automatization\_notebook

2023-10-27

# Чтение данных

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

df <- read\_csv2("./data/raw/cardio\_train\_big.csv")  
df %>% head()

## # A tibble: 6 × 13  
## id age gender height weight ap\_hi ap\_lo cholesterol gluc smoke alco  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0 18393 2 168 620 110 80 1 1 0 0  
## 2 1 20228 1 156 850 140 90 3 1 0 0  
## 3 2 18857 1 165 640 130 70 3 1 0 0  
## 4 3 17623 2 169 820 150 100 1 1 0 0  
## 5 4 17474 1 156 560 100 60 1 1 0 0  
## 6 8 21914 1 151 670 120 80 2 2 0 0  
## # ℹ 2 more variables: active <dbl>, cardio <dbl>

# Выведите общее описание данных

df %>% 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> 620, 850, 640, 820, 560, 670, 930, 950, 710, 680, 800, 600…  
## $ 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:

df %>% sapply(function(x) sum(is.na(x)))

## id age gender height weight ap\_hi   
## 0 0 0 0 0 0   
## ap\_lo cholesterol gluc smoke alco active   
## 0 0 0 0 0 0   
## cardio   
## 0

**Обоснование**: В датасете отсутствуют пропущенные значения.

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”.

df <- df %>%  
 rename('Arterial Pressure High' = ap\_hi,  
 'Arterial Pressure Low' = ap\_lo,  
 'glucose' = gluc,  
 'alcohol' = alco) %>%   
 mutate(id = as.factor(id),  
 gender = as.factor(gender),  
 cholesterol = as.factor(cholesterol),  
 glucose = as.factor(glucose),  
 smoke = as.factor(smoke),  
 alcohol = as.factor(alcohol),  
 active = as.factor(active),  
 cardio = as.factor(cardio)) %>%   
 arrange(desc(age))  
df %>% summary()

## id age gender height weight   
## 0 : 1 Min. :10798 1:45530 Min. : 55.0 Min. : 100.0   
## 1 : 1 1st Qu.:17664 2:24470 1st Qu.:159.0 1st Qu.: 650.0   
## 2 : 1 Median :19703 Median :165.0 Median : 720.0   
## 3 : 1 Mean :19469 Mean :164.4 Mean : 742.5   
## 4 : 1 3rd Qu.:21327 3rd Qu.:170.0 3rd Qu.: 820.0   
## 8 : 1 Max. :23713 Max. :250.0 Max. :7994.0   
## (Other):69994   
## Arterial Pressure High Arterial Pressure Low cholesterol glucose smoke   
## Min. : -150.0 Min. : -70.00 1:52385 1:59479 0:63831   
## 1st Qu.: 120.0 1st Qu.: 80.00 2: 9549 2: 5190 1: 6169   
## Median : 120.0 Median : 80.00 3: 8066 3: 5331   
## Mean : 128.8 Mean : 96.63   
## 3rd Qu.: 140.0 3rd Qu.: 90.00   
## Max. :16020.0 Max. :11000.00   
##   
## alcohol active cardio   
## 0:66236 0:13739 0:35021   
## 1: 3764 1:56261 1:34979   
##   
##   
##   
##   
##

tibble позволяет пробелы в названиях (хотя я не уверен, что новые названия - хорошая идея, потому что прежние названия были вполне приемлемые). Наблюдаются явные выбросы в height, weight, Arterial Pressure High, Arterial Pressure Low.

df %>% glimpse()

## Rows: 70,000  
## Columns: 13  
## $ id <fct> 81643, 97124, 72373, 29892, 52297, 53805, 573…  
## $ age <dbl> 23713, 23701, 23692, 23690, 23687, 23684, 236…  
## $ gender <fct> 2, 1, 2, 1, 2, 2, 1, 1, 1, 1, 2, 2, 1, 1, 1, …  
## $ height <dbl> 175, 178, 167, 165, 163, 165, 165, 169, 140, …  
## $ weight <dbl> 750, 750, 680, 630, 550, 640, 740, 700, 480, …  
## $ `Arterial Pressure High` <dbl> 120, 120, 120, 150, 120, 120, 120, 120, 170, …  
## $ `Arterial Pressure Low` <dbl> 80, 80, 80, 90, 80, 80, 80, 80, 100, 80, 80, …  
## $ cholesterol <fct> 1, 1, 2, 2, 3, 1, 1, 3, 2, 1, 2, 1, 1, 1, 1, …  
## $ glucose <fct> 1, 1, 2, 1, 1, 1, 1, 3, 1, 1, 2, 1, 1, 1, 1, …  
## $ smoke <fct> 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ alcohol <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …  
## $ active <fct> 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, …  
## $ cardio <fct> 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, …

Фильтровать решил по интерквартильному размаху.

find\_outliers <- function(x) {  
 q1 <- quantile(x, 0.25)  
 q3 <- quantile(x, 0.75)  
 iqr <- q3 - q1  
 lower\_bound <- q1 - 1.5 \* iqr  
 upper\_bound <- q3 + 1.5 \* iqr  
 outliers <- x < lower\_bound | x > upper\_bound  
 return(outliers)  
}  
outliers <- df %>%  
 filter\_if(is.numeric, any\_vars(find\_outliers(.))) %>%   
 write\_csv2(file = "./data/output/outliers.csv")

Данные после вычитания выбросов записаны в cleaned\_data.

cleaned\_data <- df %>%  
 anti\_join(outliers, by = "id")   
cleaned\_data %>% summary()

## id age gender height weight   
## 0 : 1 Min. :14282 1:40711 Min. :143.0 Min. : 400.0   
## 1 : 1 1st Qu.:17724 2:21787 1st Qu.:159.0 1st Qu.: 650.0   
## 2 : 1 Median :19718 Median :165.0 Median : 720.0   
## 3 : 1 Mean :19494 Mean :164.4 Mean : 731.8   
## 8 : 1 3rd Qu.:21341 3rd Qu.:170.0 3rd Qu.: 810.0   
## 9 : 1 Max. :23713 Max. :186.0 Max. :1070.0   
## (Other):62492   
## Arterial Pressure High Arterial Pressure Low cholesterol glucose smoke   
## Min. : 90.0 Min. : 65.0 1:47209 1:53401 0:57101   
## 1st Qu.:120.0 1st Qu.: 80.0 2: 8222 2: 4403 1: 5397   
## Median :120.0 Median : 80.0 3: 7067 3: 4694   
## Mean :126.4 Mean : 81.7   
## 3rd Qu.:140.0 3rd Qu.: 90.0   
## Max. :170.0 Max. :105.0   
##   
## alcohol active cardio   
## 0:59237 0:12231 0:31631   
## 1: 3261 1:50267 1:30867   
##   
##   
##   
##   
##

# Сколько осталось переменных?

cleaned\_data %>% ncol()

## [1] 13

# Сколько осталось случаев?

cleaned\_data %>% nrow()

## [1] 62498

# Есть ли в данных идентичные строки?

cleaned\_data %>%   
 duplicated() %>%   
 sum()

## [1] 0

Идентичные строки отсутствуют.

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

cleaned\_data %>%   
 sapply(function(x) sum(is.na(x))) %>%   
 as.data.frame() %>%   
 rename('NA'='.')

## NA  
## id 0  
## age 0  
## gender 0  
## height 0  
## weight 0  
## Arterial Pressure High 0  
## Arterial Pressure Low 0  
## cholesterol 0  
## glucose 0  
## smoke 0  
## alcohol 0  
## active 0  
## cardio 0

Пропущенные значения отсутствуют.

# 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.

Не нашел готового решения, которое бы удовлетворяло требованиям, поэтому написал свою функцию. Правда вывод - не в виде датафрейма.

my\_summarise\_num <- function(x){  
 Count = length(x)  
 Missing = sum(is.na(x))  
 Mean = mean(x, na.rm = TRUE)  
 Median = median(x, na.rm = TRUE)  
 StdDev = sd(x, na.rm = TRUE)  
 Q1 = quantile(x, 0.25, na.rm = TRUE)  
 Q3 = quantile(x, 0.75, na.rm = TRUE)  
 IQR = Q3 - Q1  
 Min = min(x, na.rm = TRUE)  
 Max = max(x, na.rm = TRUE)  
 CI\_95\_Lower = Mean - qt(0.975, Count - 1) \* (StdDev / sqrt(Count))  
 CI\_95\_Upper = Mean + qt(0.975, Count - 1) \* (StdDev / sqrt(Count))  
 data.frame(  
 Count = Count,  
 Missing = Missing,  
 Mean = Mean,  
 Median = Median,  
 StdDev = StdDev,  
 Q1 = Q1,  
 Q3 = Q3,  
 IQR = IQR,  
 Min = Min,  
 Max = Max,  
 CI\_95\_Lower = CI\_95\_Lower,  
 CI\_95\_Upper = CI\_95\_Upper  
 )  
}  
cat("cardio = 0\n")

## cardio = 0

cleaned\_data %>%  
 filter(cardio==0) %>%   
 select\_if(is.numeric) %>%   
 sapply(my\_summarise\_num)

## age height weight Arterial Pressure High  
## Count 31631 31631 31631 31631   
## Missing 0 0 0 0   
## Mean 18919.4 164.6116 712.2142 120.3195   
## Median 19029 165 700 120   
## StdDev 2462.397 7.460326 117.7235 10.68391   
## Q1 16877 160 630 120   
## Q3 20948.5 170 780 120   
## IQR 4071.5 10 150 0   
## Min 14282 143 400 90   
## Max 23678 186 1070 170   
## CI\_95\_Lower 18892.26 164.5293 710.9168 120.2018   
## CI\_95\_Upper 18946.53 164.6938 713.5115 120.4373   
## Arterial Pressure Low  
## Count 31631   
## Missing 0   
## Mean 79.15033   
## Median 80   
## StdDev 6.459564   
## Q1 80   
## Q3 80   
## IQR 0   
## Min 65   
## Max 105   
## CI\_95\_Lower 79.07914   
## CI\_95\_Upper 79.22152

cat("\n\ncardio = 1\n")

##   
##   
## cardio = 1

cleaned\_data %>%  
 filter(cardio==1) %>%   
 select\_if(is.numeric) %>%   
 sapply(my\_summarise\_num)

## age height weight Arterial Pressure High  
## Count 30867 30867 30867 30867   
## Missing 0 0 0 0   
## Mean 20082.18 164.1964 751.9134 132.6701   
## Median 20400 164 740 130   
## StdDev 2310.084 7.597999 124.4719 14.80564   
## Q1 18417 159 660 120   
## Q3 21909 169 840 140   
## IQR 3492 10 180 20   
## Min 14287 143 400 90   
## Max 23713 186 1070 170   
## CI\_95\_Lower 20056.41 164.1116 750.5247 132.5049   
## CI\_95\_Upper 20107.95 164.2812 753.302 132.8352   
## Arterial Pressure Low  
## Count 30867   
## Missing 0   
## Mean 84.31095   
## Median 80   
## StdDev 7.936126   
## Q1 80   
## Q3 90   
## IQR 10   
## Min 65   
## Max 105   
## CI\_95\_Lower 84.22241   
## CI\_95\_Upper 84.39948

IQR для давлений маленький, что впоследсвии отобразится на boxplot’ах. Для группы cardio - равен 0.

## 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.

Так как уровни фактора отличаются, я вывод разбил на 3 группы: 1) gender, 2) cholesterol, glucose, 3) smoke, alcohol, active. Каждая группа разбита на 2 по наличию заболевания. Для всех групп посчитаны статистики. Возможно, хорошая была бы идея привести все категориальные признаки к единому виду (levels: 0, 1), чтобы удобно было интерпретировать(?).

my\_summarise\_fac <- function(x, l) {  
 x <- as.numeric(unlist(x))  
 n <- length(x)  
 n\_f <- sum(x==l)  
 p <- n\_f / n  
 se <- sqrt(p \* (1 - p) / n)  
 ci\_lower <- p - 1.96 \* se  
 ci\_upper <- p + 1.96 \* se  
 return(data.frame(  
 Count\_all = n,  
 level = l,  
 Count\_factor = n\_f,  
 se = se,  
 Proportion = p,  
 Lower\_CI = ci\_lower,  
 Upper\_CI = ci\_upper  
 ))  
}  
cat('cardio==0\n')

## cardio==0

cleaned\_data %>%   
 filter(cardio==0) %>%   
 select(gender) %>%   
 sapply(my\_summarise\_fac, 1)

## gender   
## Count\_all 31631   
## level 1   
## Count\_factor 20533   
## se 0.002683361  
## Proportion 0.6491417   
## Lower\_CI 0.6438823   
## Upper\_CI 0.6544011

cat('\n')

cat('cardio==1\n')

## cardio==1

cleaned\_data %>%   
 filter(cardio==1) %>%   
 select(gender) %>%   
 sapply(my\_summarise\_fac, 1)

## gender   
## Count\_all 30867   
## level 1   
## Count\_factor 20178   
## se 0.002708108  
## Proportion 0.6537078   
## Lower\_CI 0.6484   
## Upper\_CI 0.6590157

cat('\n')

cat('cardio==0\n')

## cardio==0

cleaned\_data %>%   
 filter(cardio==0) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 1)

## cholesterol glucose   
## Count\_all 31631 31631   
## level 1 1   
## Count\_factor 26613 27978   
## se 0.002054201 0.001797065  
## Proportion 0.8413582 0.884512   
## Lower\_CI 0.8373319 0.8809898   
## Upper\_CI 0.8453844 0.8880343

cat('\n')

cat('cardio==1\n')

## cardio==1

cleaned\_data %>%   
 filter(cardio==1) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 1)

## cholesterol glucose   
## Count\_all 30867 30867   
## level 1 1   
## Count\_factor 20596 25423   
## se 0.002681984 0.002169356  
## Proportion 0.6672498 0.8236304   
## Lower\_CI 0.6619931 0.8193785   
## Upper\_CI 0.6725065 0.8278824

cat('\n')

cat('cardio==0\n')

## cardio==0

cleaned\_data %>%   
 filter(cardio==0) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 2)

## cholesterol glucose   
## Count\_all 31631 31631   
## level 2 2   
## Count\_factor 3325 1847   
## se 0.00172451 0.001318425  
## Proportion 0.1051184 0.05839208   
## Lower\_CI 0.1017384 0.05580797   
## Upper\_CI 0.1084984 0.0609762

cat('\n')

cat('cardio==1\n')

## cardio==1

cleaned\_data %>%   
 filter(cardio==1) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 2)

## cholesterol glucose   
## Count\_all 30867 30867   
## level 2 2   
## Count\_factor 4897 2556   
## se 0.002079502 0.001568615  
## Proportion 0.1586484 0.08280688   
## Lower\_CI 0.1545726 0.0797324   
## Upper\_CI 0.1627242 0.08588137

cat('\n')

cat('cardio==0\n')

## cardio==0

cleaned\_data %>%   
 filter(cardio==0) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 3)

## cholesterol glucose   
## Count\_all 31631 31631   
## level 3 3   
## Count\_factor 1693 1806   
## se 0.001265524 0.001304607  
## Proportion 0.05352344 0.05709589   
## Lower\_CI 0.05104301 0.05453886   
## Upper\_CI 0.05600387 0.05965292

cat('\n')

cat('cardio==1\n')

## cardio==1

cleaned\_data %>%   
 filter(cardio==1) %>%   
 select(cholesterol, glucose) %>%   
 sapply(my\_summarise\_fac, 3)

## cholesterol glucose   
## Count\_all 30867 30867   
## level 3 3   
## Count\_factor 5374 2888   
## se 0.002158329 0.001657574  
## Proportion 0.1741018 0.0935627   
## Lower\_CI 0.1698715 0.09031386   
## Upper\_CI 0.1783321 0.09681155

cat('\n')

cat('cardio==0\n')

## cardio==0

cleaned\_data %>%   
 filter(cardio==0) %>%   
 select(smoke, alcohol, active) %>%   
 sapply(my\_summarise\_fac, 1)

## smoke alcohol active   
## Count\_all 31631 31631 31631   
## level 1 1 1   
## Count\_factor 28705 29895 5723   
## se 0.001629096 0.001280574 0.002164509  
## Proportion 0.9074958 0.9451171 0.1809301   
## Lower\_CI 0.9043028 0.9426072 0.1766877   
## Upper\_CI 0.9106888 0.9476271 0.1851725

cat('\n')

cat('cardio==1\n')

## cardio==1

cleaned\_data %>%   
 filter(cardio==1) %>%   
 select(smoke, alcohol, active) %>%   
 sapply(my\_summarise\_fac, 1)

## smoke alcohol active   
## Count\_all 30867 30867 30867   
## level 1 1 1   
## Count\_factor 28396 29342 6508   
## se 0.001544626 0.001233497 0.002321731  
## Proportion 0.9199469 0.9505945 0.2108401   
## Lower\_CI 0.9169194 0.9481768 0.2062895   
## Upper\_CI 0.9229743 0.9530121 0.2153906

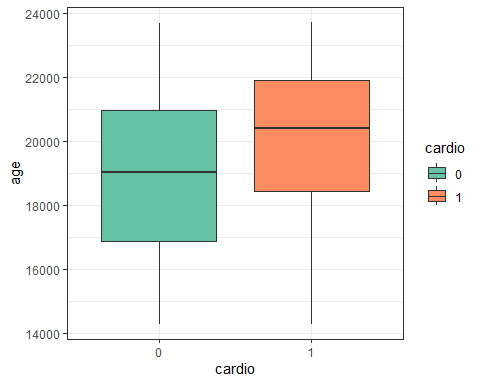
cat('\n')

# Визуализация

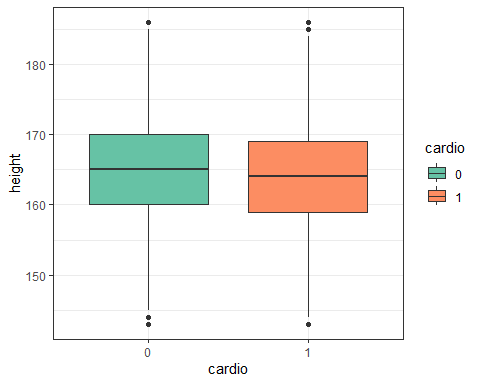
## Количественные переменные

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.

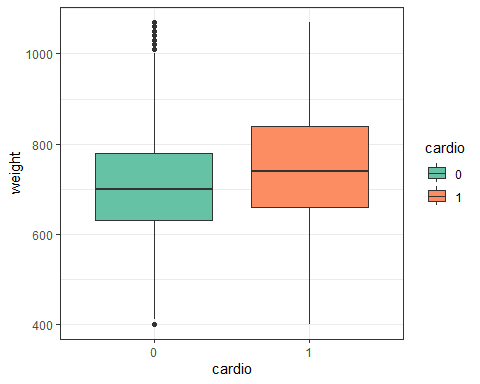
library(ggplot2)  
library(RColorBrewer)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>% ggplot() +  
 aes(cardio, age, fill=cardio) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = palette) +  
 theme\_bw()



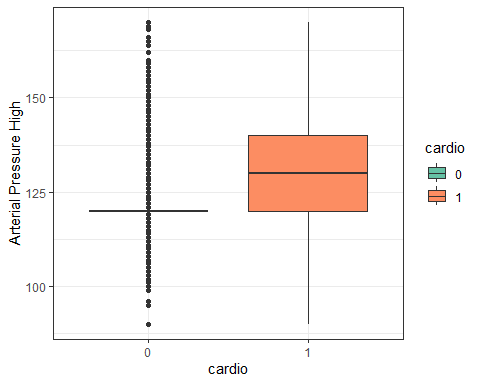
library(ggplot2)  
library(RColorBrewer)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>% ggplot() +  
 aes(cardio, height, fill=cardio) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



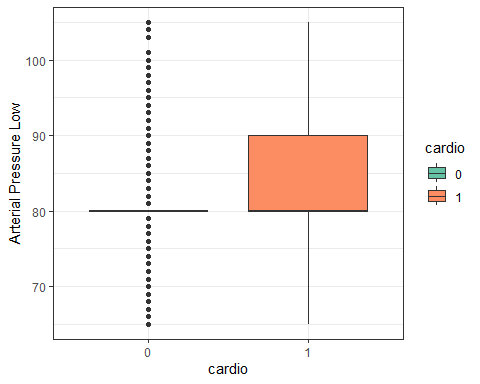
library(ggplot2)  
library(RColorBrewer)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>% ggplot() +  
 aes(cardio, weight, fill=cardio) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



library(ggplot2)  
library(RColorBrewer)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>% ggplot() +  
 aes(cardio, `Arterial Pressure High`, fill=cardio) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



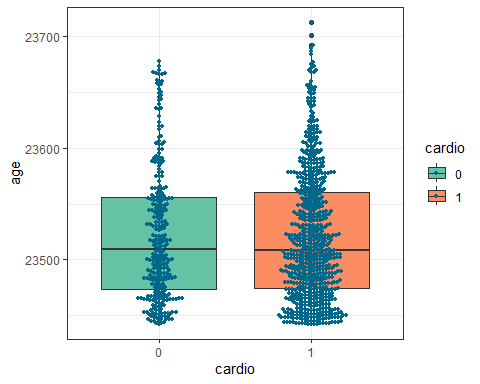
library(ggplot2)  
library(RColorBrewer)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>% ggplot() +  
 aes(cardio, `Arterial Pressure Low`, fill=cardio) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



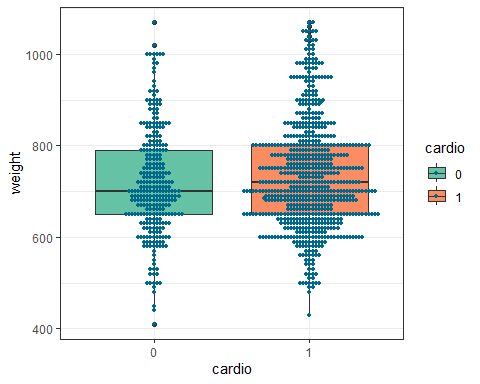
В давлениях группы cardio=0 много выбросов, из-за того, что IQR=0. Это следствие ненормального распределения (дискретность).

Далее beeswarm сделал только для 1000 наблюдений как пример. Если делать больше, то рой пчёл разлетается слишком сильно, перекрывая весь график (не разобрался, можно ли это сделать более эстетично)

library(ggplot2)  
library(RColorBrewer)  
library(ggbeeswarm)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>%   
 head(1000) %>%   
 ggplot() +  
 aes(cardio, age, fill = cardio) +  
 geom\_boxplot() +  
 geom\_beeswarm(color = "deepskyblue4", size = 1) +  
 scale\_fill\_manual(values = palette) +  
 theme\_bw()



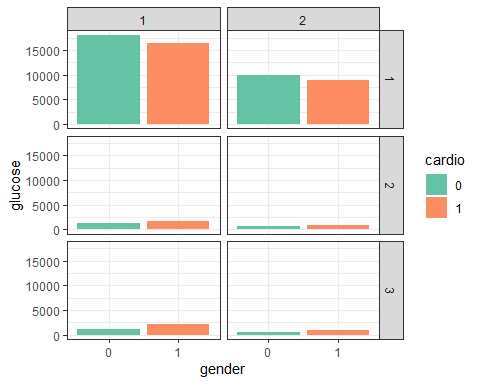
library(ggplot2)  
library(RColorBrewer)  
library(ggbeeswarm)  
palette <- brewer.pal(3, "Set2")  
cleaned\_data %>%   
 head(1000) %>%   
 ggplot() +  
 aes(cardio, weight, fill=cardio) +  
 geom\_boxplot() +  
 geom\_beeswarm(color = "deepskyblue4", size = 1) +  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



## Категориальные переменные

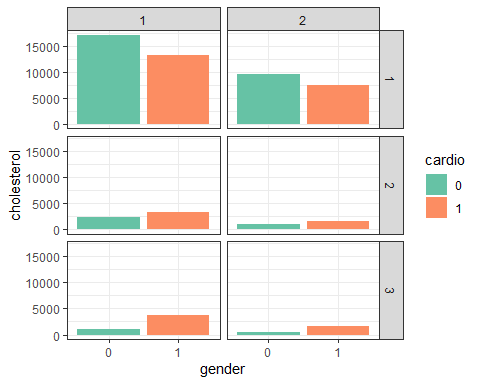
1. Сделайте подходящие визуализации категориальных переменных. Обоснуйте, почему выбрали именно этот тип.

library(ggplot2)  
palette <- brewer.pal(3, "Set2")  
ggplot(cleaned\_data, aes(x=cardio, fill=cardio)) +  
 geom\_bar() +  
 facet\_grid(glucose ~ gender) +  
 xlab("gender") +  
 ylab("glucose")+  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



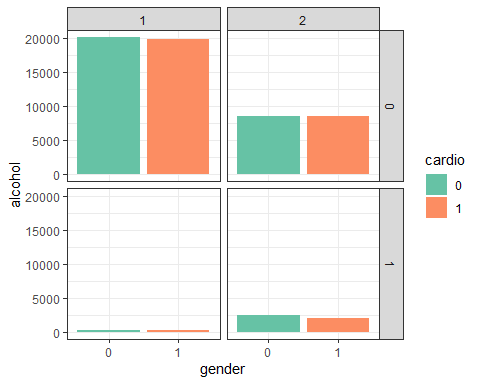
Видно, что соотношение больных меняется в худшую сторону с увеличением глюкозы, вне зависимости от пола.

library(ggplot2)  
palette <- brewer.pal(3, "Set2")  
ggplot(cleaned\_data, aes(x=cardio, fill=cardio)) +  
 geom\_bar() +  
 facet\_grid(cholesterol ~ gender) +  
 xlab("gender") +  
 ylab("cholesterol")+  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



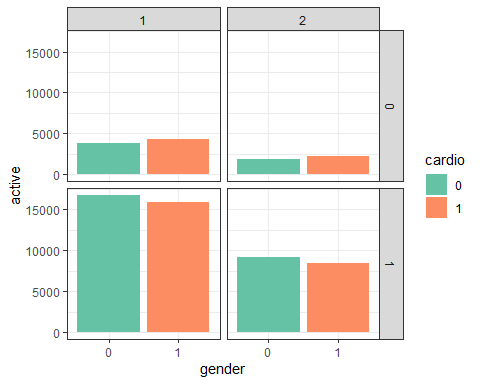
Видно, что холестерол тоже плохо влияет на соотношение больных и здоровых.

library(ggplot2)  
ggplot(cleaned\_data, aes(x=cardio, fill=cardio)) +  
 geom\_bar() +  
 facet\_grid(smoke ~ gender) +  
 xlab("gender") +  
 ylab("alcohol")+  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



Видно, что алкоголь отрицательно влияет на соотношение здоровых и больных, вне зависимости от пола.

library(ggplot2)  
ggplot(cleaned\_data, aes(x=cardio, fill=cardio)) +  
 geom\_bar() +  
 facet\_grid(active ~ gender) +  
 xlab("gender") +  
 ylab("active")+  
 scale\_fill\_manual(values = palette)+  
 theme\_bw()



Активность положительно влияет на соотношение здоровых и больных, для мужчин и женщин.

# 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?

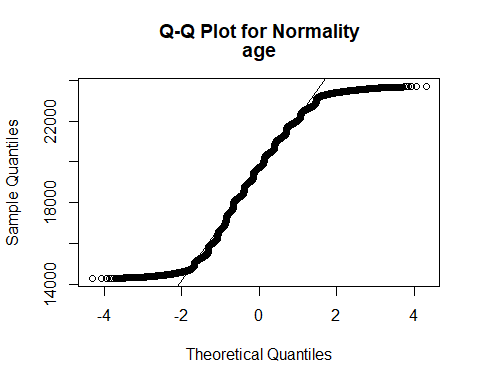
library(nortest)  
normality.test <- function(x){  
 return(sapply(x[0:5000,], shapiro.test)["p.value",])  
}  
cleaned\_data %>%   
 select\_if(is.numeric) %>%   
 normality.test() %>%   
 as.data.frame()

## age height weight Arterial.Pressure.High  
## 1 7.869306e-44 3.241247e-12 3.365501e-19 3.01076e-44  
## Arterial.Pressure.Low  
## 1 2.670767e-56

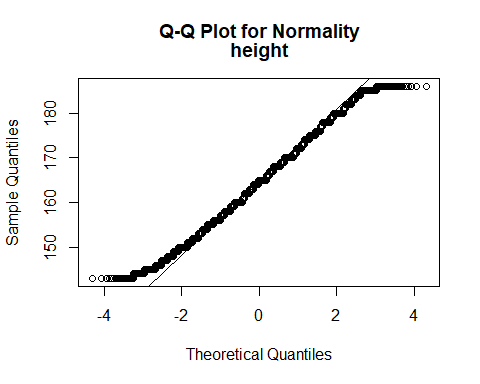
Из-за ограничений функции, shapiro.test проведен для первых 5000 наблюдений. В результат говорит о том, что все величины имеют ненормальное распределение (p < 0.05).

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

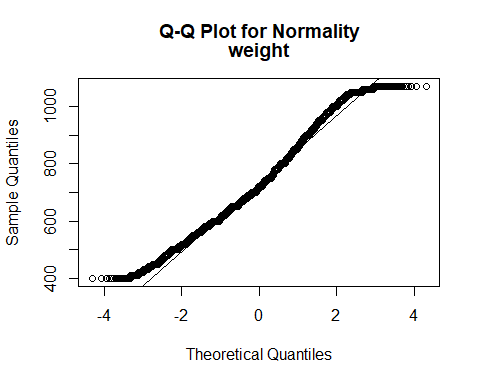
QQ <- function(x, label){  
 qqnorm(x, main = c('Q-Q Plot for Normality\t',label))  
 qqline(x)  
}  
QQ(cleaned\_data$age, 'age')



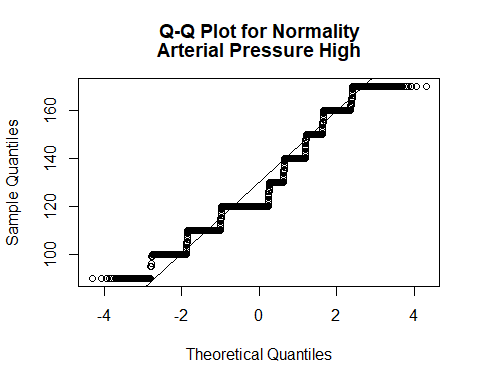
QQ(cleaned\_data$height, 'height')



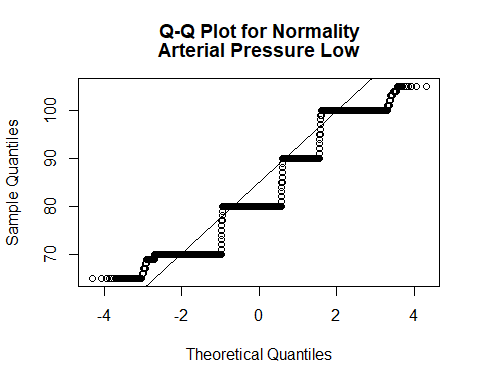
QQ(cleaned\_data$weight, 'weight')



QQ(cleaned\_data$`Arterial Pressure High`, 'Arterial Pressure High')



QQ(cleaned\_data$`Arterial Pressure Low`, 'Arterial Pressure Low')



“Arterial Pressure High” и “Arterial Pressure Low” имеют ступеньчатое распределение, в результате методики сбора данных.

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

library(nortest)  
normality.test <- function(x){  
 return(sapply(x, ad.test)["p.value",])  
}  
cleaned\_data %>%   
 select\_if(is.numeric) %>%   
 normality.test() %>%   
 as.data.frame()

## age height weight Arterial.Pressure.High Arterial.Pressure.Low  
## 1 3.7e-24 3.7e-24 3.7e-24 3.7e-24 3.7e-24

**Напишите текст здесь** Anderson-Darling тест выбран из-за отсутствия ограничений и высокой мощности. Результат - распределение ненормальное.

## 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 <- data.frame('age' = t.test(age ~ cardio, cleaned\_data)$statistic,  
 'height' = t.test(height ~ cardio, cleaned\_data)$statistic,  
 'weight' = t.test(height ~ cardio, cleaned\_data)$statistic,  
 'Arterial Pressure High' = t.test(`Arterial Pressure High` ~ cardio, cleaned\_data)$statistic,  
 'Arterial Pressure Low' = t.test(`Arterial Pressure Low` ~ cardio, cleaned\_data)$statistic)  
p <- data.frame('age' = t.test(age ~ cardio, cleaned\_data)$p.value,  
 'height' = t.test(height ~ cardio, cleaned\_data)$p.value,  
 'weight' = t.test(height ~ cardio, cleaned\_data)$p.value,  
 'Arterial Pressure High' = t.test(`Arterial Pressure High` ~ cardio, cleaned\_data)$p.value,  
 'Arterial Pressure Low' = t.test(`Arterial Pressure Low` ~ cardio, cleaned\_data)$p.value)  
rownames(p) <- "p"  
rbind(t, p)

## age height weight Arterial.Pressure.High  
## t -60.89823 6.890866e+00 6.890866e+00 -119.3396  
## p 0.00000 5.597803e-12 5.597803e-12 0.0000  
## Arterial.Pressure.Low  
## t -89.03468  
## p 0.00000

t-test показал статистически значимые различия в количественных переменных. Была идея сделать Манна-Уитни из-за нормального распределения (но я не сделал).

tribble(  
 ~Variable , ~statistic, ~p\_value,  
 'gender', chisq.test(cleaned\_data$gender, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$gender, cleaned\_data$cardio)$p.value,  
 'cholesterol', chisq.test(cleaned\_data$cholesterol, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$cholesterol, cleaned\_data$cardio)$p.value,  
 'glucose', chisq.test(cleaned\_data$glucose, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$glucose, cleaned\_data$cardio)$p.value,  
 'smoke', chisq.test(cleaned\_data$smoke, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$smoke, cleaned\_data$cardio)$p.value,  
 'alcohol', chisq.test(cleaned\_data$alcohol, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$alcohol, cleaned\_data$cardio)$p.value,  
 'active', chisq.test(cleaned\_data$active, cleaned\_data$cardio)$statistic, chisq.test(cleaned\_data$active, cleaned\_data$cardio)$p.value,  
 )

## # A tibble: 6 × 3  
## Variable statistic p\_value  
## <chr> <dbl> <dbl>  
## 1 gender 1.41 2.34e- 1  
## 2 cholesterol 2976. 0   
## 3 glucose 477. 3.29e-104  
## 4 smoke 30.5 3.27e- 8  
## 5 alcohol 9.37 2.21e- 3  
## 6 active 88.6 4.83e- 21

chisq.test показал, что различия во всех факторных переменных, кроме пола, являются статистически значимыми.

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

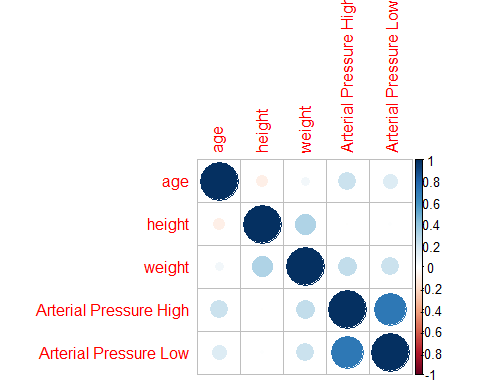
## Корреляционный анализ

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.

library(corrplot)

## corrplot 0.92 loaded

cleaned\_data %>%   
 select\_if(is.numeric) %>%   
 cor(method = "spearman") %>%   
 corrplot()



library(psych)

##   
## Присоединяю пакет: 'psych'

## Следующие объекты скрыты от 'package:ggplot2':  
##   
## %+%, alpha

cleaned\_data %>%   
 select\_if(is.numeric) %>%  
 corr.test(adjust = "bonferroni", method = "spearman")

## Call:corr.test(x = ., method = "spearman", adjust = "bonferroni")  
## Correlation matrix   
## age height weight Arterial Pressure High  
## age 1.00 -0.09 0.06 0.21  
## height -0.09 1.00 0.30 0.00  
## weight 0.06 0.30 1.00 0.24  
## Arterial Pressure High 0.21 0.00 0.24 1.00  
## Arterial Pressure Low 0.15 0.01 0.21 0.72  
## Arterial Pressure Low  
## age 0.15  
## height 0.01  
## weight 0.21  
## Arterial Pressure High 0.72  
## Arterial Pressure Low 1.00  
## Sample Size   
## [1] 62498  
## Probability values (Entries above the diagonal are adjusted for multiple tests.)   
## age height weight Arterial Pressure High  
## age 0 0.00 0 0  
## height 0 0.00 0 1  
## weight 0 0.00 0 0  
## Arterial Pressure High 0 0.34 0 0  
## Arterial Pressure Low 0 0.04 0 0  
## Arterial Pressure Low  
## age 0.00  
## height 0.39  
## weight 0.00  
## Arterial Pressure High 0.00  
## Arterial Pressure Low 0.00  
##   
## To see confidence intervals of the correlations, print with the short=FALSE option

Так как распределение ненормальное, выбрал Спирмена с поправкой Бонферони. Несмотря на консервативность поправки Бонферони, корреляции всех переменных, кроме “Arterial Pressure High” и “height”, “Arterial Pressure Low” и “height”, оказались статистически значимыми.

Плюсы корреляционных исследований: - Корреляция может помочь идентифицировать и измерить силу связи между переменными. - Простота и интерпретируемость. - Исследования корреляции могут помочь исключить из анализа переменные, которые слабо связаны с целевой переменной.

Минусы корреляционных исследований: - Не означают причинно-следственная связь. - Ограничение на линейную связь. - Корреляция чувствительна к выбросам и аномалиям в данных.

## Моделирование

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

glm(cardio ~ ., cleaned\_data[,-1], family = binomial) %>%   
 step() %>%   
 summary()

## Start: AIC=70253.06  
## cardio ~ age + gender + height + weight + `Arterial Pressure High` +   
## `Arterial Pressure Low` + cholesterol + glucose + smoke +   
## alcohol + active  
##   
## Df Deviance AIC  
## - gender 1 70227 70253  
## <none> 70225 70253  
## - height 1 70233 70259  
## - smoke 1 70245 70271  
## - alcohol 1 70249 70275  
## - glucose 2 70286 70310  
## - `Arterial Pressure Low` 1 70299 70325  
## - active 1 70332 70358  
## - weight 1 70432 70458  
## - cholesterol 2 71213 71237  
## - age 1 71550 71576  
## - `Arterial Pressure High` 1 74364 74390  
##   
## Step: AIC=70252.54  
## cardio ~ age + height + weight + `Arterial Pressure High` + `Arterial Pressure Low` +   
## cholesterol + glucose + smoke + alcohol + active  
##   
## Df Deviance AIC  
## <none> 70227 70253  
## - height 1 70241 70265  
## - smoke 1 70250 70274  
## - alcohol 1 70251 70275  
## - glucose 2 70288 70310  
## - `Arterial Pressure Low` 1 70300 70324  
## - active 1 70333 70357  
## - weight 1 70434 70458  
## - cholesterol 2 71217 71239  
## - age 1 71551 71575  
## - `Arterial Pressure High` 1 74364 74388

##   
## Call:  
## glm(formula = cardio ~ age + height + weight + `Arterial Pressure High` +   
## `Arterial Pressure Low` + cholesterol + glucose + smoke +   
## alcohol + active, family = binomial, data = cleaned\_data[,   
## -1])  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -1.177e+01 2.537e-01 -46.396 < 2e-16 \*\*\*  
## age 1.401e-04 3.900e-06 35.922 < 2e-16 \*\*\*  
## height -5.006e-03 1.325e-03 -3.777 0.000159 \*\*\*  
## weight 1.187e-03 8.250e-05 14.391 < 2e-16 \*\*\*  
## `Arterial Pressure High` 6.205e-02 1.046e-03 59.335 < 2e-16 \*\*\*  
## `Arterial Pressure Low` 1.519e-02 1.764e-03 8.609 < 2e-16 \*\*\*  
## cholesterol2 3.640e-01 2.892e-02 12.587 < 2e-16 \*\*\*  
## cholesterol3 1.092e+00 3.764e-02 29.011 < 2e-16 \*\*\*  
## glucose2 5.601e-03 3.844e-02 0.146 0.884143   
## glucose3 -3.231e-01 4.162e-02 -7.764 8.22e-15 \*\*\*  
## smoke1 -1.733e-01 3.558e-02 -4.871 1.11e-06 \*\*\*  
## alcohol1 -2.210e-01 4.479e-02 -4.934 8.04e-07 \*\*\*  
## active1 -2.371e-01 2.295e-02 -10.331 < 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: 86631 on 62497 degrees of freedom  
## Residual deviance: 70227 on 62485 degrees of freedom  
## AIC: 70253  
##   
## Number of Fisher Scoring iterations: 4

Функция step выбирает наилучшую модель логистической регрессии. В результате переменная gender была исключена из модели. В результате все переменные, кроме glucose level 2, оказались статистически значимыми для предсказания заболевания.