232 |
| Станд. откл. | 43.77 |
| Q1 - Q3 | 205 - 261 |
| IQR | 56 |
| мин. | 113 |
| макс. | 696 |
| 95% ДИ для среднего | 233.73 - 236.61 |
| САД | Количество значений | 3594 |
| Нет данных | 0 |
| Среднее | 130.33 |
| Медиана | 127 |
| Станд. откл. | 20.46 |
| Q1 - Q3 | 116 - 141 |
| IQR | 25 |
| мин. | 83.5 |
| макс. | 243 |
| 95% ДИ для среднего | 129.66 - 131 |
| ДАД | Количество значений | 3594 |
| Нет данных | 0 |
| Среднее | 82.16 |
| Медиана | 81 |
| Станд. откл. | 11.34 |
| Q1 - Q3 | 74 - 88 |
| IQR | 14 |
| мин. | 50 |
| макс. | 142.5 |
| 95% ДИ для среднего | 81.79 - 82.53 |
| ИМТ | Количество значений | 3585 |
| Нет данных | 9 |
| Среднее | 25.67 |
| Медиана | 25.25 |
| Станд. откл. | 3.98 |
| Q1 - Q3 | 23.01 - 27.87 |
| IQR | 4.86 |
| мин. | 15.54 |
| макс. | 51.28 |
| 95% ДИ для среднего | 25.54 - 25.8 |
| ЧСС | Количество значений | 3594 |
| Нет данных | 0 |
| Среднее | 75.76 |
| Медиана | 75 |
| Станд. откл. | 11.99 |
| Q1 - Q3 | 68 - 83 |
| IQR | 15 |
| мин. | 44 |
| макс. | 143 |
| 95% ДИ для среднего | 75.37 - 76.15 |
| Глюкоза | Количество значений | 3256 |
| Нет данных | 338 |
| Среднее | 80.68 |
| Медиана | 78 |
| Станд. откл. | 18.97 |
| Q1 - Q3 | 71 - 86 |
| IQR | 15 |
| мин. | 40 |
| макс. | 386 |
| 95% ДИ для среднего | 80.03 - 81.33 |
| 1 | Возраст | Количество значений | 644 |
| Нет данных | 0 |
| Среднее | 54.15 |
| Медиана | 55 |
| Станд. откл. | 8.01 |
| Q1 - Q3 | 48 - 61 |
| IQR | 13 |
| мин. | 35 |
| макс. | 70 |
| 95% ДИ для среднего | 53.53 - 54.76 |
| Количество\_сигарет\_в\_день | Количество значений | 642 |
| Нет данных | 2 |
| Среднее | 10.63 |
| Медиана | 2.5 |
| Станд. откл. | 13.01 |
| Q1 - Q3 | 0 - 20 |
| IQR | 20 |
| мин. | 0 |
| макс. | 60 |
| 95% ДИ для среднего | 9.62 - 11.64 |
| Общий\_холестерин | Количество значений | 635 |
| Нет данных | 9 |
| Среднее | 245.39 |
| Медиана | 241 |
| Станд. откл. | 48.08 |
| Q1 - Q3 | 213.5 - 272 |
| IQR | 58.5 |
| мин. | 107 |
| макс. | 600 |
| 95% ДИ для среднего | 241.65 - 249.13 |
| САД | Количество значений | 644 |
| Нет данных | 0 |
| Среднее | 143.62 |
| Медиана | 139 |
| Станд. откл. | 26.69 |
| Q1 - Q3 | 125 - 158 |
| IQR | 33 |
| мин. | 83.5 |
| макс. | 295 |
| 95% ДИ для среднего | 141.56 - 145.68 |
| ДАД | Количество значений | 644 |
| Нет данных | 0 |
| Среднее | 86.98 |
| Медиана | 85.5 |
| Станд. откл. | 14.03 |
| Q1 - Q3 | 78 - 95 |
| IQR | 17 |
| мин. | 48 |
| макс. | 140 |
| 95% ДИ для среднего | 85.9 - 88.06 |
| ИМТ | Количество значений | 634 |
| Нет данных | 10 |
| Среднее | 26.53 |
| Медиана | 26.16 |
| Станд. откл. | 4.52 |
| Q1 - Q3 | 23.5 - 28.89 |
| IQR | 5.4 |
| мин. | 15.96 |
| макс. | 56.8 |
| 95% ДИ для среднего | 26.18 - 26.88 |
| ЧСС | Количество значений | 643 |
| Нет данных | 1 |
| Среднее | 76.53 |
| Медиана | 75 |
| Станд. откл. | 12.22 |
| Q1 - Q3 | 68 - 85 |
| IQR | 17 |
| мин. | 50 |
| макс. | 120 |
| 95% ДИ для среднего | 75.59 - 77.47 |
| Глюкоза | Количество значений | 594 |
| Нет данных | 50 |
| Среднее | 89.01 |
| Медиана | 79 |
| Станд. откл. | 41.14 |
| Q1 - Q3 | 72 - 90 |
| IQR | 18 |
| мин. | 40 |
| макс. | 394 |
| 95% ДИ для среднего | 85.7 - 92.32 |

## Категориальные переменные

1. Рассчитайте для всех категориальных переменных для каждой группы (TenYearCHD):

1.1) Абсолютное количество;

1.2) Относительное количество внутри группы;

1.3) 95% ДИ для доли внутри группы - задание со звёздочкой.

# Функция для расчета 95% ДИ для доли  
ci\_prop\_95 <- function(x, n) {  
 p <- x / n  
 se <- sqrt((p \* (1 - p)) / n)  
 ci\_lower <- p - 1.96 \* se  
 ci\_upper <- p + 1.96 \* se  
 paste0(round(100 \* ci\_lower, 2), "% - ", round(100 \* ci\_upper, 2), "%")  
}  
  
cleaned\_data %>%  
 select(`Риск\_СЗ\_на\_10\_лет`, where(is.factor)) %>%  
 mutate(across(where(is.factor), ~ as.character(.) %>% replace\_na("Нет данных") %>% as.factor())) %>%  
 pivot\_longer(-`Риск\_СЗ\_на\_10\_лет`, names\_to = "Переменная", values\_to = "Значение") %>%  
 group\_by(`Риск\_СЗ\_на\_10\_лет`, Переменная, Значение) %>%  
 summarise(n = n(), .groups = 'drop') %>%  
 group\_by(`Риск\_СЗ\_на\_10\_лет`, Переменная) %>%  
 mutate(`Процент по группе` = paste0(round((n / sum(n)) \* 100, 2), "%"),  
 `95% ДИ для доли` = ci\_prop\_95(n, sum(n)) ) %>%  
 ungroup() %>%  
 select(`Риск\_СЗ\_на\_10\_лет`, Переменная, Значение, n, `Процент по группе`, `95% ДИ для доли`) %>%  
 arrange(`Риск\_СЗ\_на\_10\_лет`, Переменная, Значение) %>%  
 flextable() %>%  
 theme\_zebra() %>%  
 merge\_v(c("Риск\_СЗ\_на\_10\_лет", "Переменная")) %>%  
 autofit() %>%  
 set\_table\_properties(width = 0.9, layout = "autofit")

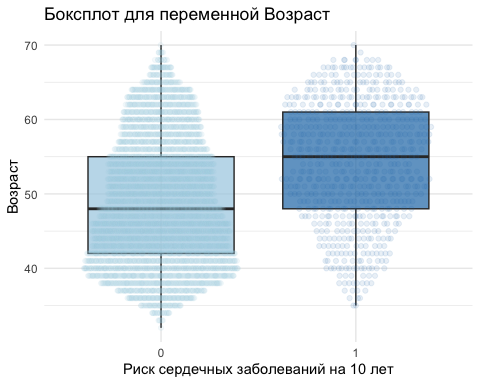
| **Риск\_СЗ\_на\_10\_лет** | **Переменная** | **Значение** | **n** | **Процент по группе** | **95% ДИ для доли** |
| --- | --- | --- | --- | --- | --- |
| 0 | Гипертония | 0 | 2,603 | 72.43% | 70.97% - 73.89% |
| 1 | 991 | 27.57% | 26.11% - 29.03% |
| Диабет | 0 | 3,525 | 98.08% | 97.63% - 98.53% |
| 1 | 69 | 1.92% | 1.47% - 2.37% |
| Инсульт | 0 | 3,580 | 99.61% | 99.41% - 99.81% |
| 1 | 14 | 0.39% | 0.19% - 0.59% |
| Курильщик | 0 | 1,833 | 51% | 49.37% - 52.64% |
| 1 | 1,761 | 49% | 47.36% - 50.63% |
| Пол | 0 | 2,118 | 58.93% | 57.32% - 60.54% |
| 1 | 1,476 | 41.07% | 39.46% - 42.68% |
| Прием\_лекарств | 0 | 3,469 | 96.52% | 95.92% - 97.12% |
| 1 | 83 | 2.31% | 1.82% - 2.8% |
| Нет данных | 42 | 1.17% | 0.82% - 1.52% |
| Уровень\_образования | 1 | 1,397 | 38.87% | 37.28% - 40.46% |
| 2 | 1,106 | 30.77% | 29.26% - 32.28% |
| 3 | 599 | 16.67% | 15.45% - 17.89% |
| 4 | 403 | 11.21% | 10.18% - 12.24% |
| Нет данных | 89 | 2.48% | 1.97% - 2.98% |
| 1 | Гипертония | 0 | 319 | 49.53% | 45.67% - 53.4% |
| 1 | 325 | 50.47% | 46.6% - 54.33% |
| Диабет | 0 | 604 | 93.79% | 91.92% - 95.65% |
| 1 | 40 | 6.21% | 4.35% - 8.08% |
| Инсульт | 0 | 633 | 98.29% | 97.29% - 99.29% |
| 1 | 11 | 1.71% | 0.71% - 2.71% |
| Курильщик | 0 | 311 | 48.29% | 44.43% - 52.15% |
| 1 | 333 | 51.71% | 47.85% - 55.57% |
| Пол | 0 | 301 | 46.74% | 42.89% - 50.59% |
| 1 | 343 | 53.26% | 49.41% - 57.11% |
| Прием\_лекарств | 0 | 592 | 91.93% | 89.82% - 94.03% |
| 1 | 41 | 6.37% | 4.48% - 8.25% |
| Нет данных | 11 | 1.71% | 0.71% - 2.71% |
| Уровень\_образования | 1 | 323 | 50.16% | 46.29% - 54.02% |
| 2 | 147 | 22.83% | 19.58% - 26.07% |
| 3 | 88 | 13.66% | 11.01% - 16.32% |
| 4 | 70 | 10.87% | 8.47% - 13.27% |
| Нет данных | 16 | 2.48% | 1.28% - 3.69% |

# Визуализация

## Количественные переменные

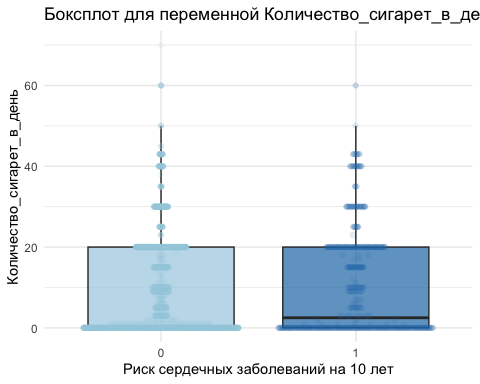
1. Для каждой количественной переменной сделайте боксплоты по группам. Расположите их либо на отдельных рисунках, либо на одном, но читаемо;
2. Наложите на боксплоты beeplots - задание со звёздочкой.
3. Раскрасьте боксплоты с помощью библиотеки RColorBrewer.

# Создаем переменную с количественными переменными  
quantitative\_vars <- cleaned\_data %>%  
 select(where(is.numeric))  
  
# Создаем графики для количественных переменных  
p <- list()  
  
# Создаем боксплот для каждой количественной переменной  
for (var in names(quantitative\_vars)) {  
 p[[var]] <- ggplot(cleaned\_data, aes(x = `Риск\_СЗ\_на\_10\_лет`, y = .data[[var]], fill = `Риск\_СЗ\_на\_10\_лет`))+  
 geom\_boxplot(outlier.shape = NA, alpha = 0.7)+  
 geom\_quasirandom(aes(color = `Риск\_СЗ\_на\_10\_лет`), method = "quasirandom", size = 1.5, alpha = 0.1)+  
 scale\_fill\_brewer(palette = "Paired") +  
 scale\_color\_brewer(palette = "Paired") +  
 labs(x = 'Риск сердечных заболеваний на 10 лет', y = var, title = paste('Боксплот для переменной', var))+  
 theme\_minimal() +  
 theme(legend.position = "none")  
}  
  
# Отобразим боксплоты на отдельных графиках  
for (plot in p) {  
 print(plot)  
}



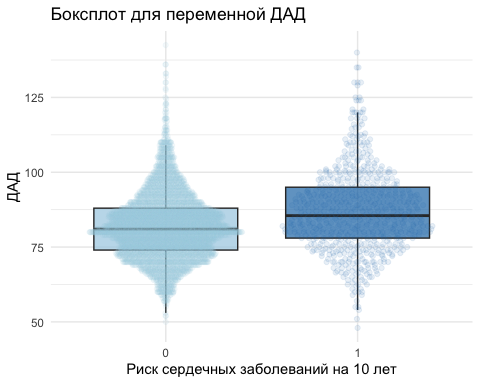
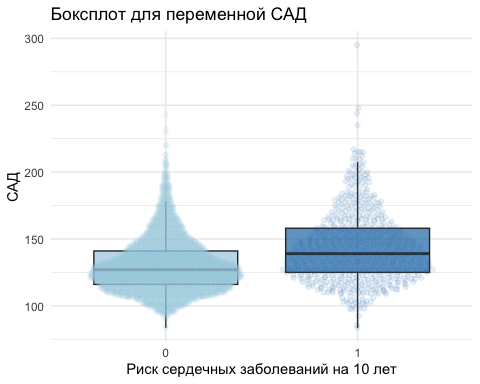
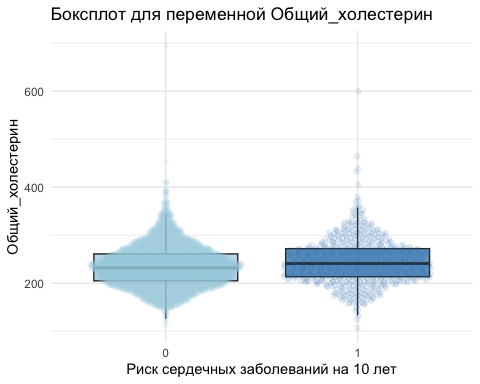
## Warning: Removed 29 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 29 rows containing missing values or values outside the scale range  
## (`position\_quasirandom()`).



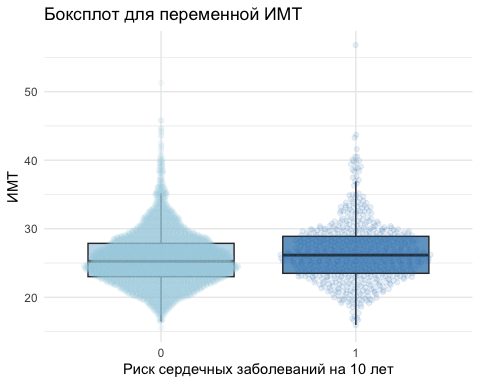
## Warning: Removed 50 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 50 rows containing missing values or values outside the scale range  
## (`position\_quasirandom()`).



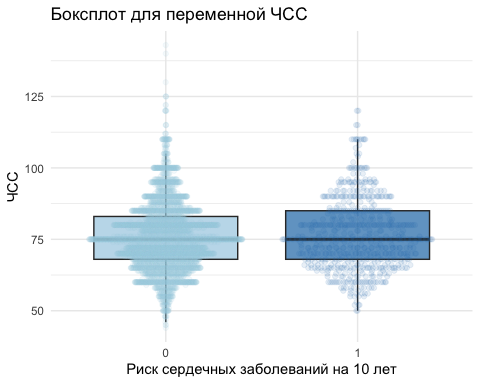
## Warning: Removed 19 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 19 rows containing missing values or values outside the scale range  
## (`position\_quasirandom()`).



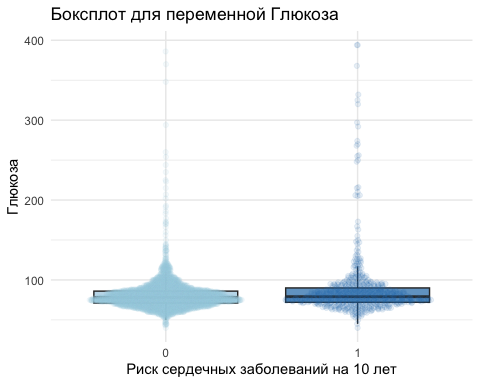
## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 1 row containing missing values or values outside the scale range  
## (`position\_quasirandom()`).



## Warning: Removed 388 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).

## Warning: Removed 388 rows containing missing values or values outside the scale range  
## (`position\_quasirandom()`).



## Категориальные переменные

1. Сделайте подходящие визуализации категориальных переменных. Обоснуйте, почему выбрали именно этот тип.

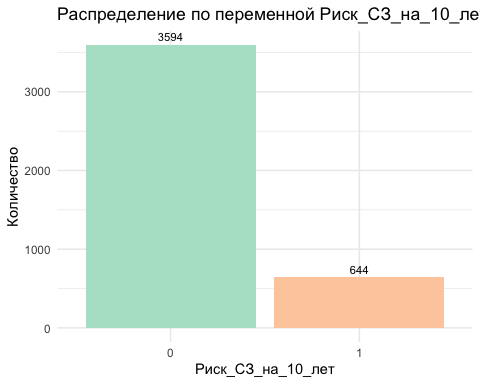
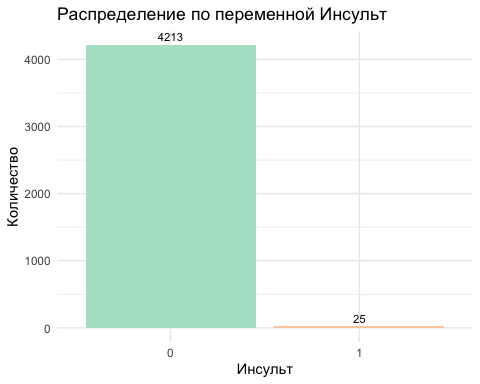
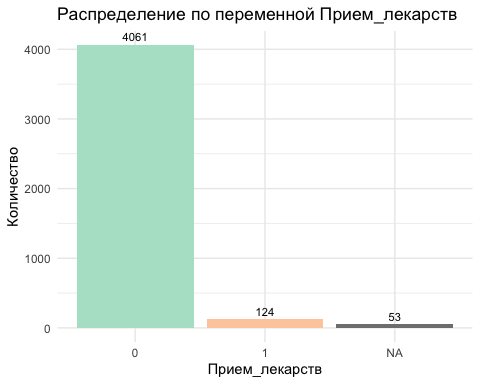
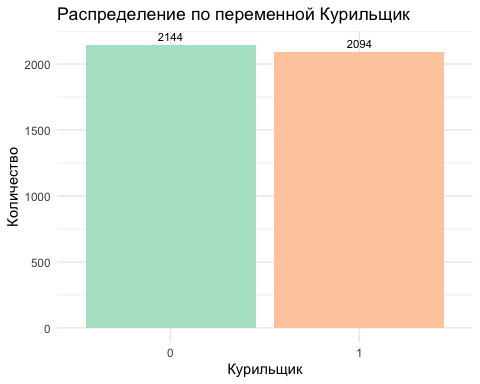
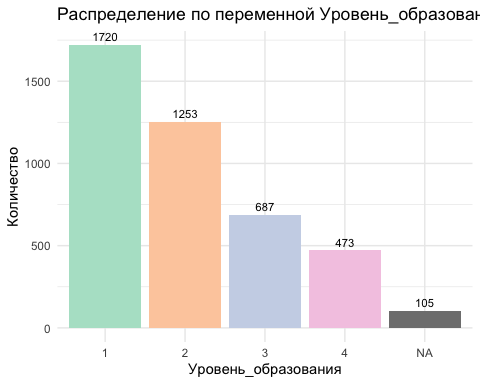
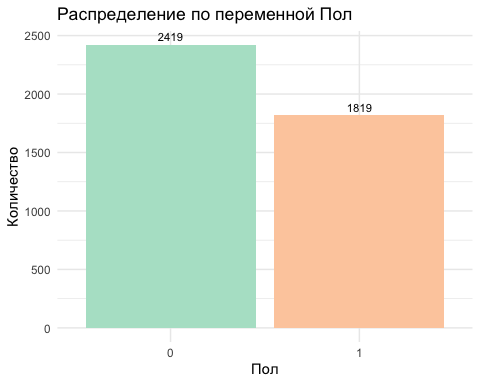
Категориальные переменные хорошо визуализируются с помощью столбчатых диаграмм. Барплоты четко показывают количество или частоты для каждой категории.

# Выбираем категориальные переменные  
categorical\_vars <- cleaned\_data %>%  
 select(where(is.factor))  
  
# Создаем графики для каждой категориальной переменной  
p <- list()  
  
for (var in names(categorical\_vars)) {  
 # Определяем количество уникальных категорий для текущей переменной  
 n\_levels <- nlevels(categorical\_vars[[var]])  
   
 # Определяем цветовую палитру с количеством цветов, равным числу категорий испльзуя RColorBrewer  
 palette <- brewer.pal(min(n\_levels, 8), name = "Pastel2")  
   
 # Строим график  
 p[[var]] <- ggplot(cleaned\_data, aes(x = .data[[var]], fill = .data[[var]])) +  
 geom\_bar() +  
 scale\_fill\_manual(values = palette) +  
 geom\_text(stat = "count", aes(label = ..count..), vjust = -0.5, size = 3) +  
 labs(x = var, y = "Количество", title = paste("Распределение по переменной", var)) +  
 theme\_minimal() +  
 theme(legend.position = "none")  
}

## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels  
## Warning in brewer.pal(min(n\_levels, 8), name = "Pastel2"): minimal value for n is 3, returning requested palette with 3 different levels

# Отображаем все графики  
for (plot in p) {  
 print(plot)  
}

## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.  
## ℹ Please use `after\_stat(count)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# Статистические оценки

## Проверка на нормальность

1. Оцените каждую переменную на соответствие нормальному распределению с помощью теста Шапиро-Уилка. Какие из переменных являются нормальными и как как вы это поняли?

по результатам теста Шапиро-Уилка все переменные не соответствуют нормальному распределению, так как P value > 0.05.

# Применяем тест Шапиро-Уилка ко всем количественным переменным  
shapiro\_results <- sapply(quantitative\_vars, function(x) {  
 if (length(na.omit(x)) < 3) { # Тест требует минимум 3 значений  
 return(NA)  
 } else {  
 return(shapiro.test(x)$p.value)  
 }  
})  
  
shapiro\_results\_df <- data.frame(  
 Переменная = names(shapiro\_results),  
 P\_value = shapiro\_results  
) %>%  
 mutate(Нормальное\_распределение = ifelse(!is.na(P\_value) &P\_value > 0.05, "Нормальное", "Не нормальное")) %>%  
 mutate(P\_value = scientific\_format()(P\_value))  
  
shapiro\_results\_ft <- flextable(shapiro\_results\_df)  
   
  
shapiro\_results\_ft <- shapiro\_results\_ft %>%  
 set\_header\_labels(  
 Переменная = "Переменная",  
 P\_value = "P-значение",  
 Нормальное\_распределение = "Нормальное распределение"  
 ) %>%  
 autofit()  
  
shapiro\_results\_ft

| Переменная | P-значение | Нормальное распределение |
| --- | --- | --- |
| Возраст | 3.96e-30 | Не нормальное |
| Количество\_сигарет\_в\_день | 2.89e-61 | Не нормальное |
| Общий\_холестерин | 2.48e-29 | Не нормальное |
| САД | 1.57e-39 | Не нормальное |
| ДАД | 2.86e-27 | Не нормальное |
| ИМТ | 2.11e-33 | Не нормальное |
| ЧСС | 7.92e-27 | Не нормальное |
| Глюкоза | 3.16e-71 | Не нормальное |

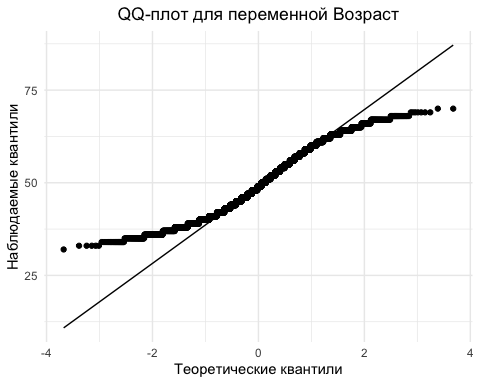
1. Постройте для каждой количественной переменной QQ-плот. Отличаются ли выводы от теста Шапиро-Уилка? Какой метод вы бы предпочли и почему?

Если точки лежат на линии или близко к ней, это указывает на то, что данные имеют нормальное распределение (ни на одном из получившхся у меня графиков такой картины не наблюдается, соотвественно результаты соответствуют таковым при проведении теста Шапиро-Уилка).

Тест Шапиро-Уилка - статистический метод, который даёт чёткий вывод о том, следует ли отклонить гипотезу нормальности. Он полезен, когда нужно формальное заключение. QQ-плот напротив визуальный метод, который показывает детали отклонения данных от нормального распределения.

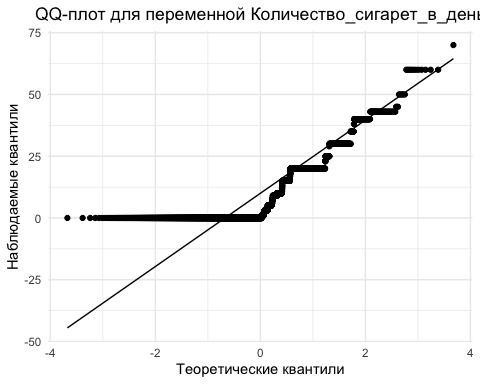
Использование 2 методов совместно является хорошей практикой. Тест Шапиро-Уилка может дать точное заключение, но QQ-плот предоставляет визуальные доказательства, которые могут помочь лучше понять природу распределения данных.

# Построение QQ-плотов для каждой количественной переменной  
for (var in names(quantitative\_vars)) {  
 ggplot(cleaned\_data, aes(sample = .data[[var]])) +  
 stat\_qq() +  
 stat\_qq\_line() +  
 labs(title = paste("QQ-плот для переменной", var),  
 x = "Теоретические квантили", y = "Наблюдаемые квантили") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5)) -> plot  
   
 print(plot)  
}



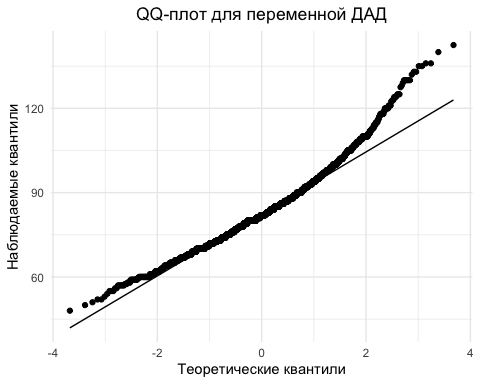
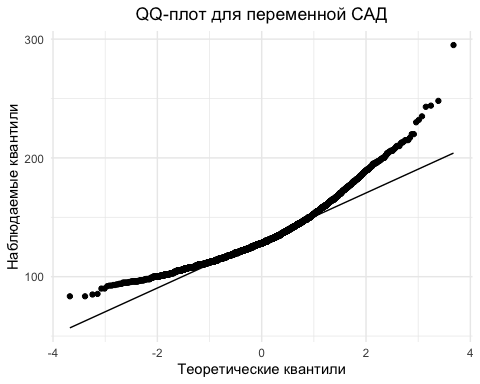
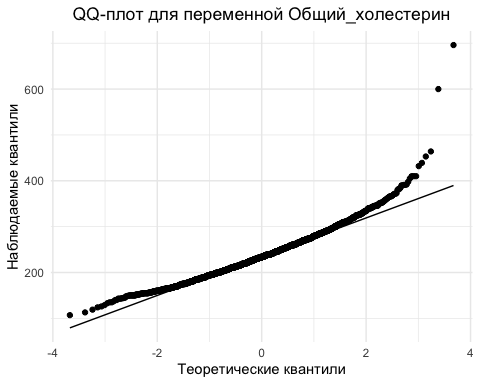
## Warning: Removed 29 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 29 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



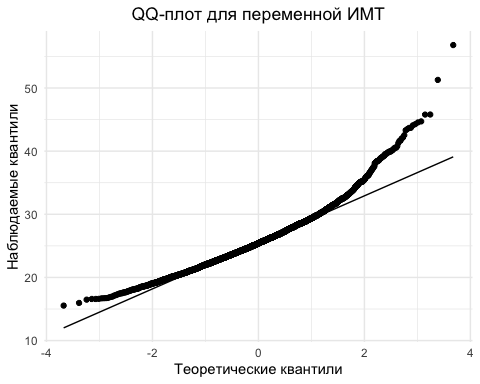
## Warning: Removed 50 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 50 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



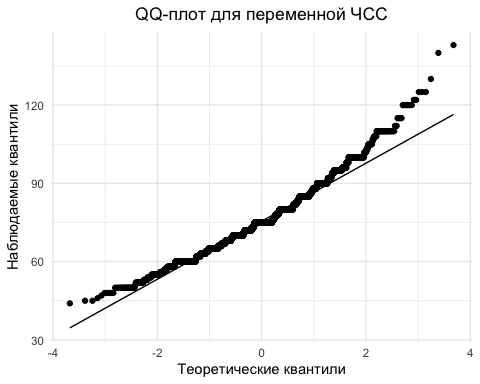
## Warning: Removed 19 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 19 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



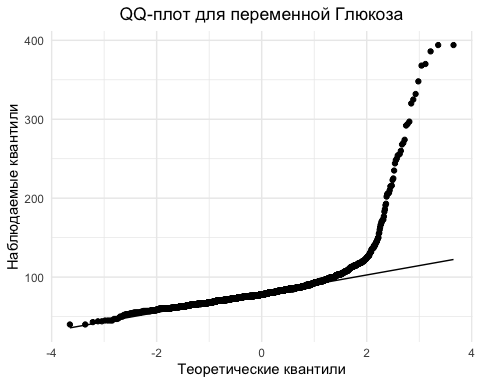
## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 1 row containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



## Warning: Removed 388 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 388 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



1. Ниже напишите, какие ещё методы проверки на нормальность вы знаете и какие у них есть ограничения.
2. Тест Колмогорова-Смирнова (K-S тест) Плюсы: Прост в использовании. Минусы: Чувствителен к большим выборкам, где даже небольшие отклонения могут отклонить гипотезу о нормальности.
3. Тест Андерсона-Дарлинга Плюсы: Более чувствителен к отклонениям в хвостах распределения. Минусы: Слишком чувствителен для больших выборок.

## Сравнение групп

1. Сравните группы (переменная **TenYearCHD**) по каждой переменной (как количественной, так и категориальной). Для каждой переменной выберите нужный критерий и кратко обоснуйте его выбор в комментариях.

grouped\_data <- cleaned\_data %>%  
 group\_by(`Риск\_СЗ\_на\_10\_лет`)  
  
# Результаты для количественных переменных  
quantitative\_results <- lapply(names(quantitative\_vars), function(var) {  
 # Проверка на нормальность для каждой группы Риск\_СЗ\_на\_10\_лет  
 shapiro\_test <- grouped\_data %>%  
 summarise(p\_value = shapiro.test(.data[[var]])$p.value)  
   
 if (all(shapiro\_test$p\_value > 0.05)) {  
 # Используем t-тест для нормальных данных  
 t\_test <- t.test(cleaned\_data[[var]] ~ cleaned\_data$`Риск\_СЗ\_на\_10\_лет`)  
 interpretation <- ifelse(t\_test$p.value < 0.05, "Статистически значимо", "Не значимо")  
 return(data.frame(Переменная = var, Тест = "t-тест", P\_value = t\_test$p.value, Интерпретация = interpretation))  
 } else {  
 # Используем тест Манна-Уитни для ненормальных данных  
 mann\_whitney <- wilcox.test(cleaned\_data[[var]] ~ cleaned\_data$`Риск\_СЗ\_на\_10\_лет`)  
 interpretation <- ifelse(mann\_whitney$p.value < 0.05, "Статистически значимо", "Не значимо")  
 return(data.frame(Переменная = var, Тест = "Mann-Whitney", P\_value = mann\_whitney$p.value, Интерпретация = interpretation))  
 }  
})  
  
# Результаты для категориальных переменных  
categorical\_results <- lapply(names(categorical\_vars), function(var) {  
 # Таблица сопряженности для каждой переменной и группы Риск\_СЗ\_на\_10\_лет  
 contingency\_table <- table(cleaned\_data[[var]], cleaned\_data$`Риск\_СЗ\_на\_10\_лет`)  
   
 if (all(contingency\_table >= 5)) {  
 # Используем хи-квадрат тест для категориальных переменных с достаточными частотами  
 chi\_test <- chisq.test(contingency\_table)  
 interpretation <- ifelse(chi\_test$p.value < 0.05, "Статистически значимо", "Не значимо")  
 return(data.frame(Переменная = var, Тест = "Chi-squared", P\_value = chi\_test$p.value, Интерпретация = interpretation))  
 } else {  
 # Используем точный тест Фишера для малых частот  
 fisher\_test <- fisher.test(contingency\_table)  
 interpretation <- ifelse(fisher\_test$p.value < 0.05, "Статистически значимо", "Не значимо")  
 return(data.frame(Переменная = var, Тест = "Fisher's Exact", P\_value = fisher\_test$p.value, Интерпретация = interpretation))  
 }  
})

## Warning in chisq.test(contingency\_table): Chi-squared approximation may be  
## incorrect

# Объединяем результаты  
quantitative\_results\_df <- do.call(rbind, quantitative\_results) %>%  
 mutate(P\_value = scientific\_format()(P\_value))  
   
categorical\_results\_df <- do.call(rbind, categorical\_results) %>%  
 mutate(P\_value = scientific\_format()(P\_value))  
  
# flextable для количественных переменных  
quantitative\_results\_ft <- flextable(quantitative\_results\_df) %>%  
 set\_header\_labels(  
 Переменная = "Переменная",  
 Тест = "Тест",  
 P\_value = "P-значение",  
 Интерпретация = "Интерпретация"  
 ) %>%  
 theme\_box() %>%  
 autofit()  
  
# flextable для категориальных переменных  
categorical\_results\_ft <- flextable(categorical\_results\_df) %>%  
 set\_header\_labels(  
 Переменная = "Переменная",  
 Тест = "Тест",  
 P\_value = "P-значение",  
 Интерпретация = "Интерпретация"  
 ) %>%  
 theme\_box() %>%  
 autofit()  
  
quantitative\_results\_ft

| **Переменная** | **Тест** | **P-значение** | **Интерпретация** |
| --- | --- | --- | --- |
| Возраст | Mann-Whitney | 1.17e-47 | Статистически значимо |
| Количество\_сигарет\_в\_день | Mann-Whitney | 4.71e-03 | Статистически значимо |
| Общий\_холестерин | Mann-Whitney | 2.44e-07 | Статистически значимо |
| САД | Mann-Whitney | 9.82e-37 | Статистически значимо |
| ДАД | Mann-Whitney | 1.01e-17 | Статистически значимо |
| ИМТ | Mann-Whitney | 1.77e-06 | Статистически значимо |
| ЧСС | Mann-Whitney | 2.41e-01 | Не значимо |
| Глюкоза | Mann-Whitney | 7.50e-04 | Статистически значимо |

categorical\_results\_ft

| **Переменная** | **Тест** | **P-значение** | **Интерпретация** |
| --- | --- | --- | --- |
| Пол | Chi-squared | 1.11e-08 | Статистически значимо |
| Уровень\_образования | Chi-squared | 5.49e-07 | Статистически значимо |
| Курильщик | Chi-squared | 2.21e-01 | Не значимо |
| Прием\_лекарств | Chi-squared | 3.16e-08 | Статистически значимо |
| Инсульт | Chi-squared | 1.81e-04 | Статистически значимо |
| Гипертония | Chi-squared | 1.10e-30 | Статистически значимо |
| Диабет | Chi-squared | 5.64e-10 | Статистически значимо |
| Риск\_СЗ\_на\_10\_лет | Fisher's Exact | 0.00e+00 | Статистически значимо |

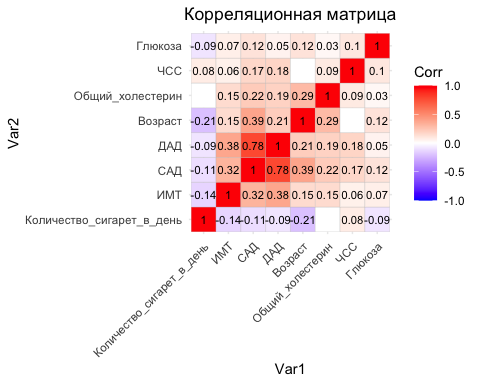
# Далее идут **необязательные** дополнительные задания, которые могут принести вам дополнительные баллы в том числе в случае ошибок в предыдущих

## Корреляционный анализ

1. Создайте корреляционную матрицу с визуализацией и поправкой на множественные сравнения. Объясните, когда лучше использовать корреляционные матрицы и в чём минусы и плюсы корреляционных исследований.

Корреляционные матрицы позволяют быстро выявить взаимосвязи между количественными переменными и понять общие тенденции в данных. Их плюсы — простота и наглядность, но они не выявляют причинно-следственные связи и подвержены влиянию скрытых факторов и выбросов. Корреляционные исследования полезны для первичного анализа, но требуют дополнительных экспериментов для подтверждения взаимосвязей.

# Определяем cor\_test с обработкой ошибок  
cor\_test <- function(x, y) {  
 tryCatch(cor.test(x, y)$p.value, error = function(e) NA)  
}  
  
# Генерируем корреляционную матрицу  
cor\_matrix <- cor(quantitative\_vars, use = "pairwise.complete.obs", method = "spearman")  
  
# Вычисляем p-значения  
p\_values <- outer(  
 1:ncol(quantitative\_vars),   
 1:ncol(quantitative\_vars),   
 Vectorize(function(i, j) cor\_test(quantitative\_vars[[i]], quantitative\_vars[[j]]))  
)  
  
# Применяем поправку на множественные сравнения  
p\_adj <- p.adjust(p\_values, method = "fdr")  
p\_adj\_matrix <- matrix(p\_adj, ncol = ncol(quantitative\_vars), dimnames = list(colnames(quantitative\_vars), colnames(quantitative\_vars)))  
  
# Если они совпадают, попробуйте отобразить всю матрицу:  
ggcorrplot(cor\_matrix, hc.order = TRUE, type = "full", lab = TRUE,   
 p.mat = p\_adj\_matrix, sig.level = 0.05, insig = "blank",   
 title = "Корреляционная матрица",  
 lab\_size = 3) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(hjust = 0.5),  
 axis.text.x = element\_text(angle = 45, hjust = 1, vjust = 1) # Поворачиваем текст на оси X  
 )



## Моделирование

1. Постройте регрессионную модель для переменной **TenYearCHD**. Опишите процесс построения

# Подготовка данных для модели  
cleaned\_data <- cleaned\_data %>%  
 na.omit() %>%  
 mutate(across(where(is.factor), ~ . %>% fct\_relabel(~ str\_c(": ", .))))  
  
# Построение логистической регрессии  
logit\_model <- glm(`Риск\_СЗ\_на\_10\_лет` ~ ., data = cleaned\_data, family = binomial)  
  
# Форматирование результатов модели  
logit\_results <- tidy(logit\_model, conf.int = TRUE) %>%  
 mutate(across(c(estimate, std.error, statistic, conf.low, conf.high),   
 ~ formatC(.x, format = "f", digits = 2, decimal.mark = ".")),  
 `p.value` = ifelse(`p.value` < 0.001, "<0.001", round(`p.value`, 3)),  
 term = term %>% str\_remove\_all("`") %>% str\_replace("\\(Intercept\\)", "Intercept")) %>%  
 unite("95% CI", conf.low, conf.high, sep = ", ") %>%  
 rename(`Переменная` = term, `Коэффициент` = estimate, `Стд.ошибка` = std.error, `Статистика` = statistic) %>%  
 relocate(`95% CI`, .after = `Коэффициент`)  
  
# Создаем таблицу с flextable и форматируем её  
logit\_results\_ft <- flextable(logit\_results) %>%  
 theme\_box() %>%  
 fontsize(size = 10, part = "all") %>%  
 align(align = "center", part = "all") %>%  
 bold(i = ~ `p.value` %>% str\_extract("\\d.\\d{3}") %>% as.numeric() %>% `<`(0.05), j = "p.value") %>%  
 color(i = ~ `Коэффициент` > 0 & `p.value` %>% str\_extract("\\d.\\d{3}") %>% as.numeric() %>% `<`(0.05), color = "green", j = "Коэффициент") %>%  
 color(i = ~ `Коэффициент` < 0 & `p.value` %>% str\_extract("\\d.\\d{3}") %>% as.numeric() %>% `<`(0.05), color = "red", j = "Коэффициент") %>%  
 autofit() %>%  
 set\_table\_properties(width = 0.9, layout = "autofit")   
  
logit\_results\_ft

| **Переменная** | **Коэффициент** | **95% CI** | **Стд.ошибка** | **Статистика** | **p.value** |
| --- | --- | --- | --- | --- | --- |
| Intercept | -8.26 | -9.66, -6.88 | 0.71 | -11.63 | **<0.001** |
| Пол: 1 | 0.53 | 0.32, 0.75 | 0.11 | 4.87 | **<0.001** |
| Возраст | 0.06 | 0.05, 0.08 | 0.01 | 9.20 | **<0.001** |
| Уровень\_образования: 2 | -0.19 | -0.44, 0.05 | 0.12 | -1.56 | 0.12 |
| Уровень\_образования: 3 | -0.19 | -0.49, 0.10 | 0.15 | -1.29 | 0.197 |
| Уровень\_образования: 4 | -0.06 | -0.39, 0.26 | 0.16 | -0.36 | 0.716 |
| Курильщик: 1 | 0.07 | -0.24, 0.38 | 0.16 | 0.46 | 0.644 |
| Количество\_сигарет\_в\_день | 0.02 | 0.01, 0.03 | 0.01 | 2.89 | **0.004** |
| Прием\_лекарств: 1 | 0.17 | -0.30, 0.62 | 0.23 | 0.70 | 0.481 |
| Инсульт: 1 | 0.70 | -0.30, 1.66 | 0.49 | 1.43 | 0.152 |
| Гипертония: 1 | 0.23 | -0.04, 0.50 | 0.14 | 1.69 | 0.091 |
| Диабет: 1 | 0.03 | -0.61, 0.63 | 0.32 | 0.08 | 0.934 |
| Общий\_холестерин | 0.00 | 0.00, 0.00 | 0.00 | 2.10 | **0.036** |
| САД | 0.02 | 0.01, 0.02 | 0.00 | 4.05 | **<0.001** |
| ДАД | -0.00 | -0.02, 0.01 | 0.01 | -0.64 | 0.525 |
| ИМТ | 0.01 | -0.02, 0.03 | 0.01 | 0.40 | 0.687 |
| ЧСС | -0.00 | -0.01, 0.01 | 0.00 | -0.71 | 0.475 |
| Глюкоза | 0.01 | 0.00, 0.01 | 0.00 | 3.23 | **0.001** |