automatization\_notebook

2024-10-20

# Чтение данных

В вашем варианте нужно использовать датасеты cardio\_train\_big или cardio\_train\_not\_too\_big.

cardio <- read\_delim("C:/Users/laskn/BioStat\_2024/data/originals/cardio\_train\_big.csv", delim = ";", locale = locale(decimal\_mark = "."))

## Rows: 70000 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ";"  
## dbl (13): id, age, gender, height, weight, ap\_hi, ap\_lo, cholesterol, gluc, ...  
##   
## 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.

#удалось прочитать данные только так, иначе значения weight читаются неадекватно.

# Выведите общее описание данных

cardio %>%  
 glimpse()

## Rows: 70,000  
## Columns: 13  
## $ id <dbl> 0, 1, 2, 3, 4, 8, 9, 12, 13, 14, 15, 16, 18, 21, 23, 24, 2…  
## $ age <dbl> 18393, 20228, 18857, 17623, 17474, 21914, 22113, 22584, 17…  
## $ gender <dbl> 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2…  
## $ height <dbl> 168, 156, 165, 169, 156, 151, 157, 178, 158, 164, 169, 173…  
## $ weight <dbl> 62, 85, 64, 82, 56, 67, 93, 95, 71, 68, 80, 60, 60, 78, 95…  
## $ ap\_hi <dbl> 110, 140, 130, 150, 100, 120, 130, 130, 110, 110, 120, 120…  
## $ ap\_lo <dbl> 80, 90, 70, 100, 60, 80, 80, 90, 70, 60, 80, 80, 80, 70, 9…  
## $ cholesterol <dbl> 1, 3, 3, 1, 1, 2, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
## $ gluc <dbl> 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1…  
## $ smoke <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1…  
## $ alco <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0…  
## $ active <dbl> 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1…  
## $ cardio <dbl> 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0…

# 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 использовала оба варианта.Мы удаляем только те переменные и строки, которые содержат слишком много пропущенных значений и они могут исказить результаты анализа, как по отдельным переменным, так и по отдельным субъектам. Уменьшение размера датасета также может улучшить производительность при дальнейшем анализе данных.

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”.

sum(is.na(cardio)) #Пропущенных значений в датасете нет, но если бы они были:

## [1] 0

#cardio %>%  
 #select(where(~ mean(is.na(.x)) <= 0.2)) %>% #Удаляем переменные с более 20% пропущенных значений  
 #filter(rowSums(is.na(.)) / ncol(.) <= 0.2) #Удаляем строки, в которых более 20% значений пропущены

cardio %>%  
 rename('systolic\_blood\_pressure'='ap\_hi') %>%   
 rename('diastolic\_blood\_pressure'='ap\_lo') %>%  
 rename(smoking=smoke) %>%   
 rename('alcohol\_intake'=alco) %>%  
 rename('physical\_activity'=active) %>%  
 rename('cardiovascular\_disease'=cardio) %>%  
 relocate('cardiovascular\_disease', .after = id) %>%   
 rename(glucose=gluc)%>%  
 mutate(age\_years = round(age / 365.25, 1)) %>%   
 relocate(age\_years, .after = age) %>%   
 rename('age\_days'=age) %>% #решила пока оставить эту переменную, так как по ней точнее сортировать по возрасту  
 select(-id) %>%  
 mutate(  
 across(c(cholesterol, glucose, smoking, 'alcohol\_intake', 'physical\_activity', 'cardiovascular\_disease'), ~ as.factor(.x)))%>%   
 mutate(   
 across(c(cholesterol, glucose), ~ factor(.x, levels = c(1, 2, 3), labels = c("normal", "above normal", "well above normal")))) %>%  
 mutate(   
 across(c(gender), ~ factor(.x, levels = c(1, 2), labels = c("women", "men")))) %>%  
 mutate(   
 across(c(smoking, 'alcohol\_intake', 'physical\_activity', 'cardiovascular\_disease'), ~ factor(.x, levels = c(1, 0), labels = c("yes", "no")))) %>%  
 arrange(desc(age\_days)) -> cleaned\_data  
  
# Функция для определения выбросов по правилу трёх сигм  
find\_outliers <- function(x) {  
 mean\_x <- mean(x, na.rm = TRUE)  
 sd\_x <- sd(x, na.rm = TRUE)  
 lower\_bound <- mean\_x - 3 \* sd\_x  
 upper\_bound <- mean\_x + 3 \* sd\_x  
 x[x < lower\_bound | x > upper\_bound]  
}  
  
# Поиск выбросов для выбранных переменных  
outliers\_data <- cleaned\_data %>%  
 mutate(  
 across(c(height, weight, systolic\_blood\_pressure, diastolic\_blood\_pressure),  
 ~ ifelse(.x %in% find\_outliers(.x), .x, NA)  
 )  
 ) %>%   
 filter(!is.na(height) | !is.na(weight) |   
 !is.na(systolic\_blood\_pressure) |   
 !is.na(diastolic\_blood\_pressure))   
  
# Сохранение в файл  
write\_csv(outliers\_data, "outliers.csv")  
  
  
#Так как в данных имеются выбросы - значения, явно превосходящие физиологические возможности человека - отрицательное давление,или давление, измеряющееся в десятках тысяч, вес в 10 кг (при том, что все участники взрослые), и эти выбросы, явно внесенные по ошибке, сильно влияют на последующий анализ данных, решено их убрать и в датасете cleaned\_data, заменив на пропущенные значения.  
  
cleaned\_data <- cleaned\_data %>%  
 mutate(  
 across(c(height, weight, systolic\_blood\_pressure, diastolic\_blood\_pressure),  
 ~ ifelse(.x %in% outliers\_data[[cur\_column()]], NA, .x)  
 )  
 )

## Снова пункт 1, но с дастасетом cleaned\_data после замены в нем выбросов на NA

cleaned\_data <- cleaned\_data %>%  
 select(where(~ mean(is.na(.x)) <= 0.2)) %>% #Удаляем переменные с более 20% пропущенных значений  
 filter(rowSums(is.na(.)) / ncol(.) <= 0.2) #Удаляем строки, в которых более 20% значений пропущены

# Сколько осталось переменных?

ncol(cleaned\_data)

## [1] 13

# Сколько осталось случаев?

nrow(cleaned\_data)

## [1] 70000

# Есть ли в данных идентичные строки?

any(duplicated(cleaned\_data))

## [1] TRUE

anyDuplicated(cleaned\_data)

## [1] 10486

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

sum(colSums(is.na(cleaned\_data)) > 0)

## [1] 4

colSums(is.na(cleaned\_data))

## cardiovascular\_disease age\_days age\_years   
## 0 0 0   
## gender height weight   
## 0 293 702   
## systolic\_blood\_pressure diastolic\_blood\_pressure cholesterol   
## 38 951 0   
## glucose smoking alcohol\_intake   
## 0 0 0   
## physical\_activity   
## 0

# Описательные статистики

## Количественные переменные

1. Рассчитайте для всех количественных переменных для каждой группы (наличие или отсутствие сердечно-сосудистых заболеваний):

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.

statistics <- list(  
   
 `\_Количество субъектов` = ~length(.x) %>% as.character(),  
 `\_Количество (есть данные)` = ~sum(!is.na(.x)) %>% as.character(),  
 `\_Нет данных` = ~sum(is.na(.x)) %>% as.character(),  
 `\_Ср. знач.` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", mean(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_Станд. отклон.` = ~ifelse(sum(!is.na(.x)) < 3, "Н/П\*", sd(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_95% ДИ для среднего` = ~{  
 n <- sum(!is.na(.x))  
 ifelse(n < 3, "Н/П\*",   
 paste0(round(mean(.x, na.rm = TRUE) - 1.96 \* sd(.x, na.rm = TRUE) / sqrt(n), 2), " - ", round(mean(.x, na.rm = TRUE) + 1.96 \* sd(.x, na.rm = TRUE) / sqrt(n), 2))  
 )  
 },  
 `\_мин. - макс.` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", paste0(min(.x, na.rm = TRUE) %>% round(2), " - ", max(.x, na.rm = TRUE) %>% round(2))),  
 `\_Медиана` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", median(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_Q1 - Q3` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", paste0(quantile(.x, 0.25, na.rm = TRUE) %>% round(2), " - ", quantile(.x, 0.75, na.rm = TRUE) %>% round(2)))  
 )  
  
cleaned\_data %>%  
 select(cardiovascular\_disease, where(is.numeric)) %>%   
 group\_by(cardiovascular\_disease) %>%   
 summarize(across(where(is.numeric), statistics)) %>%  
 pivot\_longer(!cardiovascular\_disease) %>%  
 separate(name, into = c("variable", "statistics"), sep = "\_\_") %>%   
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%   
 merge\_v(c("cardiovascular\_disease", "variable"))

| **cardiovascular\_disease** | **variable** | **statistics** | **value** |
| --- | --- | --- | --- |
| yes | age\_days | Количество субъектов | 34979 |
| Количество (есть данные) | 34979 |
| Нет данных | 0 |
| Ср. знач. | 20056.81 |
| Станд. отклон. | 2315.93 |
| 95% ДИ для среднего | 20032.54 - 20081.08 |
| мин. - макс. | 14275 - 23713 |
| Медиана | 20384 |
| Q1 - Q3 | 18387 - 21899 |
| age\_years | Количество субъектов | 34979 |
| Количество (есть данные) | 34979 |
| Нет данных | 0 |
| Ср. знач. | 54.91 |
| Станд. отклон. | 6.34 |
| 95% ДИ для среднего | 54.85 - 54.98 |
| мин. - макс. | 39.1 - 64.9 |
| Медиана | 55.8 |
| Q1 - Q3 | 50.3 - 60 |
| height | Количество субъектов | 34979 |
| Количество (есть данные) | 34834 |
| Нет данных | 145 |
| Ср. знач. | 164.31 |
| Станд. отклон. | 7.82 |
| 95% ДИ для среднего | 164.22 - 164.39 |
| мин. - макс. | 140 - 188 |
| Медиана | 165 |
| Q1 - Q3 | 159 - 170 |
| weight | Количество субъектов | 34979 |
| Количество (есть данные) | 34474 |
| Нет данных | 505 |
| Ср. знач. | 76.06 |
| Станд. отклон. | 13.51 |
| 95% ДИ для среднего | 75.92 - 76.21 |
| мин. - макс. | 33 - 117 |
| Медиана | 75 |
| Q1 - Q3 | 66 - 85 |
| systolic\_blood\_pressure | Количество субъектов | 34979 |
| Количество (есть данные) | 34955 |
| Нет данных | 24 |
| Ср. знач. | 134.01 |
| Станд. отклон. | 18.95 |
| 95% ДИ для среднего | 133.81 - 134.21 |
| мин. - макс. | -150 - 401 |
| Медиана | 130 |
| Q1 - Q3 | 120 - 140 |
| diastolic\_blood\_pressure | Количество субъектов | 34979 |
| Количество (есть данные) | 34176 |
| Нет данных | 803 |
| Ср. знач. | 84.62 |
| Станд. отклон. | 10.47 |
| 95% ДИ для среднего | 84.51 - 84.73 |
| мин. - макс. | -70 - 585 |
| Медиана | 80 |
| Q1 - Q3 | 80 - 90 |
| no | age\_days | Количество субъектов | 35021 |
| Количество (есть данные) | 35021 |
| Нет данных | 0 |
| Ср. знач. | 18881.62 |
| Станд. отклон. | 2473.93 |
| 95% ДИ для среднего | 18855.71 - 18907.53 |
| мин. - макс. | 10798 - 23678 |
| Медиана | 19005 |
| Q1 - Q3 | 16838 - 20886 |
| age\_years | Количество субъектов | 35021 |
| Количество (есть данные) | 35021 |
| Нет данных | 0 |
| Ср. знач. | 51.7 |
| Станд. отклон. | 6.77 |
| 95% ДИ для среднего | 51.62 - 51.77 |
| мин. - макс. | 29.6 - 64.8 |
| Медиана | 52 |
| Q1 - Q3 | 46.1 - 57.2 |
| height | Количество субъектов | 35021 |
| Количество (есть данные) | 34873 |
| Нет данных | 148 |
| Ср. знач. | 164.51 |
| Станд. отклон. | 7.62 |
| 95% ДИ для среднего | 164.43 - 164.59 |
| мин. - макс. | 140 - 188 |
| Медиана | 165 |
| Q1 - Q3 | 159 - 170 |
| weight | Количество субъектов | 35021 |
| Количество (есть данные) | 34824 |
| Нет данных | 197 |
| Ср. знач. | 71.28 |
| Станд. отклон. | 12.55 |
| 95% ДИ для среднего | 71.14 - 71.41 |
| мин. - макс. | 32 - 117 |
| Медиана | 70 |
| Q1 - Q3 | 63 - 79 |
| systolic\_blood\_pressure | Количество субъектов | 35021 |
| Количество (есть данные) | 35007 |
| Нет данных | 14 |
| Ср. знач. | 119.37 |
| Станд. отклон. | 14.13 |
| 95% ДИ для среднего | 119.23 - 119.52 |
| мин. - макс. | -120 - 309 |
| Медиана | 120 |
| Q1 - Q3 | 110 - 120 |
| diastolic\_blood\_pressure | Количество субъектов | 35021 |
| Количество (есть данные) | 34873 |
| Нет данных | 148 |
| Ср. знач. | 78.12 |
| Станд. отклон. | 9.04 |
| 95% ДИ для среднего | 78.03 - 78.22 |
| мин. - макс. | 0 - 602 |
| Медиана | 80 |
| Q1 - Q3 | 70 - 80 |

## Категориальные переменные

1. Рассчитайте для всех категориальных переменных для каждой группы (наличие или отсутствие сердечно-сосудистых заболеваний):

1.1) Абсолютное количество;

1.2) Относительное количество внутри группы;

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

cleaned\_data %>%  
 select(cardiovascular\_disease, where(is.factor)) %>%  
 mutate(gender = gender %>% as.character() %>% replace\_na("Нет данных") %>%   
 as.factor()) %>%  
 count(cardiovascular\_disease, gender) %>%  
 group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>%   
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

| **cardiovascular\_disease** | **gender** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | men | 12363 | 35.34% | 34.84% - 35.85% | 17.66% |
| women | 22616 | 64.66% | 64.15% - 65.16% | 32.31% |
| no | men | 12107 | 34.57% | 34.07% - 35.07% | 17.3% |
| women | 22914 | 65.43% | 64.93% - 65.93% | 32.73% |

cleaned\_data %>%  
select(cardiovascular\_disease, where(is.factor)) %>%  
mutate(cholesterol = cholesterol %>% as.character() %>% replace\_na("Нет данных") %>%   
 as.factor()) %>%  
count(cardiovascular\_disease, cholesterol) %>%  
group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>%   
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

| **cardiovascular\_disease** | **cholesterol** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | above normal | 5750 | 16.44% | 16.05% - 16.83% | 8.21% |
| normal | 23055 | 65.91% | 65.41% - 66.41% | 32.94% |
| well above normal | 6174 | 17.65% | 17.25% - 18.05% | 8.82% |
| no | above normal | 3799 | 10.85% | 10.52% - 11.17% | 5.43% |
| normal | 29330 | 83.75% | 83.36% - 84.14% | 41.9% |
| well above normal | 1892 | 5.4% | 5.17% - 5.64% | 2.7% |

cleaned\_data %>%  
select(cardiovascular\_disease, where(is.factor)) %>%  
mutate(glucose = glucose %>% as.character() %>% replace\_na("Нет данных") %>%   
 as.factor()) %>%  
count(cardiovascular\_disease, glucose) %>%  
group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>% # Переименование колонки n  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

| **cardiovascular\_disease** | **glucose** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | above normal | 3078 | 8.8% | 8.50% - 9.10% | 4.4% |
| normal | 28585 | 81.72% | 81.32% - 82.13% | 40.84% |
| well above normal | 3316 | 9.48% | 9.17% - 9.79% | 4.74% |
| no | above normal | 2112 | 6.03% | 5.78% - 6.28% | 3.02% |
| normal | 30894 | 88.22% | 87.88% - 88.55% | 44.13% |
| well above normal | 2015 | 5.75% | 5.51% - 6.00% | 2.88% |

cleaned\_data %>%  
select(cardiovascular\_disease, where(is.factor)) %>%  
mutate(smoking = smoking %>% as.character() %>% replace\_na("Нет данных") %>%   
 as.factor()) %>%  
count(cardiovascular\_disease, smoking) %>%  
group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>% # Переименование колонки n  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

| **cardiovascular\_disease** | **smoking** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | no | 32050 | 91.63% | 91.34% - 91.92% | 45.79% |
| yes | 2929 | 8.37% | 8.08% - 8.66% | 4.18% |
| no | no | 31781 | 90.75% | 90.44% - 91.05% | 45.4% |
| yes | 3240 | 9.25% | 8.95% - 9.56% | 4.63% |

cleaned\_data %>%  
select(cardiovascular\_disease, where(is.factor)) %>%  
mutate(alcohol\_intake = alcohol\_intake %>% as.character() %>% replace\_na("Нет данных") %>%   
 as.factor()) %>%  
count(cardiovascular\_disease, alcohol\_intake) %>%  
group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>% # Переименование колонки n  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

| **cardiovascular\_disease** | **alcohol\_intake** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | no | 33156 | 94.79% | 94.56% - 95.02% | 47.37% |
| yes | 1823 | 5.21% | 4.98% - 5.44% | 2.6% |
| no | no | 33080 | 94.46% | 94.22% - 94.70% | 47.26% |
| yes | 1941 | 5.54% | 5.30% - 5.78% | 2.77% |

cleaned\_data %>%  
select(cardiovascular\_disease, where(is.factor)) %>%  
mutate(physical\_activity = physical\_activity %>% as.character() %>% replace\_na("Нет данных") %>%  
 as.factor()) %>%  
count(cardiovascular\_disease, physical\_activity) %>%  
group\_by(cardiovascular\_disease) %>%  
 mutate(  
 "процент по группе" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 "95% ДИ " = {  
 n\_group <- sum(n)   
 p <- n / n\_group  
 paste0(  
 scales::percent(p - 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01),  
 " - ",  
 scales::percent(p + 1.96 \* sqrt(p \* (1 - p) / n\_group), accuracy = 0.01)  
 )  
 }  
 ) %>%  
 ungroup() %>%  
 mutate(  
 "процент по выборке" = (n / sum(n)) %>% round(4) %>% '\*' (100) %>% str\_c("%"),  
 n = formatC(n, format = "d", big.mark = "")   
 )%>%  
 rename("абсолютное количество" = n) %>% # Переименование колонки n  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v("cardiovascular\_disease")

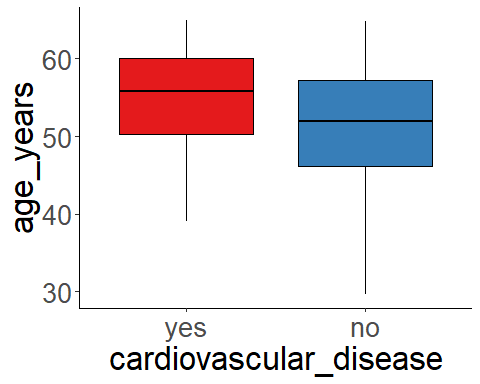
| **cardiovascular\_disease** | **physical\_activity** | **абсолютное количество** | **процент по группе** | **95% ДИ** | **процент по выборке** |
| --- | --- | --- | --- | --- | --- |
| yes | no | 7361 | 21.04% | 20.62% - 21.47% | 10.52% |
| yes | 27618 | 78.96% | 78.53% - 79.38% | 39.45% |
| no | no | 6378 | 18.21% | 17.81% - 18.62% | 9.11% |
| yes | 28643 | 81.79% | 81.38% - 82.19% | 40.92% |

# Визуализация

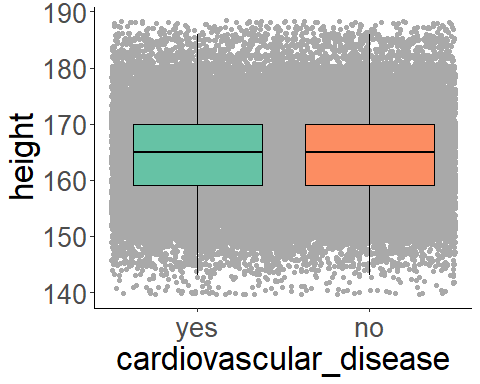
## Количественные переменные

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.

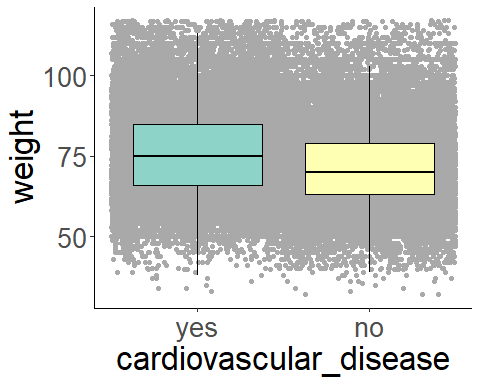
theme\_custom <- theme(  
 axis.text = element\_text(size = 20),  
 axis.title = element\_text(size = 25),  
 legend.position = "none"  
 )  
  
boxplot\_years <- ggplot(cleaned\_data)+  
 geom\_boxplot(aes(y = age\_years,   
 x = cardiovascular\_disease, fill = cardiovascular\_disease),   
 colour = "black", na.rm = TRUE)+  
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 labs(x = "cardiovascular\_disease",   
 y = "age\_years")+  
 theme\_classic()+  
 theme\_custom +  
 scale\_fill\_brewer(palette = "Set1")  
  
print(boxplot\_years)



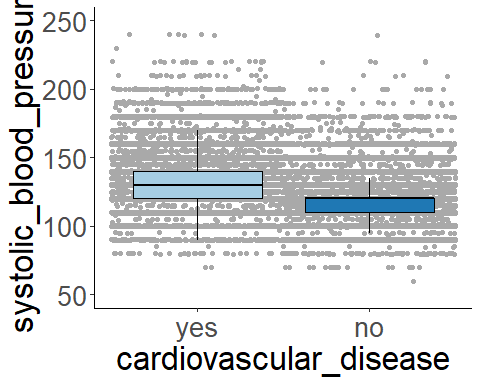
boxplot\_height <- ggplot(cleaned\_data) +  
 geom\_jitter(aes(y = height,   
 x = cardiovascular\_disease),   
 colour = "darkgrey",  
 width =0.5,   
 alpha = 3, na.rm = TRUE   
 )+  
 geom\_boxplot(aes(y = height,   
 x = cardiovascular\_disease,   
 fill = cardiovascular\_disease),   
 colour = "black",  
 outlier.shape = NA, na.rm = TRUE) +  
 scale\_x\_discrete(name = "cardiovascular\_disease") +  
 labs(x = "cardiovascular\_disease",   
 y = "height") +  
 theme\_classic() +  
 theme\_custom +  
 scale\_fill\_brewer(palette = "Set2")  
  
print(boxplot\_height)



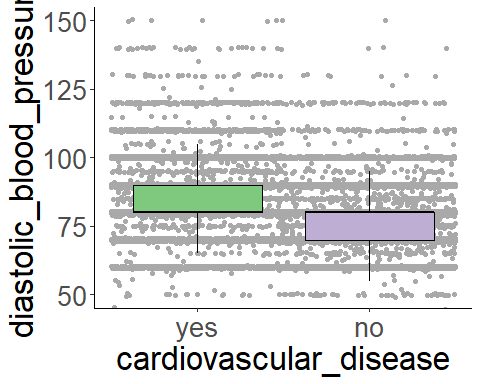
boxplot\_weight <- ggplot(cleaned\_data) +  
 geom\_jitter(aes(y = weight,   
 x = cardiovascular\_disease),   
 colour = "darkgrey",  
 width =0.5,   
 alpha = 3, na.rm = TRUE   
 )+  
 geom\_boxplot(aes(y = weight,   
 x = cardiovascular\_disease,   
 fill = cardiovascular\_disease),   
 colour = "black",  
 outlier.shape = NA, na.rm = TRUE) +  
 scale\_x\_discrete(name = "cardiovascular\_disease") +  
 labs(x = "cardiovascular\_disease",   
 y = "weight") +  
 theme\_classic() +  
 theme\_custom +  
 scale\_fill\_brewer(palette = "Set3")  
  
print(boxplot\_weight)



boxplot\_systolic\_blood\_pressure <- ggplot(cleaned\_data) +  
 geom\_jitter(aes(y = systolic\_blood\_pressure,   
 x = cardiovascular\_disease),   
 colour = "darkgrey",  
 width =0.5,   
 alpha = 3, na.rm = TRUE   
 )+  
 geom\_boxplot(aes(y = systolic\_blood\_pressure,   
 x = cardiovascular\_disease,   
 fill = cardiovascular\_disease),   
 colour = "black",  
 outlier.shape = NA, na.rm = TRUE) +  
 scale\_x\_discrete(name = "cardiovascular\_disease") +  
 labs(x = "cardiovascular\_disease",   
 y = "systolic\_blood\_pressure") +  
 theme\_classic() +  
 theme\_custom +  
 scale\_fill\_brewer(palette = "Paired")+  
 coord\_cartesian(ylim = c(50, 250)) #несмотря на то, что мы убрали выбросы из анализа,   
#некоторые нефизиологичные данные все же остались в датасете, и делают график нечитаемым  
#(он "сплющивается" на очень узком диапазоне), поэтому ось y ограничили в диапазоне чисел  
  
print(boxplot\_systolic\_blood\_pressure)



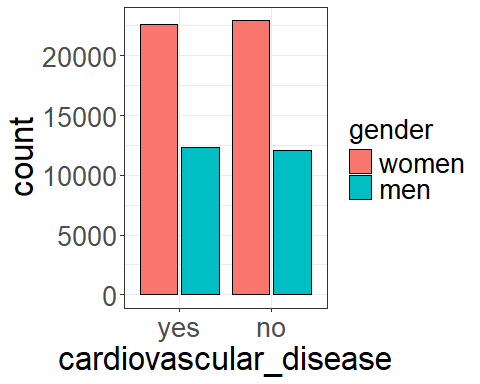
boxplot\_diastolic\_blood\_pressure <- ggplot(cleaned\_data) +  
 geom\_jitter(aes(y = diastolic\_blood\_pressure,   
 x = cardiovascular\_disease),   
 colour = "darkgrey",  
 width =0.5,   
 alpha = 3, na.rm = TRUE   
 )+  
 geom\_boxplot(aes(y = diastolic\_blood\_pressure,   
 x = cardiovascular\_disease,   
 fill = cardiovascular\_disease),   
 colour = "black",  
 outlier.shape = NA, na.rm = TRUE) +  
 scale\_x\_discrete(name = "cardiovascular\_disease") +  
 labs(x = "cardiovascular\_disease",   
 y = "diastolic\_blood\_pressure") +  
 theme\_classic() +  
 theme\_custom +  
 scale\_fill\_brewer(palette = "Accent")+  
 coord\_cartesian(ylim = c(50, 150)) #несмотря на то, что мы убрали выбросы из анализа,   
#некоторые нефизиологичные данные все же остались в датасете, и делают график нечитаемым  
#(он "сплющивается" на очень узком диапазоне), поэтому ось y ограничили в диапазоне чисел  
  
print(boxplot\_diastolic\_blood\_pressure)



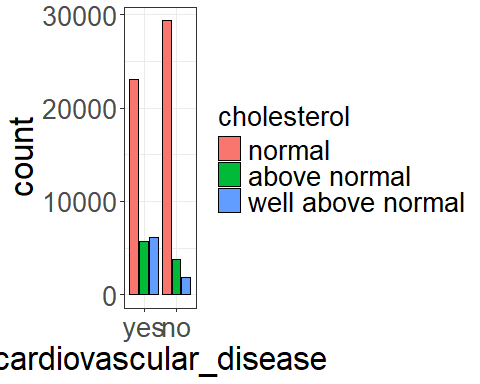
## Категориальные переменные

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

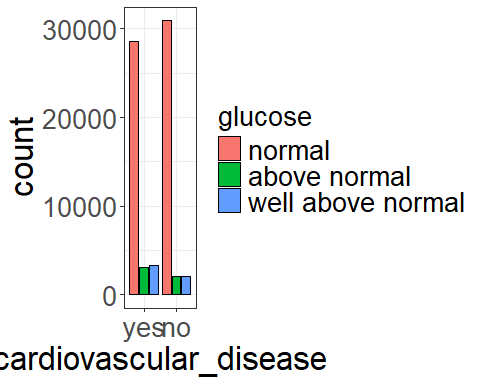
theme\_custom\_2 <- theme(  
 axis.text = element\_text(size = 20),  
 axis.title = element\_text(size = 25),  
 legend.title = element\_text(size = 20),  
 legend.text = element\_text(size = 20)  
 )  
  
geom\_bar\_gender <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = gender),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_gender)



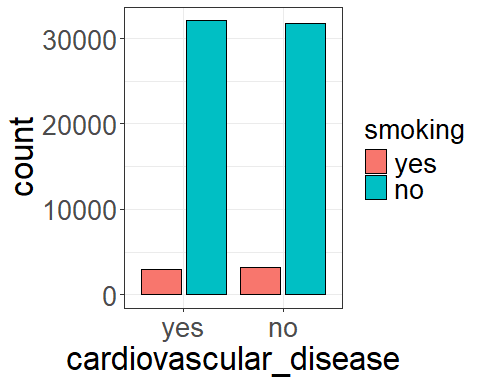
geom\_bar\_cholesterol <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = cholesterol),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_cholesterol)



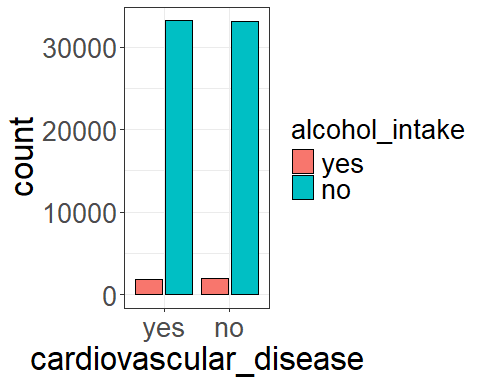
geom\_bar\_glucose <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = glucose),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_glucose)



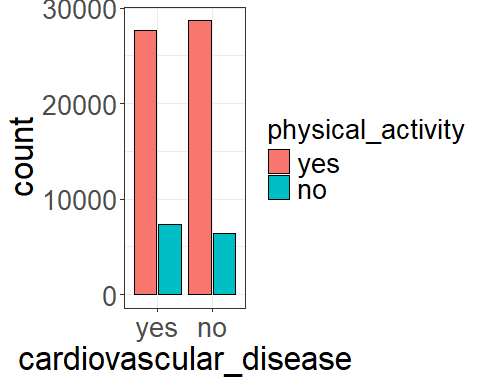
geom\_bar\_smoking <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = smoking),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_smoking)



geom\_bar\_alcohol\_intake <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = alcohol\_intake),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_alcohol\_intake)



geom\_bar\_physical\_activity <- ggplot(cleaned\_data)+  
 geom\_bar(aes(x = cardiovascular\_disease,   
 fill = physical\_activity),   
 colour = "black",   
 position = "dodge2")+   
 scale\_x\_discrete(name = "cardiovascular\_disease")+  
 theme\_bw()+  
 theme\_custom\_2  
  
print(geom\_bar\_physical\_activity)



#geom\_bar строит столбчатую диаграмму, где высота каждого столбца представляет количество наблюдений в каждой категории.  
  
#Преимущества:  
#Простой и интуитивно понятный способ визуализации категориальных данных.  
#Хорошо подходит для сравнения частот между категориями.  
#Можно использовать для визуализации соотношения одной категориальной переменной с другой, также  
#категориальной переменной

# 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?

shapiro\_test\_all <- function(data) {  
 # Отбираем количественные переменные  
 quant\_vars <- names(data)[sapply(data, is.numeric)]  
  
 # Проводим тест Шапиро-Уилка для каждой количественной переменной  
 shapiro\_results <- lapply(quant\_vars, function(var) {  
 # Отбираем 5000 значений из переменной  
 sample\_data <- sample(data[[var]], min(5000, length(data[[var]])), replace = TRUE)  
 result <- shapiro.test(sample\_data)  
 c(result$p.value, result$statistic)   
 })  
  
 # Преобразуем результаты в датафрейм  
 shapiro\_results <- data.frame(variable = quant\_vars,  
 p\_value = sapply(shapiro\_results, `[`, 1),  
 shapiro\_statistic = sapply(shapiro\_results, `[`, 2))  
  
 return(shapiro\_results)  
}  
  
shapiro\_results <- shapiro\_test\_all(cleaned\_data)  
shapiro\_results$p\_value <- round(shapiro\_results$p\_value, 10)  
shapiro\_results$shapiro\_statistic <- round(shapiro\_results$shapiro\_statistic, 5)  
  
 shapiro\_results%>%  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all")

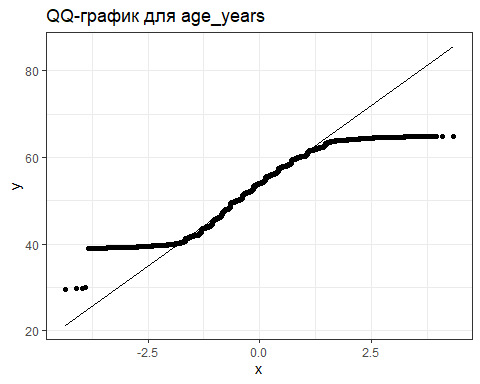
| **variable** | **p\_value** | **shapiro\_statistic** |
| --- | --- | --- |
| age\_days | 0 | 0.96594 |
| age\_years | 0 | 0.96313 |
| height | 0 | 0.99540 |
| weight | 0 | 0.97655 |
| systolic\_blood\_pressure | 0 | 0.89265 |
| diastolic\_blood\_pressure | 0 | 0.61474 |

#Тест Шапиро-Уилкачувствителен к отклонениям от нормального распределения. Если p-value <= 0.05: достаточно оснований для отклонения нулевой гипотезы о нормальном распределении, переменная вероятно не является нормально распределенной. Полученные #p-value намного меньше 0.05, поэтому с большой степенью вероятности можно сказать, что данные не соответствуют нормальному распределению.

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

quant\_vars <- c("age\_years", "height", "weight", "systolic\_blood\_pressure", "diastolic\_blood\_pressure")  
  
# Функция для построения QQ-плота  
  
plot\_qq <- function(var\_name) {  
 ggplot(cleaned\_data, aes(sample = .data[[var\_name]])) +   
 stat\_qq() +  
 stat\_qq\_line() +  
 labs(title = paste0("QQ-график для ", var\_name))+  
 theme\_bw()  
}  
  
# Строим QQ-плоты для каждой переменной  
lapply(quant\_vars, plot\_qq)

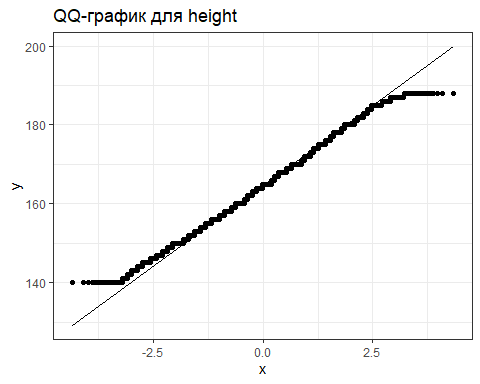
## [[1]]



##   
## [[2]]

## Warning: Removed 293 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

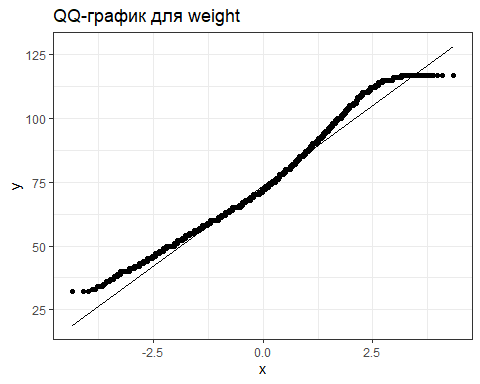
## Warning: Removed 293 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[3]]

## Warning: Removed 702 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

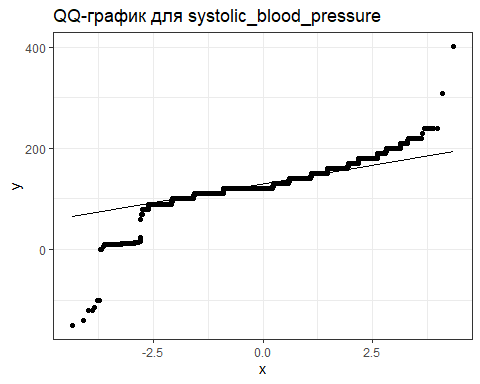
## Warning: Removed 702 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[4]]

## Warning: Removed 38 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

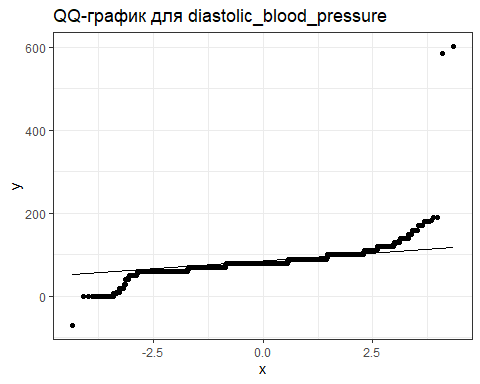
## Warning: Removed 38 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[5]]

## Warning: Removed 951 rows containing non-finite outside the scale range  
## (`stat\_qq()`).

## Warning: Removed 951 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



#В тесте Шапиро-Уилка мы получили p-value значительно меньше 0.05. Это указывало на то, что данные не соответствуют нормальному распределению.  
#QQ-плоты подтверждают результаты теста. На большинстве плотов точки отклоняются от прямой линии, что указывает на не нормальное распределение. Но видим, что у переменных height и в меньшей степени weight все не так плохо, и они уже более и менее приближены к нормальному распределению.  
#В идеале лучше использовать и тест Шапиро-Уилка, и QQ-плоты, чтобы получить более полное представление о распределении данных.  
#QQ-плоты особенно полезны для визуального анализа отклонений от нормальности.  
#Тест Шапиро-Уилка дает количественную оценку соответствия данных нормальному распределению.

1. Ниже напишите, какие ещё методы проверки на нормальность вы знаете и какие у них есть ограничения.

**Напишите текст здесь** 1. Тест Колмогорова-Смирнова (Kolmogorov-Smirnov test):

Принцип: Сравнивает эмпирическое распределение данных с теоретическим нормальным распределением. Ограничения: Более чувствителен к отклонениям в хвостах распределения, чем в центральной части. Может быть слишком строгим для малых выборок. Не подходит для проверки нормальности в случае наличия группирующих переменных (например, для разных групп пациентов).

1. Тест Андерсона-Дарлинга (Anderson-Darling test):

Принцип: Сравнивает эмпирическое распределение с теоретическим, при этом большее внимание уделяется хвостам распределения. Ограничения: Чувствителен к выбросам. Может быть слишком строгим для малых выборок. Не подходит для проверки нормальности в случае наличия группирующих переменных.

1. Критерий хи-квадрат (Chi-square test):

Принцип: Сравнивает эмпирические частоты с теоретическими частотами нормального распределения в разбитых интервалах. Ограничения: Требует большого размера выборки. Чувствителен к выбору интервалов. Не подходит для проверки нормальности в случае наличия группирующих переменных.

1. Графический анализ (Histograms, Boxplots):

Принцип: Визуальная оценка распределения данных с помощью гистограмм и боксплотов. Ограничения: Субъективный метод. Не дает количественной оценки соответствия данных нормальному распределению. Может быть трудно интерпретировать для малых выборок.

## Сравнение групп

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

#Пока не получилось автоматизировать эти тесты, поэтому каждый отдельно. Я понимаю, что  
#здесь тоже надо оформить нормальные таблицы, но еще не успела, буду рада продолжать  
#с этим возиться до второго дедлайна.  
  
# Сравнение групп по количественным переменным. Используем тест Манна-Уитни, так как   
#переменные распределены ненормально.  
  
# 1. age\_years  
test\_age\_years <- cleaned\_data %>%  
 wilcox\_test(age\_years ~ cardiovascular\_disease)  
print(test\_age\_years)

## # A tibble: 1 × 7  
## .y. group1 group2 n1 n2 statistic p  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 age\_years yes no 34979 35021 778302765 0

# 2. height  
test\_height <- cleaned\_data %>%  
 wilcox\_test(height ~ cardiovascular\_disease)  
print(test\_height)

## # A tibble: 1 × 7  
## .y. group1 group2 n1 n2 statistic p  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 height yes no 34834 34873 598267814. 0.000592

# 3. weight  
test\_weight <- cleaned\_data %>%  
 wilcox\_test(weight ~ cardiovascular\_disease)  
print(test\_weight)

## # A tibble: 1 × 7  
## .y. group1 group2 n1 n2 statistic p  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 weight yes no 34474 34824 724059970. 0

# 4. systolic\_blood\_pressure  
test\_systolic\_blood\_pressure <- cleaned\_data %>%  
 wilcox\_test(systolic\_blood\_pressure ~ cardiovascular\_disease)  
print(test\_systolic\_blood\_pressure)

## # A tibble: 1 × 7  
## .y. group1 group2 n1 n2 statistic p  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 systolic\_blood\_pressure yes no 34955 35007 920060298 0

# 5. diastolic\_blood\_pressure  
test\_diastolic\_blood\_pressure <- cleaned\_data %>%  
 wilcox\_test(diastolic\_blood\_pressure ~ cardiovascular\_disease)  
print(test\_diastolic\_blood\_pressure)

## # A tibble: 1 × 7  
## .y. group1 group2 n1 n2 statistic p  
## \* <chr> <chr> <chr> <int> <int> <dbl> <dbl>  
## 1 diastolic\_blood\_pressure yes no 34176 34873 822887754 0

#Все количественные переменные показали значимые различия в значениях в двух группах:  
# пациенты с сердечно-сосудистыми заболеваниями и без них.  
  
# Сравнение групп по категориальным переменным. Используем тест хи-квадрат, так как имеем дело с   
# 6. gender  
  
chisq\_result\_gender <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$gender, correct = FALSE)  
print(chisq\_result\_gender)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$gender  
## X-squared = 4.6035, df = 1, p-value = 0.03191

# 7. cholesterol  
  
chisq\_result\_cholesterol <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$cholesterol, correct = FALSE)  
print(chisq\_result\_cholesterol)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$cholesterol  
## X-squared = 3423.4, df = 2, p-value < 2.2e-16

# 8. glucose  
  
chisq\_result\_glucose <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$glucose, correct = FALSE)  
print(chisq\_result\_glucose)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$glucose  
## X-squared = 586.91, df = 2, p-value < 2.2e-16

# 9. smoking  
  
chisq\_result\_smoking <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$smoking, correct = FALSE)  
print(chisq\_result\_smoking)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$smoking  
## X-squared = 16.787, df = 1, p-value = 4.182e-05

# 10. alcohol\_intake  
  
chisq\_result\_alcohol\_intake <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$alcohol\_intake, correct = FALSE)  
print(chisq\_result\_alcohol\_intake)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$alcohol\_intake  
## X-squared = 3.7613, df = 1, p-value = 0.05245

# 11. physical\_activity  
  
chisq\_result\_physical\_activity <- chisq.test(cleaned\_data$cardiovascular\_disease, cleaned\_data$physical\_activity, correct = FALSE)  
print(chisq\_result\_physical\_activity)

##   
## Pearson's Chi-squared test  
##   
## data: cleaned\_data$cardiovascular\_disease and cleaned\_data$physical\_activity  
## X-squared = 88.981, df = 1, p-value < 2.2e-16

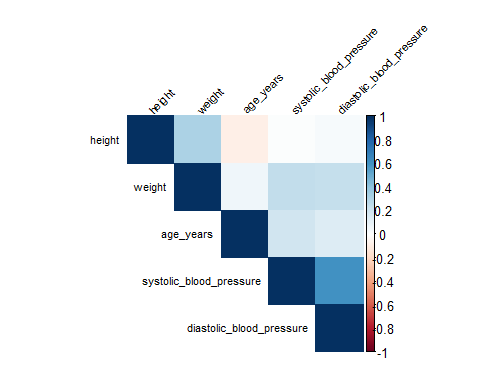
#Все категориальные переменные показали значимые взаимосвязи с переменной cardiovascular\_disease (только у употребления алкоголя  
#эта взаимосвязь на уровне тенденции, p.value чуть больше 0.05)

# 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.

# Выбираем количественные переменные  
quant\_vars <- c("age\_years", "height", "weight", "systolic\_blood\_pressure", "diastolic\_blood\_pressure")  
  
# Удаляем строки с NA в количественных переменных  
cleaned\_data\_no\_NA <- cleaned\_data[complete.cases(cleaned\_data[quant\_vars]), ]  
  
# Создаем корреляционную матрицу  
correlation\_matrix <- cor(cleaned\_data\_no\_NA[quant\_vars])  
  
# Визуализируем корреляционную матрицу с помощью corrplot  
corrplot(correlation\_matrix, method = "color", type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45, tl.cex = 0.7)



## Моделирование

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

#Так как cardiovascular\_disease — это бинарная переменная (да/нет), мы будем использовать логистическую регрессию, glm() с аргументом family = binomial. На первом этаме засовываем в нее все факторы, которые могут имент взаимосвязь с cardiovascular\_disease.  
  
# Удаляем строки с NA в количественных переменных  
cleaned\_data\_no\_NA <- cleaned\_data[complete.cases(cleaned\_data[quant\_vars]), ]  
  
mymodel <- glm(cardiovascular\_disease ~ age\_years + gender + smoking + weight + systolic\_blood\_pressure + cholesterol + glucose + height + diastolic\_blood\_pressure + alcohol\_intake + physical\_activity,   
 data = cleaned\_data\_no\_NA, family = binomial)  
  
#Выводим результаты модели  
summary(mymodel)

##   
## Call:  
## glm(formula = cardiovascular\_disease ~ age\_years + gender + smoking +   
## weight + systolic\_blood\_pressure + cholesterol + glucose +   
## height + diastolic\_blood\_pressure + alcohol\_intake + physical\_activity,   
## family = binomial, data = cleaned\_data\_no\_NA)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 10.9977197 0.2577155 42.674 < 2e-16 \*\*\*  
## age\_years -0.0525573 0.0013546 -38.799 < 2e-16 \*\*\*  
## gendermen 0.0102025 0.0222400 0.459 0.646418   
## smokingno -0.1430912 0.0348387 -4.107 4.00e-05 \*\*\*  
## weight -0.0126371 0.0007418 -17.035 < 2e-16 \*\*\*  
## systolic\_blood\_pressure -0.0463181 0.0008290 -55.873 < 2e-16 \*\*\*  
## cholesterolabove normal -0.3901076 0.0272924 -14.294 < 2e-16 \*\*\*  
## cholesterolwell above normal -1.0958890 0.0357842 -30.625 < 2e-16 \*\*\*  
## glucoseabove normal -0.0227801 0.0362190 -0.629 0.529379   
## glucosewell above normal 0.3294130 0.0395923 8.320 < 2e-16 \*\*\*  
## height 0.0053515 0.0013999 3.823 0.000132 \*\*\*  
## diastolic\_blood\_pressure -0.0219885 0.0012981 -16.939 < 2e-16 \*\*\*  
## alcohol\_intakeno -0.2005799 0.0423182 -4.740 2.14e-06 \*\*\*  
## physical\_activityno -0.2309616 0.0218775 -10.557 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 94352 on 68070 degrees of freedom  
## Residual deviance: 77167 on 68057 degrees of freedom  
## AIC: 77195  
##   
## Number of Fisher Scoring iterations: 4

#Я не успела как следует поработать с моделью. При беглом наброске со всеми переменными сразу результаты получились противоречивые. Надо тестировать модель, анализировать остатки, смотреть влияние выбросов, учесть в модели сочетание разных факторов.Вероятно, какие-то из них могут быть конфаундерами и коллайдерами. Также не всегда модель со всеми предикторами сразу - самая лучшая.  
#Надо убирать из модели те или иные факторы и подбирать модель с наиболее высоким значением R^2 adj. Надеюсь еще повозиться с этим до второго дедлайна.  
  
#Интерпретация первичных результатов:  
#age\_years: С каждым дополнительным годом возраста вероятность сердечно-сосудистого заболевания уменьшается на 0.0525573.Это, конечно, очень странный результат, надо разбираться, как он получился.  
#gendermen: Вероятность сердечно-сосудистого заболевания у мужчин выше на 0.0102025 по сравнению с женщинами, но эта разница не является статистически значимой (p-value > 0.05).  
#smokingno: У тех, кто не курит, вероятность сердечно-сосудистого заболевания ниже на 0.1430912 по сравнению с теми, кто курит (статистическая значимость p < 0.001).  
#weight: С каждыми дополнительными 1 кг веса вероятность сердечно-сосудистого заболевания уменьшается на 0.0126371 (статистическая значимость p < 0.001).Тоже странный результат.  
#systolic\_blood\_pressure: С каждым дополнительным мм рт. ст. систолического артериального давления вероятность сердечно-сосудистого заболевания снижается на 0.0463181 (статистическая значимость p < 0.001).??  
#cholesterolabove normal: При уровне холестерина выше нормы вероятность сердечно-сосудистого заболевания ниже на 0.3901076 по сравнению с нормальным уровнем (статистическая значимость p < 0.001). ??  
#cholesterolwell above normal: При уровне холестерина значительно выше нормы вероятность сердечно-сосудистого заболевания еще ниже на 1.0958890 по сравнению с нормальным уровнем (статистическая значимость p < 0.001). ??  
#glucoseabove normal: При уровне глюкозы выше нормы вероятность сердечно-сосудистого заболевания ниже на 0.0227801 по сравнению с нормальным уровнем, но эта разница не является статистически значимой (p-value > 0.05).  
#glucosewell above normal: При уровне глюкозы значительно выше нормы вероятность сердечно-сосудистого заболевания выше на 0.3294130 по сравнению с нормальным уровнем (статистическая значимость p < 0.001). Хоть один логичный результат!  
#height: С каждыми дополнительными 1 см роста вероятность сердечно-сосудистого заболевания повышается на 0.0053515 (статистическая значимость p < 0.001).  
#diastolic\_blood\_pressure: С каждым дополнительным мм рт. ст. диастолического артериального давления вероятность сердечно-сосудистого заболевания снижается на 0.0219885 (статистическая значимость p < 0.001). ??  
#alcohol\_intakeno: У тех, кто не употребляет алкоголь, вероятность сердечно-сосудистого заболевания ниже на 0.2005799 по сравнению с теми, кто употребляет алкоголь (статистическая значимость p < 0.001).  
#physical\_activityno: У тех, кто не занимается физической активностью, вероятность сердечно-сосудистого заболевания выше на 0.2309616 по сравнению с теми, кто занимается физической активностью (статистическая значимость p < 0.001).