automatization\_notebook

Anna Andreychenko

2024-10-27

# Чтение данных

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

read.csv2("data/raw/cardio\_train\_big.csv", dec=".") -> original\_data  
original\_data |> head()

## id age gender height weight ap\_hi ap\_lo cholesterol gluc smoke alco active  
## 1 0 18393 2 168 62 110 80 1 1 0 0 1  
## 2 1 20228 1 156 85 140 90 3 1 0 0 1  
## 3 2 18857 1 165 64 130 70 3 1 0 0 0  
## 4 3 17623 2 169 82 150 100 1 1 0 0 1  
## 5 4 17474 1 156 56 100 60 1 1 0 0 0  
## 6 8 21914 1 151 67 120 80 2 2 0 0 0  
## cardio  
## 1 0  
## 2 1  
## 3 1  
## 4 1  
## 5 0  
## 6 0

# Выведите общее описание данных

glimpse(original\_data)

## Rows: 70,000  
## Columns: 13  
## $ id <int> 0, 1, 2, 3, 4, 8, 9, 12, 13, 14, 15, 16, 18, 21, 23, 24, 2…  
## $ age <int> 18393, 20228, 18857, 17623, 17474, 21914, 22113, 22584, 17…  
## $ gender <int> 2, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 2, 2, 1, 2, 2, 1, 1, 1, 2…  
## $ height <int> 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 <int> 110, 140, 130, 150, 100, 120, 130, 130, 110, 110, 120, 120…  
## $ ap\_lo <int> 80, 90, 70, 100, 60, 80, 80, 90, 70, 60, 80, 80, 80, 70, 9…  
## $ cholesterol <int> 1, 3, 3, 1, 1, 2, 3, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
## $ gluc <int> 1, 1, 1, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 1…  
## $ smoke <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1…  
## $ alco <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0…  
## $ active <int> 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1…  
## $ cardio <int> 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**: Выбрала оба варианта: 1. Оставляем записи с не менее 50% переменных. 2. Осталяем переменные заполненные в более 20% записей

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”.

original\_data[which(rowMeans(!is.na(original\_data)) > 0.5), which(colMeans(!is.na(original\_data)) > 0.2)] %>%  
 rename\_with(function(x) x %>%   
 str\_replace("ap\_hi", "systolic\_blood\_pressure") %>%   
 str\_replace("ap\_lo", "diastolic\_blood\_pressure") %>%   
 str\_replace("gluc", "glucose") %>%  
 str\_replace("active", "physical\_activity")) %>%  
 mutate(across(c(gender,smoke,alco,physical\_activity,cardio), function(x) as.factor(x))) %>%  
 mutate(across(!c(id, gender,smoke,alco,physical\_activity,cardio), function(x) as.numeric(x))) %>%  
 arrange(desc(age)) -> cleaned\_data  
  
outliers1 <- outliers(cleaned\_data)  
temp <- list()  
for (col in names(outliers1)) {  
 outlier <- outliers1[[col]]  
 if (length(outlier) > 0) {  
 temp <- cleaned\_data[which(cleaned\_data[[col]] %in% outlier),]  
 }   
}  
temp <- temp[!duplicated(temp$id),]  
write.csv2(temp, "data/outliers.csv")  
rm(temp, outliers1, col, outlier)

# Сколько осталось переменных?

cleaned\_data |> ncol()

## [1] 13

# Сколько осталось случаев?

cleaned\_data |> nrow()

## [1] 70000

# Есть ли в данных идентичные строки?

print("Кол-во идентичных строк c учетом id")

## [1] "Кол-во идентичных строк c учетом id"

sum(as.numeric(duplicated(cleaned\_data)))

## [1] 0

print("Кол-во идентичных строк без учета id")

## [1] "Кол-во идентичных строк без учета id"

sum(as.numeric(duplicated(cleaned\_data[,-1])))

## [1] 24

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

#  
#cleaned\_data[2:5,3:4] <- NA  
print("Кол-во переменных с пропущенными значениями")

## [1] "Кол-во переменных с пропущенными значениями"

print(length(names(cleaned\_data)[sapply(cleaned\_data, anyNA)]))

## [1] 0

print("Кол-во пропущенных значений в каждой такой переменной")

## [1] "Кол-во пропущенных значений в каждой такой переменной"

colSums(is.na(cleaned\_data[sapply(cleaned\_data, anyNA)]))

## numeric(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 значений` = ~length(.x) %>% as.character(),  
 `\_Количество пропущенных значений` = ~sum(is.na(.x)) %>% as.character(),  
 `\_Среднее` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", mean(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_Медиана` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", median(.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()),  
 `\_25% квантиль и 75% квантиль` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*", paste0(quantile(.x, 0.25, na.rm = TRUE) %>% round(2), " и ", quantile(.x, 0.75, na.rm = TRUE) %>% round(2))),  
 `\_Интерквартильный размах` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*",  
IQR(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_Минимум` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*",  
 min(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_Максимум` = ~ifelse(sum(!is.na(.x)) == 0, "Н/П\*",  
 max(.x, na.rm = TRUE) %>% round(2) %>% as.character()),  
 `\_95% ДИ для среднего` = ~ifelse(sum(!is.na(.x)) < 3, "Н/П\*", paste0((mean(.x, na.rm = TRUE) - qt(0.975, df = length(.x) - 1)\*sd(.x, na.rm = TRUE)/sqrt(length(.x))) %>% round(2), " - ", (mean(.x, na.rm = TRUE) + qt(0.975, df = length(.x) - 1)\*sd(.x, na.rm = TRUE)/sqrt(length(.x))) %>% round(2)))  
  
   
)  
  
cleaned\_data %>%  
 select(cardio, where(is.numeric)) %>%  
 group\_by(cardio) %>%  
 summarize(across(where(is.numeric) & !c(id), statistics)) %>%  
 pivot\_longer(!cardio) %>%  
 separate(name, into = c("Variable", "Statistics"), sep = "\_\_") %>%  
 mutate(cardio = factor(cardio, levels=c(0,1), labels=c("Нет", "Да"))) %>%  
 rename (Value = value, "Наличие ССЗ" = cardio) %>%  
 flextable() %>% merge\_v(, j=c("Наличие ССЗ", "Variable")) %>%  
 autofit()

| Наличие ССЗ | Variable | Statistics | Value |
| --- | --- | --- | --- |
| Нет | age | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 18881.62 |
| Медиана | 19005 |
| Станд. отклон. | 2473.93 |
| 25% квантиль и 75% квантиль | 16838 и 20886 |
| Интерквартильный размах | 4048 |
| Минимум | 10798 |
| Максимум | 23678 |
| 95% ДИ для среднего | 18855.71 - 18907.53 |
| height | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 164.45 |
| Медиана | 165 |
| Станд. отклон. | 8.15 |
| 25% квантиль и 75% квантиль | 159 и 170 |
| Интерквартильный размах | 11 |
| Минимум | 59 |
| Максимум | 207 |
| 95% ДИ для среднего | 164.36 - 164.53 |
| weight | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 71.59 |
| Медиана | 70 |
| Станд. отклон. | 13.31 |
| 25% квантиль и 75% квантиль | 63 и 79 |
| Интерквартильный размах | 16 |
| Минимум | 22 |
| Максимум | 200 |
| 95% ДИ для среднего | 71.45 - 71.73 |
| systolic\_blood\_pressure | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 120.43 |
| Медиана | 120 |
| Станд. отклон. | 103.55 |
| 25% квантиль и 75% квантиль | 110 и 120 |
| Интерквартильный размах | 10 |
| Минимум | -120 |
| Максимум | 14020 |
| 95% ДИ для среднего | 119.35 - 121.52 |
| diastolic\_blood\_pressure | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 84.25 |
| Медиана | 80 |
| Станд. отклон. | 152.69 |
| 25% квантиль и 75% квантиль | 70 и 80 |
| Интерквартильный размах | 10 |
| Минимум | 0 |
| Максимум | 9800 |
| 95% ДИ для среднего | 82.65 - 85.85 |
| cholesterol | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 1.22 |
| Медиана | 1 |
| Станд. отклон. | 0.53 |
| 25% квантиль и 75% квантиль | 1 и 1 |
| Интерквартильный размах | 0 |
| Минимум | 1 |
| Максимум | 3 |
| 95% ДИ для среднего | 1.21 - 1.22 |
| glucose | Количество значений | 35021 |
| Количество пропущенных значений | 0 |
| Среднее | 1.18 |
| Медиана | 1 |
| Станд. отклон. | 0.51 |
| 25% квантиль и 75% квантиль | 1 и 1 |
| Интерквартильный размах | 0 |
| Минимум | 1 |
| Максимум | 3 |
| 95% ДИ для среднего | 1.17 - 1.18 |
| Да | age | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 20056.81 |
| Медиана | 20384 |
| Станд. отклон. | 2315.93 |
| 25% квантиль и 75% квантиль | 18387 и 21899 |
| Интерквартильный размах | 3512 |
| Минимум | 14275 |
| Максимум | 23713 |
| 95% ДИ для среднего | 20032.54 - 20081.08 |
| height | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 164.27 |
| Медиана | 165 |
| Станд. отклон. | 8.27 |
| 25% квантиль и 75% квантиль | 159 и 170 |
| Интерквартильный размах | 11 |
| Минимум | 55 |
| Максимум | 250 |
| 95% ДИ для среднего | 164.18 - 164.36 |
| weight | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 76.82 |
| Медиана | 75 |
| Станд. отклон. | 14.96 |
| 25% квантиль и 75% квантиль | 66 и 85 |
| Интерквартильный размах | 19 |
| Минимум | 10 |
| Максимум | 200 |
| 95% ДИ для среднего | 76.67 - 76.98 |
| systolic\_blood\_pressure | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 137.21 |
| Медиана | 130 |
| Станд. отклон. | 191.29 |
| 25% квантиль и 75% квантиль | 120 и 140 |
| Интерквартильный размах | 20 |
| Минимум | -150 |
| Максимум | 16020 |
| 95% ДИ для среднего | 135.21 - 139.22 |
| diastolic\_blood\_pressure | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 109.02 |
| Медиана | 80 |
| Станд. отклон. | 217.81 |
| 25% квантиль и 75% квантиль | 80 и 90 |
| Интерквартильный размах | 10 |
| Минимум | -70 |
| Максимум | 11000 |
| 95% ДИ для среднего | 106.74 - 111.31 |
| cholesterol | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 1.52 |
| Медиана | 1 |
| Станд. отклон. | 0.78 |
| 25% квантиль и 75% квантиль | 1 и 2 |
| Интерквартильный размах | 1 |
| Минимум | 1 |
| Максимум | 3 |
| 95% ДИ для среднего | 1.51 - 1.53 |
| glucose | Количество значений | 34979 |
| Количество пропущенных значений | 0 |
| Среднее | 1.28 |
| Медиана | 1 |
| Станд. отклон. | 0.62 |
| 25% квантиль и 75% квантиль | 1 и 1 |
| Интерквартильный размах | 0 |
| Минимум | 1 |
| Максимум | 3 |
| 95% ДИ для среднего | 1.27 - 1.28 |

## 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.

CI\_categ <- function(t){  
tb <- prop.table(table(t))  
n <- length(t)  
result <- ''  
for (i in names(tb)) {  
 p<- tb[i]  
 result <- paste0(result, toString(paste0((p - qnorm(0.975)\*sqrt(p\*(1-p)/n)) %>% round(2),"-",(p + qnorm(0.975)\*sqrt(p\*(1-p)/n)) %>% round(2))), ", ")  
}  
return(substr(result,1, nchar(result)-2))  
}  
  
  
statistics\_categ <- list(  
 `\_Абсолютное количество` = ~length(.x) %>% as.character(),  
 `\_Количество пропущенных значений` = ~sum(is.na(.x)) %>% as.character(),  
 `\_Группы` = ~ toString(names(prop.table(table(.x)))),  
 `\_Относительное количество внутри группы` = ~toString(as.numeric(prop.table(table(.x)) %>% round(2))),  
 `\_95% ДИ для долей` = ~ifelse(sum(!is.na(.x)) < 3, "Н/П\*", CI\_categ(.x))  
)  
  
cleaned\_data %>%  
 select(cardio, where(is.factor)) %>%  
 group\_by(cardio) %>%  
 reframe(across(where(is.factor), statistics\_categ)) %>%  
 pivot\_longer(!cardio) %>%  
 separate(name, into = c("Variable", "Statistics"), sep = "\_\_") %>%  
 mutate(cardio = factor(cardio, levels=c(0,1), labels=c("Нет", "Да"))) %>%  
 rename (Value = value, "Наличие ССЗ" = cardio) %>%  
 flextable() %>% merge\_v(, j=c("Наличие ССЗ", "Variable")) %>%  
 autofit()

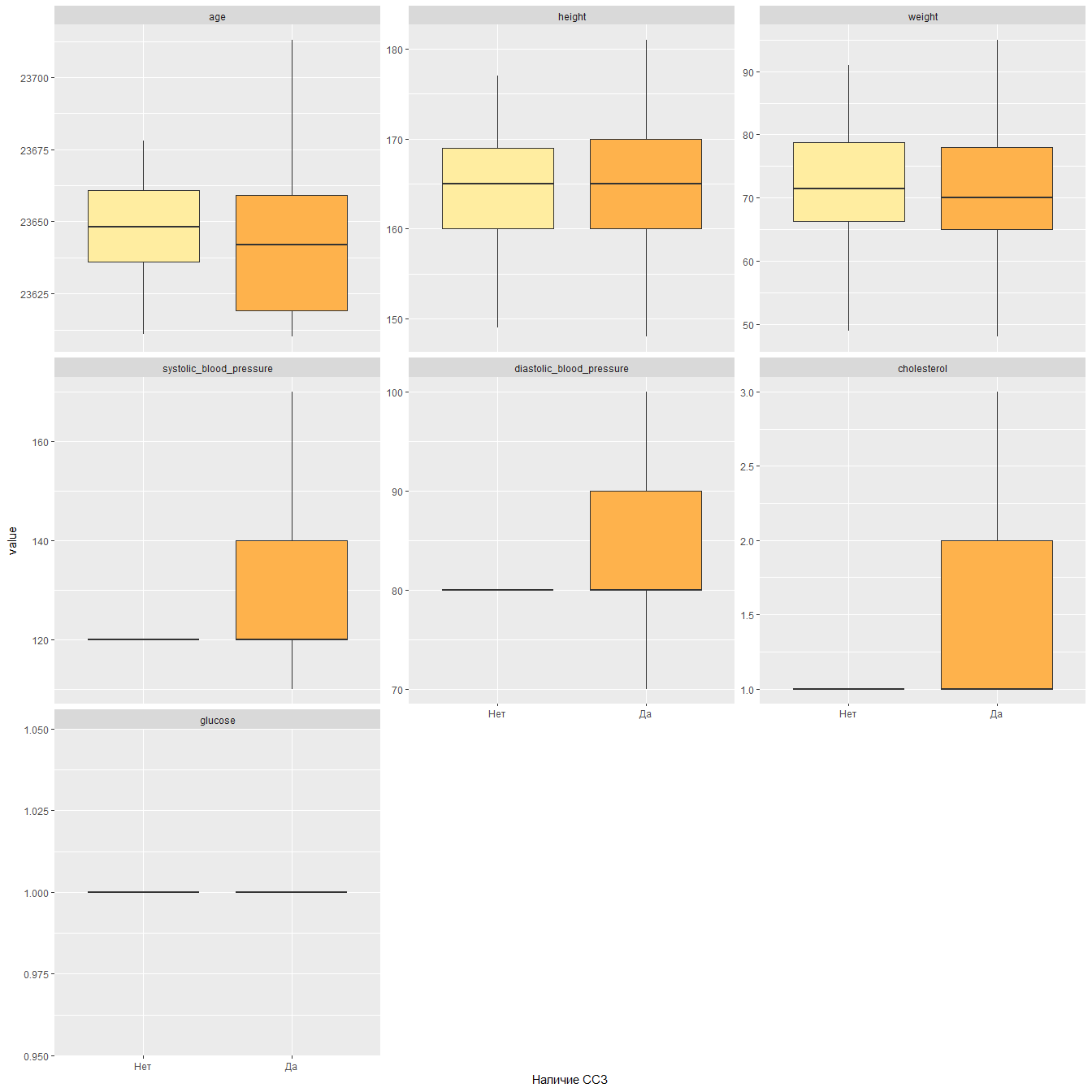
| Наличие ССЗ | Variable | Statistics | Value |
| --- | --- | --- | --- |
| Нет | gender | Абсолютное количество | 35021 |
| Количество пропущенных значений | 0 |
| Группы | 1, 2 |
| Относительное количество внутри группы | 0.65, 0.35 |
| 95% ДИ для долей | 0.65-0.66, 0.34-0.35 |
| smoke | Абсолютное количество | 35021 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.91, 0.09 |
| 95% ДИ для долей | 0.9-0.91, 0.09-0.1 |
| alco | Абсолютное количество | 35021 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.94, 0.06 |
| 95% ДИ для долей | 0.94-0.95, 0.05-0.06 |
| physical\_activity | Абсолютное количество | 35021 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.18, 0.82 |
| 95% ДИ для долей | 0.18-0.19, 0.81-0.82 |
| Да | gender | Абсолютное количество | 34979 |
| Количество пропущенных значений | 0 |
| Группы | 1, 2 |
| Относительное количество внутри группы | 0.65, 0.35 |
| 95% ДИ для долей | 0.64-0.65, 0.35-0.36 |
| smoke | Абсолютное количество | 34979 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.92, 0.08 |
| 95% ДИ для долей | 0.91-0.92, 0.08-0.09 |
| alco | Абсолютное количество | 34979 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.95, 0.05 |
| 95% ДИ для долей | 0.95-0.95, 0.05-0.05 |
| physical\_activity | Абсолютное количество | 34979 |
| Количество пропущенных значений | 0 |
| Группы | 0, 1 |
| Относительное количество внутри группы | 0.21, 0.79 |
| 95% ДИ для долей | 0.21-0.21, 0.79-0.79 |

# 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.

cleaned\_data[1:100,] |>   
 select(cardio, where(is.numeric)) |>   
 mutate(cardio = factor(cardio, levels=c(0,1), labels=c("Нет", "Да"))) |>  
 melt(id=c('id','cardio')) |>  
 ggplot()+  
 #geom\_beeswarm(aes(x=cardio, y=value), method = "hex") +  
 geom\_boxplot(aes(x=cardio, y=value, fill = cardio), outliers = FALSE, show.legend=FALSE)+  
 scale\_fill\_brewer(palette = "YlOrRd") +  
 scale\_x\_discrete(name = "Наличие ССЗ")+  
 facet\_wrap(vars(variable), scales = "free\_y")

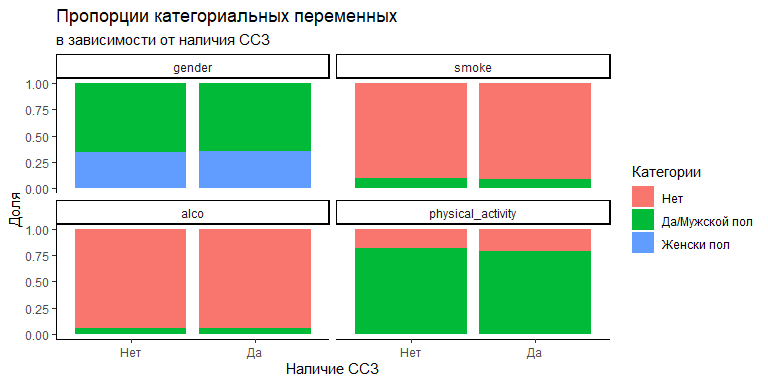


## Категориальные переменные

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

cleaned\_data |>   
 select(id, cardio, where(is.factor)) |>   
 mutate(cardio = factor(cardio, levels=c(0,1), labels=c("Нет", "Да"))) |>  
 melt(id=c('id','cardio')) |>  
 ggplot()+  
 geom\_bar(aes(x = fct\_infreq(cardio),   
 fill = value),   
 position = "fill")+   
 scale\_x\_discrete(name = "Наличие ССЗ")+  
 labs(y = "Доля", title="Пропорции категориальных переменных", subtitle= "в зависимости от наличия ССЗ")+  
 facet\_wrap(vars(variable))+  
 scale\_fill\_discrete(name = "Категории", labels = c("Нет", "Да/Мужской пол", "Женски пол"))+  
  
 theme\_classic()

## Warning: attributes are not identical across measure variables; they will be  
## dropped



**Для визуализации пропорций подходят отнормированные столбики, наглядно демонстрирующие разницу в относительных долях каждой категории между группами.**

# 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?

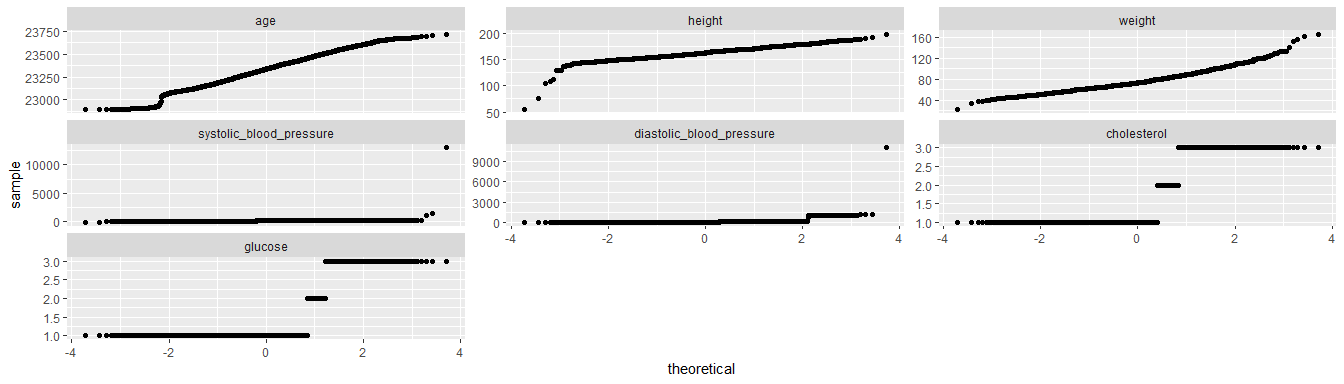
cleaned\_data[sample(5000),] |>   
 select(where(is.numeric) & !c(id)) |> lapply(shapiro.test) |>  
 sapply(`[`, c("statistic","p.value")) |> t()

## statistic p.value   
## age 0.9936364 3.97661e-14   
## height 0.963332 7.954605e-34  
## weight 0.9706512 8.114204e-31  
## systolic\_blood\_pressure 0.02511309 2.988254e-95  
## diastolic\_blood\_pressure 0.0834362 8.441763e-94  
## cholesterol 0.6494206 1.163127e-72  
## glucose 0.5038943 5.576004e-80

**Ответ:** Ни одна из численных переменных не имеет нормальное распределение.

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

cleaned\_data[sample(5000),] |>   
 select(where(is.numeric)) |>  
 melt(id=c('id')) |>  
 ggplot()+  
 geom\_qq(aes(sample = value))+  
 facet\_wrap(vars(variable), scales = "free\_y")



Согласно QQ-графику приближенное к нормальному распределению имеют переменные возраст и рост. Предпочла бы графический метод, так как у него нет ограничений на объем выборки и визуально можно оценить “нормальность”.

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

**Еще есть метод Колмогорова-Смирнова, метод Андерсона-Дарлинга, метод Д’Агостино-Пирсона. Метод Колмогорова-Смирнова считается более подходящим для больших выборок, метод Андерсона-Дарлинга является неспецифическим к форме распределения, метод Д’Агостино следует применять, когда нет сведений об альтернативном распределении. Он показывает хорошую мощность против большого спектра альтернатив, по мощности немного уступая критерию Шапиро - Уилка.**

## 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.

t.test(age~cardio, cleaned\_data)

##   
## Welch Two Sample t-test  
##   
## data: age by cardio  
## t = -64.879, df = 69706, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## -1210.692 -1139.687  
## sample estimates:  
## mean in group 0 mean in group 1   
## 18881.62 20056.81

t.test(height~cardio, cleaned\_data)

##   
## Welch Two Sample t-test  
##   
## data: height by cardio  
## t = 2.8631, df = 69980, p-value = 0.004197  
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0  
## 95 percent confidence interval:  
## 0.05604563 0.29932092  
## sample estimates:  
## mean in group 0 mean in group 1   
## 164.4480 164.2703

wilcox.test(weight~cardio,cleaned\_data)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: weight by cardio  
## W = 483337844, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(systolic\_blood\_pressure~cardio,cleaned\_data)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: systolic\_blood\_pressure by cardio  
## W = 304098842, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(diastolic\_blood\_pressure~cardio,cleaned\_data)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: diastolic\_blood\_pressure by cardio  
## W = 374048881, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(cholesterol~cardio,cleaned\_data)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: cholesterol by cardio  
## W = 496949309, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

wilcox.test(glucose~cardio,cleaned\_data)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: glucose by cardio  
## W = 572316198, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

chisq.test(cleaned\_data$gender, cleaned\_data$cardio)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: cleaned\_data$gender and cleaned\_data$cardio  
## X-squared = 4.5695, df = 1, p-value = 0.03255

chisq.test(cleaned\_data$smoke, cleaned\_data$cardio)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: cleaned\_data$smoke and cleaned\_data$cardio  
## X-squared = 16.678, df = 1, p-value = 4.429e-05

chisq.test(cleaned\_data$alco, cleaned\_data$cardio)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: cleaned\_data$alco and cleaned\_data$cardio  
## X-squared = 3.6965, df = 1, p-value = 0.05453

chisq.test(cleaned\_data$physical\_activity, cleaned\_data$cardio)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: cleaned\_data$physical\_activity and cleaned\_data$cardio  
## X-squared = 88.801, df = 1, p-value < 2.2e-16

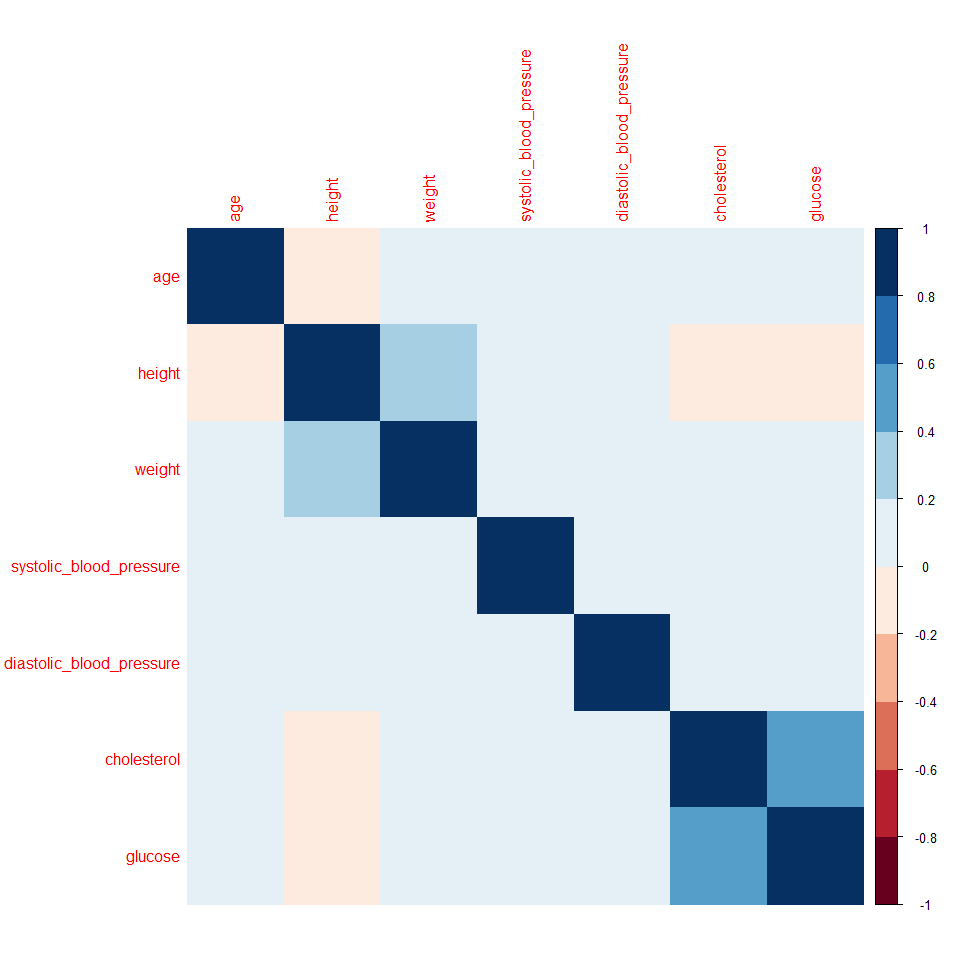
**Для нормально распределенных переменных был выбран t-test. Для ненормально распределенных переменных был выбран критерий критерий Манна-Уитни. Для категориальных переменных был выбран критерий хи-квадрат.**

# 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.

cleaned\_data |>   
 select(where(is.numeric) & !c(id)) |>  
 psych::corr.test(adjust = "bonferroni") -> cor\_results  
   
corrplot(cor\_results$r, method = 'color', col = COL2('RdBu', 10))



**Корреляционные матрицы удобно использовать для обобщения большого набора данных с большим числом количественных и ранговых переменных. Преимущества: результаты корреляционного анализа служат основой для идей и гипотез о взаимосвязях в данных. Недостатки: корреляционный анализ не позволяет определить причину взаимосвязи между двумя переменными.**

## Моделирование

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

library(caTools)  
library(ROCR)  
  
split <- sample.split(cleaned\_data, SplitRatio = 0.8)  
  
   
train\_reg <- subset(cleaned\_data, split == "TRUE")  
test\_reg <- subset(cleaned\_data, split == "FALSE")  
  
# Training model  
logistic\_model <- glm(cardio ~ age + gender + height + weight + systolic\_blood\_pressure + diastolic\_blood\_pressure + cholesterol + glucose + smoke + alco + physical\_activity,  
 data = train\_reg,  
 family = "binomial")

## Warning: glm.fit: алгоритм не сошелся

## Warning: glm.fit: возникли подогнанные вероятности 0 или 1

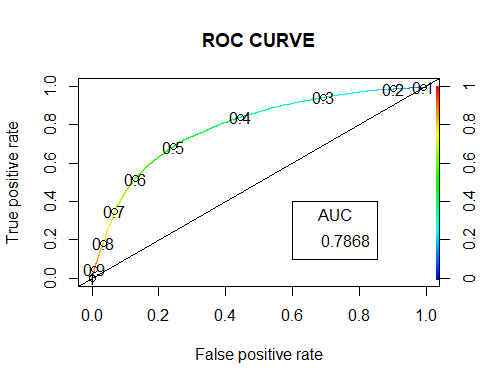
logistic\_model

##   
## Call: glm(formula = cardio ~ age + gender + height + weight + systolic\_blood\_pressure +   
## diastolic\_blood\_pressure + cholesterol + glucose + smoke +   
## alco + physical\_activity, family = "binomial", data = train\_reg)  
##   
## Coefficients:  
## (Intercept) age gender2   
## -7.9535267 0.0001469 0.0449404   
## height weight systolic\_blood\_pressure   
## -0.0061491 0.0158688 0.0353183   
## diastolic\_blood\_pressure cholesterol glucose   
## 0.0005129 0.5366379 -0.1133863   
## smoke1 alco1 physical\_activity1   
## -0.1410915 -0.1293784 -0.2012303   
##   
## Degrees of Freedom: 53845 Total (i.e. Null); 53834 Residual  
## Null Deviance: 74650   
## Residual Deviance: 62710 AIC: 62730

# Summary  
summary(logistic\_model)

##   
## Call:  
## glm(formula = cardio ~ age + gender + height + weight + systolic\_blood\_pressure +   
## diastolic\_blood\_pressure + cholesterol + glucose + smoke +   
## alco + physical\_activity, family = "binomial", data = train\_reg)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.954e+00 2.497e-01 -31.857 < 2e-16 \*\*\*  
## age 1.469e-04 4.021e-06 36.525 < 2e-16 \*\*\*  
## gender2 4.494e-02 2.387e-02 1.882 0.059788 .   
## height -6.149e-03 1.392e-03 -4.418 9.98e-06 \*\*\*  
## weight 1.587e-02 7.471e-04 21.240 < 2e-16 \*\*\*  
## systolic\_blood\_pressure 3.532e-02 6.769e-04 52.178 < 2e-16 \*\*\*  
## diastolic\_blood\_pressure 5.129e-04 1.002e-04 5.120 3.06e-07 \*\*\*  
## cholesterol 5.366e-01 1.706e-02 31.453 < 2e-16 \*\*\*  
## glucose -1.134e-01 1.932e-02 -5.870 4.35e-09 \*\*\*  
## smoke1 -1.411e-01 3.758e-02 -3.754 0.000174 \*\*\*  
## alco1 -1.294e-01 4.538e-02 -2.851 0.004360 \*\*   
## physical\_activity1 -2.012e-01 2.381e-02 -8.450 < 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: 74646 on 53845 degrees of freedom  
## Residual deviance: 62707 on 53834 degrees of freedom  
## AIC: 62731  
##   
## Number of Fisher Scoring iterations: 25

predict\_reg <- predict(logistic\_model,  
 test\_reg, type = "response")  
# ROC-AUC Curve  
ROCPred <- prediction(predict\_reg, test\_reg$cardio)  
ROCPer <- performance(ROCPred, measure = "tpr",  
 x.measure = "fpr")  
auc <- performance(ROCPred, measure = "auc")  
auc <- auc@y.values[[1]]  
   
# Plotting curve  
plot(ROCPer, colorize = TRUE,  
 print.cutoffs.at = seq(0.1, by = 0.1),  
 main = "ROC CURVE")  
abline(a = 0, b = 1)  
   
auc <- round(auc, 4)  
legend(.6, .4, auc, title = "AUC", cex = 1)



**Процесс построения модели: 1. Выбор линейной либо логистической регрессии. 2. Разделение данных на обучающую и тестовую выборки. 3. Построение модели. 4. Оценка модели на тестовой выборке.**