automatization\_notebook

2024-10-27

# 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 (вместо 62.5 - 625)

# 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))

## [1] 0

#Пропущенных значений в датасете нет, но если бы они были:  
  
#cardio %>% select(where(~ mean(is.na(.x)) <= 0.2)) #Удаляем переменные с более 20% пропущенных значений  
   
#cardio %>% filter(rowSums(is.na(.)) / ncol(.) <= 0.2) #Удаляем строки, в которых более 20% значений пропущены

cardio %>%  
 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 можно сортировать по возрасту, но воспринимать намного понятнее в годах  
 rename(`Height (cm)` = height) %>%  
 rename(`Weight (kg)` = weight) %>%  
 rename(Gender = gender) %>%   
 rename(`Systolic blood pressure` = `ap\_hi`) %>%   
 rename(`Diastolic blood pressure` = `ap\_lo`) %>%  
 rename(Cholesterol = cholesterol ) %>%  
 rename(Glucose = gluc)%>%  
 rename(Smoking = smoke) %>%   
 rename(`Alcohol intake`= alco) %>%  
 rename(`Physical activity` = active) %>%  
 rename(`Cardiovascular disease` = cardio) %>%  
 relocate(`Cardiovascular disease`, .after = id) %>%   
   
 mutate(  
 across(Gender, function(x) x %>% factor(levels = c(1, 2), labels = c("Women", "Men"))),   
 across(c(Cholesterol, Glucose), function(x) x %>% factor(levels = c(1, 2, 3), labels = c("Normal", "Above normal", "Well above normal"))),  
 across(c(Smoking, `Alcohol intake`, `Physical activity`, `Cardiovascular disease`), function(x) x %>% factor(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 (cm)`, `Weight (kg)`, `Systolic blood pressure`, `Diastolic blood pressure`),  
 ~ ifelse(.x %in% find\_outliers(.x), .x, NA)  
 )  
 ) %>%   
 filter(!is.na(`Height (cm)`) | !is.na(`Weight (kg)`) |   
 !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, сильно 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 (cm)`, `Weight (kg)`, `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] 14

# Сколько осталось случаев?

nrow(cleaned\_data)

## [1] 70000

# Есть ли в данных идентичные строки?

any(duplicated(cleaned\_data))

## [1] FALSE

anyDuplicated(cleaned\_data)

## [1] 0

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

NA\_vars <- names(cleaned\_data)[sapply(cleaned\_data, function(x) any(is.na(x)))]   
sapply(cleaned\_data[NA\_vars], function(x) sum(is.na(x)))%>%  
enframe() %>%  
 rename(`NA count` = value) %>%  
 rename(`Variable` = name)

## # A tibble: 4 × 2  
## Variable `NA count`  
## <chr> <int>  
## 1 Height (cm) 293  
## 2 Weight (kg) 702  
## 3 Systolic blood pressure 38  
## 4 Diastolic blood pressure 951

# 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) & !id, -`Age (days)`) %>%   
 group\_by(`Cardiovascular disease`) %>%   
 summarize(across(where(is.numeric), statistics)) %>%  
 pivot\_longer(!`Cardiovascular disease`) %>%  
 separate(name, into = c("Variable", "Statistics"), sep = "\_\_") %>%   
 rename(Value = value) %>%  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%   
 merge\_v(c("Cardiovascular disease", "Variable"))

| **Cardiovascular disease** | **Variable** | **Statistics** | **Value** |
| --- | --- | --- | --- |
| Yes | 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 (cm) | 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 (kg) | 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 (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 (cm) | 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 (kg) | 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.

#я не смогла завставить тест Фишера работать на таком большом датасете, использую Хи-квадрат  
   
cleaned\_data %>%  
 select(where(is.factor)) -> factor\_data  
  
get\_cat\_table(factor\_data, "Cardiovascular disease") %>%  
 rename(`Cardiovascular disease` = Group) %>%  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all") %>%  
 merge\_v(c("Cardiovascular disease", "Variable")) %>%  
 merge\_v("Variable", target = "Chi-square test, p-value")

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(group\_variable)  
##   
## # Now:  
## data %>% select(all\_of(group\_variable))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.  
## ℹ Please use `all\_of()` or `any\_of()` instead.  
## # Was:  
## data %>% select(variable\_name)  
##   
## # Now:  
## data %>% select(all\_of(variable\_name))  
##   
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

| **Variable** | **Cardiovascular disease** | **Value** | **Absolute frequency** | **Relative frequency** | **95% CI** | **Relative frequency by group** | **Chi-square test, p-value** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Gender | Yes | Women | 22,616 | 32.31% | 31.96% - 32.66% | 64.66% | 0.033 |
| Men | 12,363 | 17.66% | 17.38% - 17.94% | 35.34% |
| No | Women | 22,914 | 32.73% | 32.39% - 33.08% | 65.43% |
| Men | 12,107 | 17.3% | 17.02% - 17.58% | 34.57% |
| Cholesterol | Yes | Normal | 23,055 | 32.94% | 32.59% - 33.28% | 65.91% | 0.000 |
| Above normal | 5,750 | 8.21% | 8.01% - 8.42% | 16.44% |
| Well above normal | 6,174 | 8.82% | 8.61% - 9.03% | 17.65% |
| No | Normal | 29,330 | 41.9% | 41.53% - 42.27% | 83.75% |
| Above normal | 3,799 | 5.43% | 5.26% - 5.59% | 10.85% |
| Well above normal | 1,892 | 2.7% | 2.58% - 2.82% | 5.4% |
| Glucose | Yes | Normal | 28,585 | 40.84% | 40.47% - 41.20% | 81.72% | 0.000 |
| Above normal | 3,078 | 4.4% | 4.25% - 4.55% | 8.8% |
| Well above normal | 3,316 | 4.74% | 4.58% - 4.89% | 9.48% |
| No | Normal | 30,894 | 44.13% | 43.77% - 44.50% | 88.22% |
| Above normal | 2,112 | 3.02% | 2.89% - 3.14% | 6.03% |
| Well above normal | 2,015 | 2.88% | 2.75% - 3.00% | 5.75% |
| Smoking | Yes | Yes | 2,929 | 4.18% | 4.04% - 4.33% | 8.37% | 0.000 |
| No | 32,050 | 45.79% | 45.42% - 46.15% | 91.63% |
| No | Yes | 3,240 | 4.63% | 4.47% - 4.78% | 9.25% |
| No | 31,781 | 45.4% | 45.03% - 45.77% | 90.75% |
| Alcohol intake | Yes | Yes | 1,823 | 2.6% | 2.49% - 2.72% | 5.21% | 0.055 |
| No | 33,156 | 47.37% | 47.00% - 47.74% | 94.79% |
| No | Yes | 1,941 | 2.77% | 2.65% - 2.89% | 5.54% |
| No | 33,080 | 47.26% | 46.89% - 47.63% | 94.46% |
| Physical activity | Yes | Yes | 27,618 | 39.45% | 39.09% - 39.82% | 78.96% | 0.000 |
| No | 7,361 | 10.52% | 10.29% - 10.74% | 21.04% |
| No | Yes | 28,643 | 40.92% | 40.55% - 41.28% | 81.79% |
| No | 6,378 | 9.11% | 8.90% - 9.32% | 18.21% |

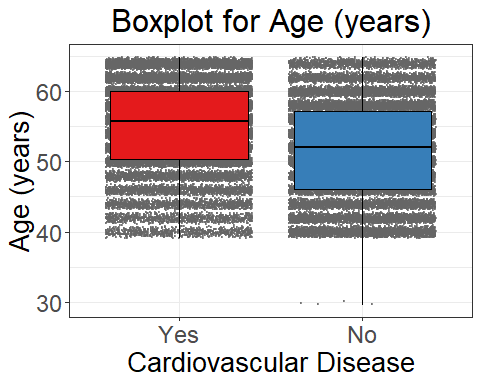
# Визуализация

## Количественные переменные

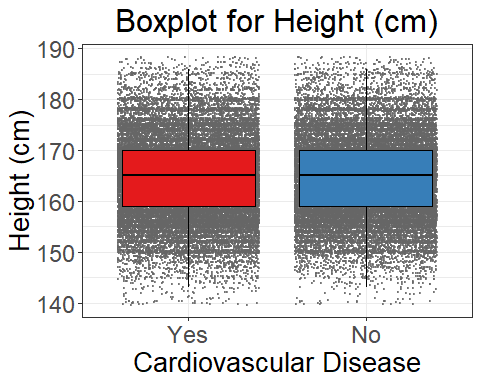
1. Для каждой количественной переменной сделайте боксплоты по группам. Расположите их либо на отдельных рисунках, либо на одном, но читаемо;
2. Наложите на боксплоты beeplots - задание со звёздочкой.
3. Раскрасьте боксплоты с помощью библиотеки RColorBrewer.

theme\_custom <- theme(  
 axis.text = element\_text(size = 18),  
 axis.title = element\_text(size = 20),  
 plot.title = element\_text(size = 24, hjust = 0.5),  
 legend.position = "none"  
 )  
  
quant\_vars <- names(cleaned\_data)[sapply(cleaned\_data, is.numeric) &   
 !names(cleaned\_data) %in% c("id", "Age (days)")]  
  
  
create\_boxplot <- function(var\_name) {  
 ggplot(cleaned\_data, aes(x = `Cardiovascular disease`, y = !!sym(var\_name), fill = `Cardiovascular disease`)) +  
 geom\_jitter(  
 colour = "gray40",  
 size = 0.5,  
 width =0.4,   
 alpha = 0.7, na.rm = TRUE   
 ) +  
 geom\_boxplot(colour = "black", fatten = 2, na.rm = TRUE, outlier.shape = NA) +  
 scale\_fill\_brewer(palette = "Set1") +  
 labs(title = paste0("Boxplot for ", var\_name),  
 x = "Cardiovascular Disease",  
 y = var\_name) +  
 theme\_bw()+  
 theme\_custom  
}  
  
lapply(quant\_vars, create\_boxplot)

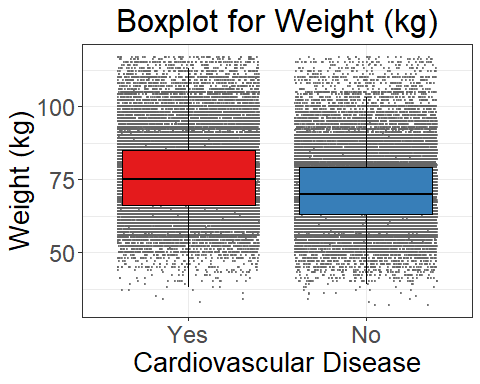
## [[1]]



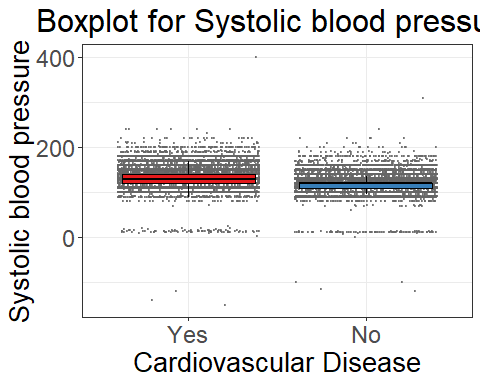
##   
## [[2]]



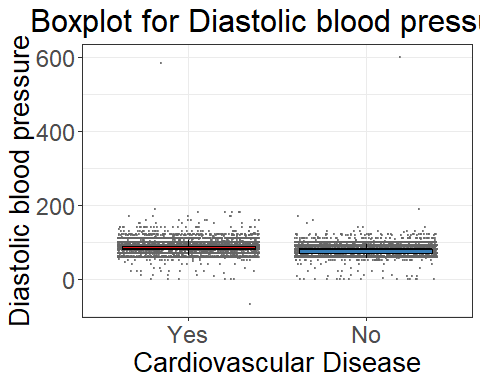
##   
## [[3]]



##   
## [[4]]



##   
## [[5]]



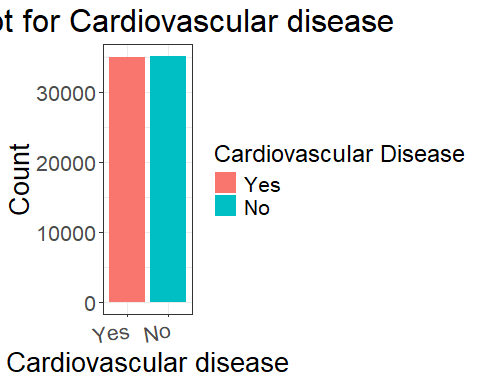
Несмотря на то, что из датасета были удалены выбросы за передлами трех сигм, все равно остались значения, вряд ли адекватные и, вероятно, связанные с ошибкой ввода (особенно это видно по показателям давления).

## Категориальные переменные

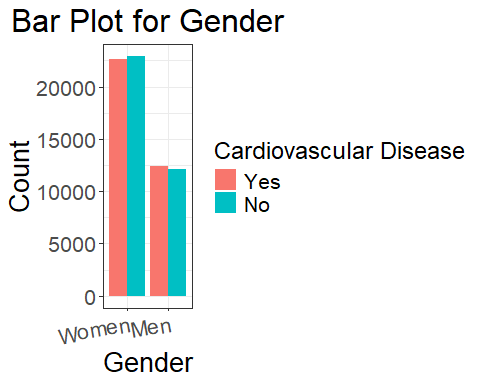
1. Сделайте подходящие визуализации категориальных переменных. Обоснуйте, почему выбрали именно этот тип.

theme\_custom\_2 <- theme(  
 axis.text = element\_text(size = 16),  
 axis.text.x = element\_text(angle = 10, hjust = 1),  
 axis.title = element\_text(size = 20),  
 plot.title = element\_text(size = 24, hjust = 0.5),  
 legend.title = element\_text(size = 18),  
 legend.text = element\_text(size = 16)  
 )  
  
  
cat\_vars <- names(cleaned\_data)[sapply(cleaned\_data, is.factor)]  
  
create\_bar\_plot <- function(var\_name) {  
 ggplot(cleaned\_data, aes(x = !!sym(var\_name), fill = `Cardiovascular disease`)) +  
 geom\_bar(position = "dodge") +   
 labs(title = paste0("Bar Plot for ", var\_name),  
 x = var\_name,  
 y = "Count",  
 fill = "Cardiovascular Disease") +  
 theme\_bw()+  
 theme\_custom\_2  
}  
  
lapply(cat\_vars, create\_bar\_plot)

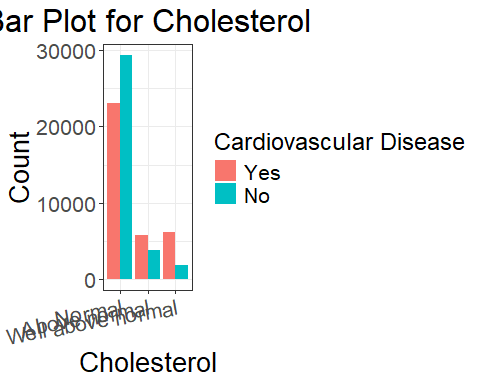
## [[1]]



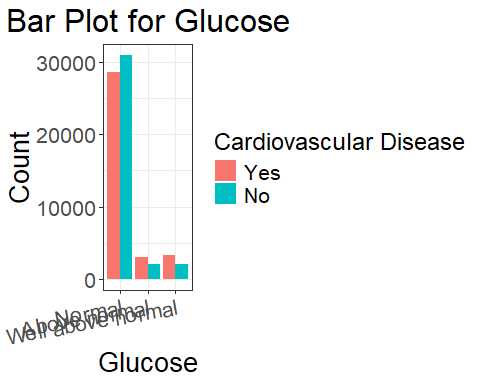
##   
## [[2]]



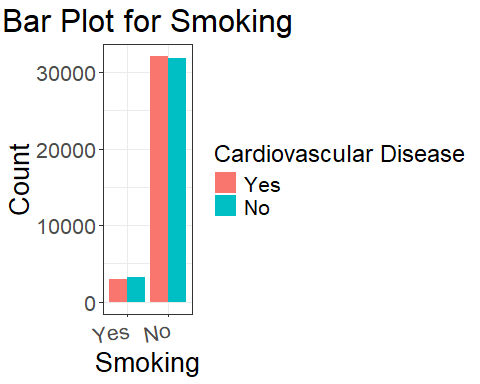
##   
## [[3]]



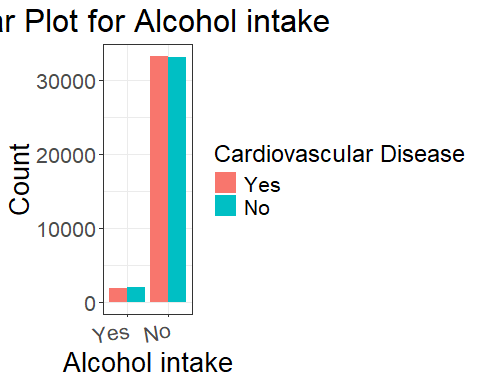
##   
## [[4]]



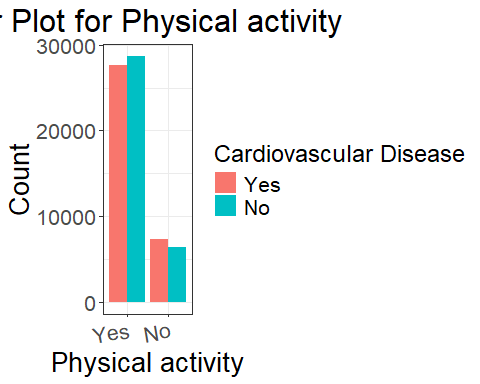
##   
## [[5]]



##   
## [[6]]



##   
## [[7]]



Geom\_bar строит столбчатую диаграмму, где высота каждого столбца представляет количество наблюдений в каждой категории.

Преимущества такого способа визуализации категориальной переменной: 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?

#Первый вариант кода, который сделан до проведенной консультации (решила оставить оба)  
  
shapiro\_test\_all <- function(data) {  
 # Отбираем количественные переменные  
 quant\_vars <- names(data)[sapply(data, is.numeric) & names(data) != "id" & names(data) != "Age (days)"]  
  
 # 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, 4)  
shapiro\_results$Shapiro\_statistic <- round(shapiro\_results$Shapiro\_statistic, 4)  
  
 shapiro\_results%>%  
 flextable() %>%  
 theme\_box() %>%  
 align(align = "center", part = "all")

| **Variable** | **p\_value** | **Shapiro\_statistic** |
| --- | --- | --- |
| Age (years) | 0 | 0.9649 |
| Height (cm) | 0 | 0.9955 |
| Weight (kg) | 0 | 0.9811 |
| Systolic blood pressure | 0 | 0.8883 |
| Diastolic blood pressure | 0 | 0.8705 |

Тест Шапиро-Уилка чувствителен к отклонениям от нормального распределения. Если p-value <= 0.05, то 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.

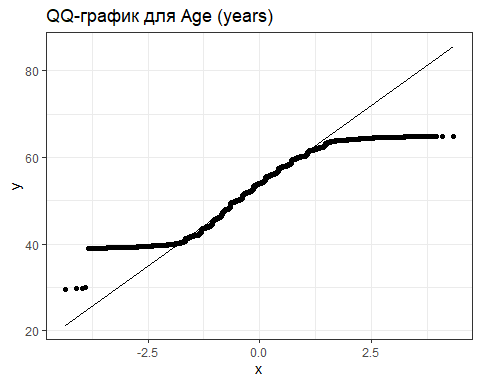
# Второй вариант кода, сделан после проведенной консультации  
  
cleaned\_data %>%  
 select(where(is.numeric), -`Age (days)`, -id) %>%   
   
 sapply(function(x) {  
 sample\_x <- sample(x, min(5000, length(x)))   
 shapiro.test(sample\_x)$p.value < 0.05   
 }) %>%  
   
 enframe() %>%  
   
 mutate(across(value, function(x) ifelse(x == TRUE, "Распределение отлично от нормального", "Распределение нормальное"))) %>%  
 rename(Variable = name) %>%  
 rename(Distribution = value)

## # A tibble: 5 × 2  
## Variable Distribution   
## <chr> <chr>   
## 1 Age (years) Распределение отлично от нормального  
## 2 Height (cm) Распределение отлично от нормального  
## 3 Weight (kg) Распределение отлично от нормального  
## 4 Systolic blood pressure Распределение отлично от нормального  
## 5 Diastolic blood pressure Распределение отлично от нормального

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 (cm)", "Weight (kg)", "Systolic blood pressure", "Diastolic blood pressure")  
  
# Функция для построения QQ-плота  
  
plot\_qq <- function(var\_name) {  
 ggplot(cleaned\_data, aes(sample = .data[[var\_name]]), na.rm = TRUE) +   
 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)

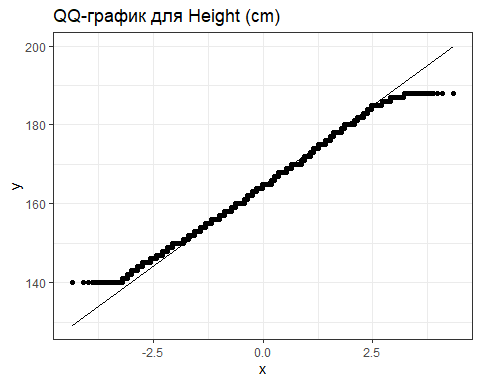
## [[1]]



##   
## [[2]]

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

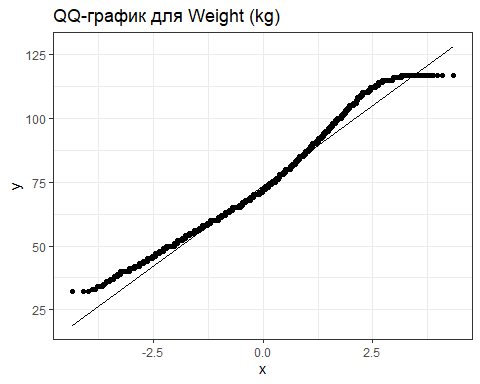
## Warning: Removed 293 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[3]]

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

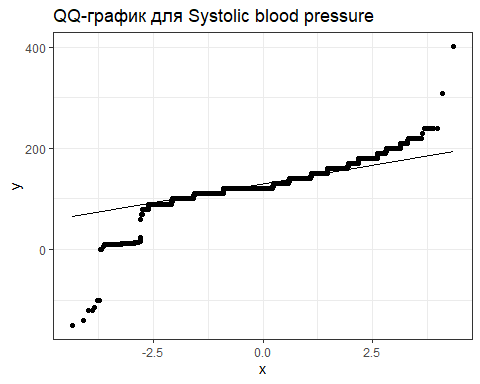
## Warning: Removed 702 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[4]]

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

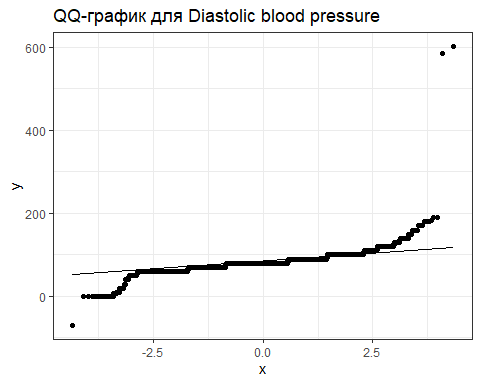
## Warning: Removed 38 rows containing non-finite outside the scale range  
## (`stat\_qq\_line()`).



##   
## [[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 больше 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.

# Сравнение групп по категориальным переменным.   
  
#Не могу пока преодолеть привычку использовать непараметрические тесты, когда данные распределены ненормально, поэтому сначала тест Манна-Уитни  
  
cleaned\_data %>%  
 select(where(is.numeric), -`Age (days)`, -id) %>%   
   
 names() %>%  
 set\_names() %>%  
 map(function(x) wilcox.test(cleaned\_data[[x]] ~ cleaned\_data$"Cardiovascular disease")$p.value < 0.05) %>%  
 enframe() %>%  
 unnest() %>%  
   
 mutate(across(value, function(x) ifelse(value == TRUE, "Различие между группами есть", "Различие между группами не доказано"))) %>%  
   
 filter(value == "Различие между группами есть") %>%  
 rename(`Quantitative variable` = name) %>%  
 rename(`Wilcox test result` = value)

## Warning: `cols` is now required when using `unnest()`.  
## ℹ Please use `cols = c(value)`.

## # A tibble: 5 × 2  
## `Quantitative variable` `Wilcox test result`   
## <chr> <chr>   
## 1 Age (years) Различие между группами есть  
## 2 Height (cm) Различие между группами есть  
## 3 Weight (kg) Различие между группами есть  
## 4 Systolic blood pressure Различие между группами есть  
## 5 Diastolic blood pressure Различие между группами есть

#Тем не менее, на таких данных можно использовать и параметрический t.test, поэтому сделала и его тоже. Результат не отличается, но тест более точный.  
  
cleaned\_data %>%  
 select(where(is.numeric), -`Age (days)`, -id) %>%   
   
 names() %>%  
 set\_names() %>%  
 map(function(x) t.test(cleaned\_data[[x]] ~ cleaned\_data$"Cardiovascular disease")$p.value < 0.05) %>%  
 enframe() %>%  
 unnest() %>%  
   
 mutate(across(value, function(x) ifelse(value == TRUE, "Различие между группами есть", "Различие между группами не доказано"))) %>%  
   
 filter(value == "Различие между группами есть") %>%  
 rename(`Quantitative variable` = name) %>%  
 rename(`T-test result` = value)

## Warning: `cols` is now required when using `unnest()`.  
## ℹ Please use `cols = c(value)`.

## # A tibble: 5 × 2  
## `Quantitative variable` `T-test result`   
## <chr> <chr>   
## 1 Age (years) Различие между группами есть  
## 2 Height (cm) Различие между группами есть  
## 3 Weight (kg) Различие между группами есть  
## 4 Systolic blood pressure Различие между группами есть  
## 5 Diastolic blood pressure Различие между группами есть

# Сравнение групп по категориальным переменным. Тест Фишера не тянет такой большой датасет, поэтому тест Хи-квадрат.   
  
cleaned\_data %>%  
 select(where(is.factor), -`Cardiovascular disease`) %>%   
   
 names() %>%  
 set\_names() %>%  
 map(function(x) {  
 table\_data <- table(cleaned\_data[[x]], cleaned\_data$"Cardiovascular disease") # Создаем таблицу соответствий  
 chisq.test(table\_data)$p.value < 0.05   
 }) %>%  
 enframe() %>%  
 unnest(cols = c(value)) %>%  
   
 mutate(across(value, function(x) ifelse(x == TRUE, "Различие между группами есть", "Различие между группами не доказано"))) %>%  
   
 filter(value == "Различие между группами есть") %>%  
 rename(`Categorical variable` = name) %>%  
 rename(`Сhi-square test result` = value)

## # A tibble: 5 × 2  
## `Categorical variable` `Сhi-square test result`   
## <chr> <chr>   
## 1 Gender Различие между группами есть  
## 2 Cholesterol Различие между группами есть  
## 3 Glucose Различие между группами есть  
## 4 Smoking Различие между группами есть  
## 5 Physical activity Различие между группами есть

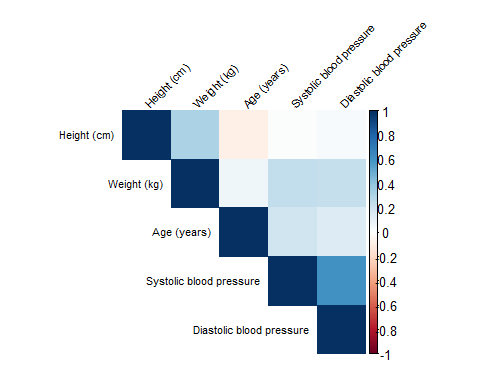
#Значимые различия обнаружены по всем переменным, кроме Alcohol intake

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

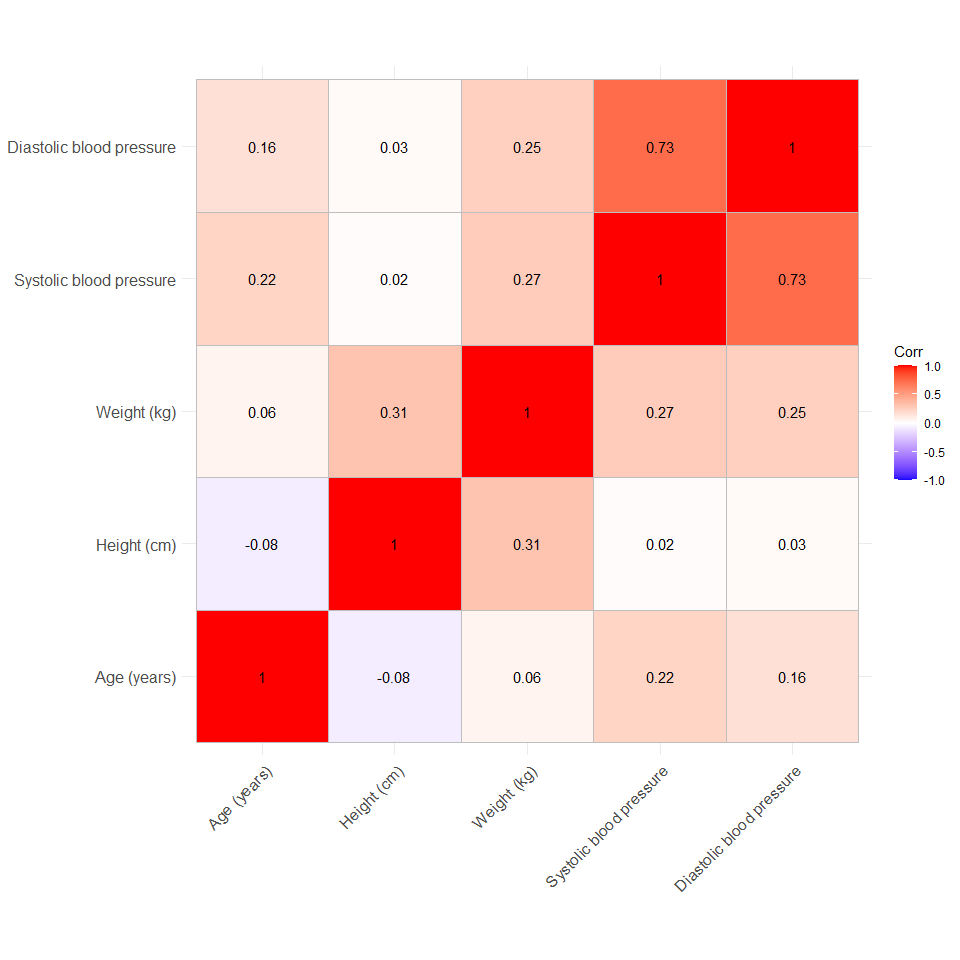
## Корреляционный анализ

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.

#Первый вариант, сделан до проведенной консультации.  
  
quant\_vars <- c("Age (years)", "Height (cm)", "Weight (kg)", "Systolic blood pressure", "Diastolic blood pressure")  
  
cleaned\_data\_no\_NA <- cleaned\_data[complete.cases(cleaned\_data[quant\_vars]), ]  
  
correlation\_matrix <- cor(cleaned\_data\_no\_NA[quant\_vars])  
  
corrplot(correlation\_matrix, method = "color", type = "upper", order = "hclust",   
 tl.col = "black", tl.srt = 45, tl.cex = 0.7)



#Второй вариант, сделан после консультации  
  
cleaned\_data %>%  
 select(where(is.numeric), -`Age (days)`, -id) %>%  
 corr.test(method = "spearman") -> corr\_data  
  
ggcorrplot(corr\_data$r, p.mat = corr\_data$p, insig = "blank", lab = TRUE)



## 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.  
  
  
cleaned\_data %>%   
 select(!id, -`Age (days)`) %>%  
 mutate(across(where(is.factor), function(x) x %>% fct\_relabel(function(x) x %>% str\_c(": ", .)))) -> model\_data  
   
glm(`Cardiovascular disease` ~ ., model\_data, family = binomial) %>%  
   
 tidy(conf.int = TRUE) %>%  
   
 mutate(across(c(estimate, std.error, statistic, conf.low, conf.high), function(x) x %>% formatC(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`) %>%  
   
 flextable() %>%  
 theme\_box() %>%  
 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")

| **Переменная** | **Коэффициент** | **95% CI** | **Стд.ошибка** | **Статистика** | **p.value** |
| --- | --- | --- | --- | --- | --- |
| Intercept | 11.00 | 10.49, 11.50 | 0.26 | 42.67 | **<0.001** |
| Age (years) | -0.05 | -0.06, -0.05 | 0.00 | -38.80 | **<0.001** |
| Gender: Men | 0.01 | -0.03, 0.05 | 0.02 | 0.46 | 0.646 |
| Height (cm) | 0.01 | 0.00, 0.01 | 0.00 | 3.82 | **<0.001** |
| Weight (kg) | -0.01 | -0.01, -0.01 | 0.00 | -17.04 | **<0.001** |
| Systolic blood pressure | -0.05 | -0.05, -0.04 | 0.00 | -55.87 | **<0.001** |
| Diastolic blood pressure | -0.02 | -0.02, -0.02 | 0.00 | -16.94 | **<0.001** |
| Cholesterol: Above normal | -0.39 | -0.44, -0.34 | 0.03 | -14.29 | **<0.001** |
| Cholesterol: Well above normal | -1.10 | -1.17, -1.03 | 0.04 | -30.62 | **<0.001** |
| Glucose: Above normal | -0.02 | -0.09, 0.05 | 0.04 | -0.63 | 0.529 |
| Glucose: Well above normal | 0.33 | 0.25, 0.41 | 0.04 | 8.32 | **<0.001** |
| Smoking: No | -0.14 | -0.21, -0.07 | 0.03 | -4.11 | **<0.001** |
| Alcohol intake: No | -0.20 | -0.28, -0.12 | 0.04 | -4.74 | **<0.001** |
| Physical activity: No | -0.23 | -0.27, -0.19 | 0.02 | -10.56 | **<0.001** |