automatization\_notebook\_04

2024-01-14

# Чтение данных

В вашем варианте нужно использовать датасет healthcare-dataset-stroke-data.

data <- read\_csv("data/raw/healthcare-dataset-stroke-data.csv")

## Rows: 5110 Columns: 12  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (6): gender, ever\_married, work\_type, Residence\_type, bmi, smoking\_status  
## dbl (6): id, age, hypertension, heart\_disease, avg\_glucose\_level, stroke  
##   
## i Use `spec()` to retrieve the full column specification for this data.  
## i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# Выведите общее описание данных

summary(data)

## id gender age hypertension   
## Min. : 67 Length:5110 Min. : 0.08 Min. :0.00000   
## 1st Qu.:17741 Class :character 1st Qu.:25.00 1st Qu.:0.00000   
## Median :36932 Mode :character Median :45.00 Median :0.00000   
## Mean :36518 Mean :43.23 Mean :0.09746   
## 3rd Qu.:54682 3rd Qu.:61.00 3rd Qu.:0.00000   
## Max. :72940 Max. :82.00 Max. :1.00000   
## heart\_disease ever\_married work\_type Residence\_type   
## Min. :0.00000 Length:5110 Length:5110 Length:5110   
## 1st Qu.:0.00000 Class :character Class :character Class :character   
## Median :0.00000 Mode :character Mode :character Mode :character   
## Mean :0.05401   
## 3rd Qu.:0.00000   
## Max. :1.00000   
## avg\_glucose\_level bmi smoking\_status stroke   
## Min. : 55.12 Length:5110 Length:5110 Min. :0.00000   
## 1st Qu.: 77.25 Class :character Class :character 1st Qu.:0.00000   
## Median : 91.89 Mode :character Mode :character Median :0.00000   
## Mean :106.15 Mean :0.04873   
## 3rd Qu.:114.09 3rd Qu.:0.00000   
## Max. :271.74 Max. :1.00000

# 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**: Пропущенные значения есть только по переменным bmi (N/A) и smoking\_status (“Unknown”). Таким образом, удаление субъектов с пропущенными значениями нерационально (пропущено не более 2 значений в каждой строке). Удалены переменные с количеством пропущенных значений более 20%.

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”.

# Задания 1,2,3,4,6  
tibble(col\_names = colnames(data), missed = colSums(is.na(data)))

## # A tibble: 12 x 2  
## col\_names missed  
## <chr> <dbl>  
## 1 id 0  
## 2 gender 0  
## 3 age 0  
## 4 hypertension 0  
## 5 heart\_disease 0  
## 6 ever\_married 0  
## 7 work\_type 0  
## 8 Residence\_type 0  
## 9 avg\_glucose\_level 0  
## 10 bmi 0  
## 11 smoking\_status 0  
## 12 stroke 0

# В таблице отсутствуют пропущенные значения, обозначенные как "NA". пробуем посмотреть, какие другие обозначения могли использоваться для пропущенных значений.  
  
data %>%  
 select(id) %>%  
 distinct() %>%  
 arrange(desc(id))

## # A tibble: 5,110 x 1  
## id  
## <dbl>  
## 1 72940  
## 2 72918  
## 3 72915  
## 4 72914  
## 5 72911  
## 6 72882  
## 7 72867  
## 8 72861  
## 9 72836  
## 10 72824  
## # i 5,100 more rows

data %>%  
 select(gender) %>%  
 distinct() %>%  
 arrange(desc(gender))

## # A tibble: 3 x 1  
## gender  
## <chr>   
## 1 Other   
## 2 Male   
## 3 Female

data %>%  
 select(age) %>%  
 distinct() %>%  
 arrange(desc(age))

## # A tibble: 104 x 1  
## age  
## <dbl>  
## 1 82  
## 2 81  
## 3 80  
## 4 79  
## 5 78  
## 6 77  
## 7 76  
## 8 75  
## 9 74  
## 10 73  
## # i 94 more rows

data %>%  
 select(hypertension) %>%  
 distinct() %>%  
 arrange(desc(hypertension))

## # A tibble: 2 x 1  
## hypertension  
## <dbl>  
## 1 1  
## 2 0

data %>%  
 select(heart\_disease) %>%  
 distinct() %>%  
 arrange(desc(heart\_disease))

## # A tibble: 2 x 1  
## heart\_disease  
## <dbl>  
## 1 1  
## 2 0

data %>%  
 select(ever\_married) %>%  
 distinct() %>%  
 arrange(desc(ever\_married))

## # A tibble: 2 x 1  
## ever\_married  
## <chr>   
## 1 Yes   
## 2 No

data %>%  
 select(work\_type) %>%  
 distinct() %>%  
 arrange(desc(work\_type))

## # A tibble: 5 x 1  
## work\_type   
## <chr>   
## 1 children   
## 2 Self-employed  
## 3 Private   
## 4 Never\_worked   
## 5 Govt\_job

data %>%  
 select(Residence\_type) %>%  
 distinct() %>%  
 arrange(desc(Residence\_type))

## # A tibble: 2 x 1  
## Residence\_type  
## <chr>   
## 1 Urban   
## 2 Rural

data %>%  
 select(avg\_glucose\_level) %>%  
 distinct() %>%  
 arrange(desc(avg\_glucose\_level))

## # A tibble: 3,979 x 1  
## avg\_glucose\_level  
## <dbl>  
## 1 272.  
## 2 268.  
## 3 268.  
## 4 268.  
## 5 267.  
## 6 264.  
## 7 263.  
## 8 262.  
## 9 261.  
## 10 260.  
## # i 3,969 more rows

data %>%  
 select(bmi) %>%  
 distinct() %>%  
 arrange(desc(bmi))

## # A tibble: 419 x 1  
## bmi   
## <chr>  
## 1 N/A   
## 2 97.6   
## 3 92   
## 4 78   
## 5 71.9   
## 6 66.8   
## 7 64.8   
## 8 64.4   
## 9 63.3   
## 10 61.6   
## # i 409 more rows

data %>%  
 select(smoking\_status) %>%  
 distinct() %>%  
 arrange(desc(smoking\_status))

## # A tibble: 4 x 1  
## smoking\_status   
## <chr>   
## 1 smokes   
## 2 never smoked   
## 3 formerly smoked  
## 4 Unknown

data %>%  
 select(stroke) %>%  
 distinct() %>%  
 arrange(desc(stroke))

## # A tibble: 2 x 1  
## stroke  
## <dbl>  
## 1 1  
## 2 0

# Пропущенные значения есть только по переменным bmi (N/A) и smoking\_status ("Unknown"). Таким образом, удаление субъектов с пропущенными значениями нерационально (пропущено не более 2 значений в каждой строке). Проверяем обе указанные переменные на количество пропущенных значений и удаляем, если их более 20%.  
  
cleaned\_data <- data %>%  
 mutate (across(where(function(x) sum(x == "N/A") > length(x) \* 0.2 | sum(x == "Unknown") > length(x) \* 0.2), function(x) x = NULL)) %>% # Переменная status\_smoking была удалена.  
 rename(ID = id, Gender = gender, Age = age, Hypertension = hypertension, Cardiovascular\_diseases = heart\_disease, Ever\_married = ever\_married, Work\_type = work\_type, Average\_glucose\_level = avg\_glucose\_level, BMI = bmi, Stroke = stroke) %>% # Пробелы в названиях оставлены в варианте нижнего подчеркивания для снижения вероятности ошибок при обращении к переменным. Предложенный вариант на английском языке достаточно читаемый, на русский язык переименование не проводилось  
 mutate(across(c(ID, Age, Average\_glucose\_level, BMI), function(x) as.character(x) %>% na\_if(., "N/A") %>% as.numeric())) %>% # Приведение переменных к типу numeric и замена значений "N/A" в переменной BMI на NA  
 mutate(across(c(Gender, Hypertension, Cardiovascular\_diseases, Ever\_married, Work\_type, Residence\_type, Stroke), function(x) as.factor(x))) %>% # Приведение переменных к типу factor  
 arrange(desc(Age)) # Сортировка по возрасту  
  
# Задание 5  
outliers <- cleaned\_data %>%   
 filter(!between(Age, mean(Age, na.rm = TRUE) - 3 \* sd(Age, na.rm = TRUE), mean(Age, na.rm = TRUE) + 3 \* sd(Age, na.rm = TRUE)) | !between(Average\_glucose\_level, mean(Average\_glucose\_level, na.rm = TRUE) - 3 \* sd(Average\_glucose\_level, na.rm = TRUE), mean(Average\_glucose\_level, na.rm = TRUE) + 3 \* sd(Average\_glucose\_level, na.rm = TRUE)) | !between(BMI, mean(BMI, na.rm = TRUE) - 3 \* sd(BMI, na.rm = TRUE), mean(BMI, na.rm = TRUE) + 3 \* sd(BMI, na.rm = TRUE)))  
  
readr::write\_excel\_csv(outliers, "data/raw/outliers.csv")

# Сколько осталось переменных?

ncol(cleaned\_data)

## [1] 11

# Сколько осталось случаев?

nrow(cleaned\_data)

## [1] 5110

# Есть ли в данных идентичные строки?

ifelse(nrow(cleaned\_data) != nrow(cleaned\_data %>% distinct()), "Есть", "Нет")

## [1] "Нет"

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

tibble(col\_names = colnames(cleaned\_data), missed = colSums(is.na(cleaned\_data))) %>%  
 filter(missed != 0) %>%  
 print() %>% # Таблица с переменными, в которых есть проущенные значения и указанием количества пропущенных значений в каждой переменной  
 nrow() # Количество переменных с пропущенными значениями

## # A tibble: 1 x 2  
## col\_names missed  
## <chr> <dbl>  
## 1 BMI 201

## [1] 1

# Описательные статистики

## Количественные переменные

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

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(  
 `\_number\_of\_values` = ~sum(!is.na(.x)) %>% as.character(),  
 `\_number\_of\_missed\_values` = ~sum(is.na(.x)) %>% as.character(),  
 `\_mean` = ~ifelse(sum(!is.na(.x)) == 0, "no data", mean(.x, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_median` = ~ifelse(sum(!is.na(.x)) == 0, "no data", median(.x, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_sd` = ~ifelse(sum(!is.na(.x)) == 0, "no data", sd(.x, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_Q1 - Q3` = ~ifelse(sum(!is.na(.x)) == 0, "no data", paste0(quantile(.x, 0.25, na.rm = TRUE) %>% round(2), " - ", quantile(.x, 0.75, na.rm = TRUE))),  
 `\_interquartile\_range` = ~ifelse(sum(!is.na(.x)) == 0, "no data", quantile(.x, 0.75, na.rm = TRUE) - quantile(.x, 0.25, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_min` = ~ifelse(sum(!is.na(.x)) == 0, "no data", min(.x, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_max` = ~ifelse(sum(!is.na(.x)) == 0, "no data", max(.x, na.rm = TRUE)) %>% round(2) %>% as.character(),  
 `\_95%\_CI\_for\_mean` = ~ifelse(sum(!is.na(.x)) == 0, "no data", paste0(DescTools::MeanCI(.x, method = "classic", conf.level = 0.95, sides = "two.sided", na.rm = TRUE)[2] %>% as.numeric() %>% round(2) %>% as.character(), " - ", DescTools::MeanCI(.x, method = "classic", sides = "two.sided", na.rm = TRUE)[3] %>% as.numeric() %>% round(2) %>% as.character()))  
 )  
   
cleaned\_data %>%  
 select(Stroke, where(is.numeric), -ID) %>%  
 group\_by(Stroke) %>%  
 summarise(across(where(is.numeric), statistics)) %>%  
 pivot\_longer(!Stroke) %>%  
 separate(name, into = c("variable", "Statistics"), sep = "\_\_")

## # A tibble: 60 x 4  
## Stroke variable Statistics value   
## <fct> <chr> <chr> <chr>   
## 1 0 Age number\_of\_values 4861   
## 2 0 Age number\_of\_missed\_values 0   
## 3 0 Age mean 41.97   
## 4 0 Age median 43   
## 5 0 Age sd 22.29   
## 6 0 Age Q1 - Q3 24 - 59   
## 7 0 Age interquartile\_range 35   
## 8 0 Age min 0.08   
## 9 0 Age max 82   
## 10 0 Age 95%\_CI\_for\_mean 41.34 - 42.6  
## # i 50 more rows

## Категориальные переменные

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

1.1) Абсолютное количество;

1.2) Относительное количество внутри группы;

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

cleaned\_data %>%  
 select(Stroke, Gender) %>%  
 count(Stroke, Gender) %>%  
 group\_by(Stroke) %>%  
 mutate(Relative\_quantity\_in\_group = (n / sum(n)) %>% round(4) %>% `\*` (100) %>% str\_c("%")) %>%   
 ungroup()

## # A tibble: 5 x 4  
## Stroke Gender n Relative\_quantity\_in\_group  
## <fct> <fct> <int> <chr>   
## 1 0 Female 2853 58.69%   
## 2 0 Male 2007 41.29%   
## 3 0 Other 1 0.02%   
## 4 1 Female 141 56.63%   
## 5 1 Male 108 43.37%

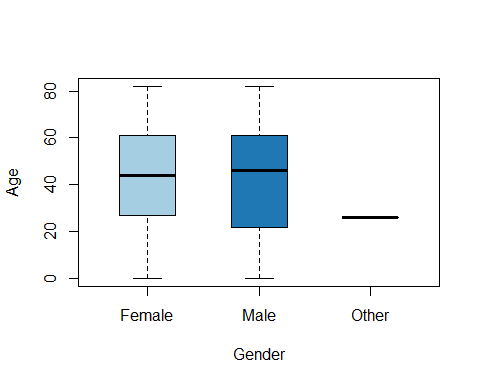
# Строку кода для 95% ДИ не получилось заставить корректно работать:  
# mutate(`95%\_CI` = paste0(DescTools::BinomCI(n/sum(n), 1, method = "clopper-pearson", conf.level = 0.95, sides = "two.sided")[2] %>% as.numeric() %>% round(2) %>% as.character(), " - ", DescTools::BinomCI(n/sum(n), 1, method = "clopper-pearson", conf.level = 0.95, sides = "two.sided")[3] %>% as.numeric() %>% round(2) %>% as.character())) %>%

# Визуализация

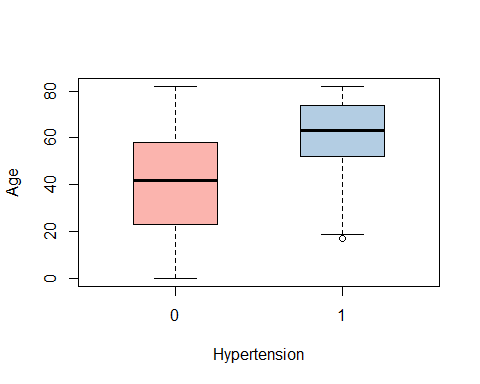
## Количественные переменные

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.

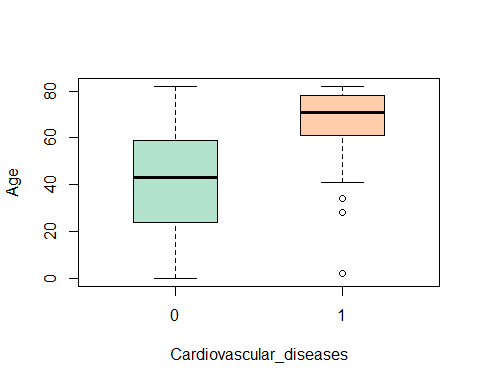
# Для переменной Age  
mypalette <- brewer.pal(4, "Paired")  
boxplot(Age ~ Gender, cleaned\_data, boxwex = 0.5, col = mypalette)



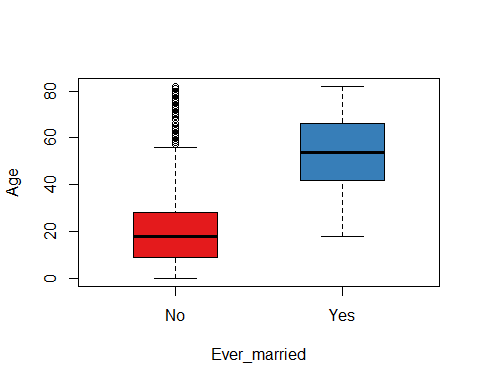
mypalette1 <- brewer.pal(4, "Pastel1")  
boxplot(Age ~ Hypertension, cleaned\_data, boxwex = 0.5, col = mypalette1)



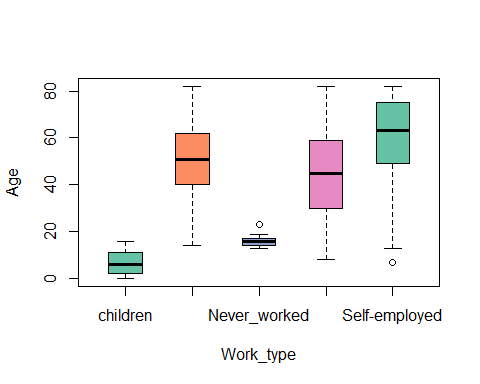
mypalette2 <- brewer.pal(4, "Pastel2")  
boxplot(Age ~ Cardiovascular\_diseases, cleaned\_data, boxwex = 0.5, col = mypalette2)



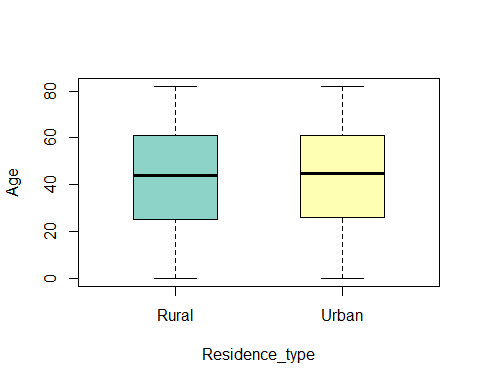
mypalette3 <- brewer.pal(4, "Set1")  
boxplot(Age ~ Ever\_married, cleaned\_data, boxwex = 0.5, col = mypalette3)



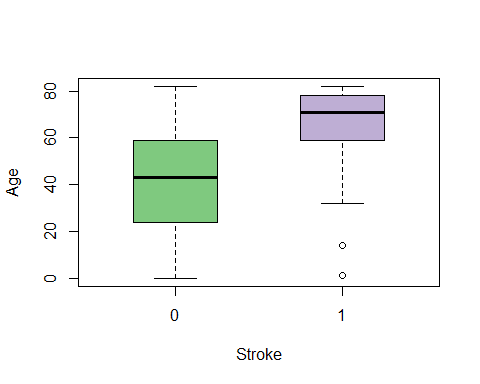
mypalette4 <- brewer.pal(4, "Set2")  
boxplot(Age ~ Work\_type, cleaned\_data, boxwex = 0.5, col = mypalette4)



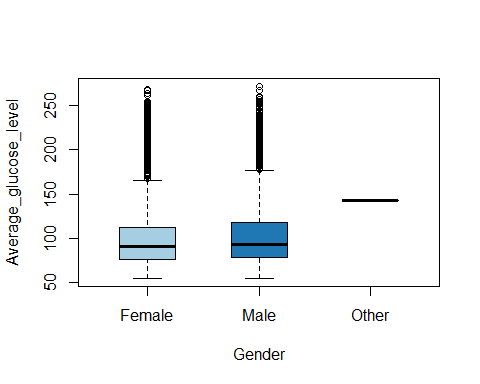
mypalette5 <- brewer.pal(4, "Set3")  
boxplot(Age ~ Residence\_type, cleaned\_data, boxwex = 0.5, col = mypalette5)



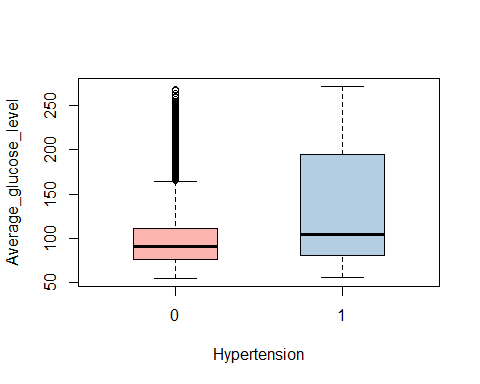
mypalette6 <- brewer.pal(4, "Accent")  
boxplot(Age ~ Stroke, cleaned\_data, boxwex = 0.5, col = mypalette6)



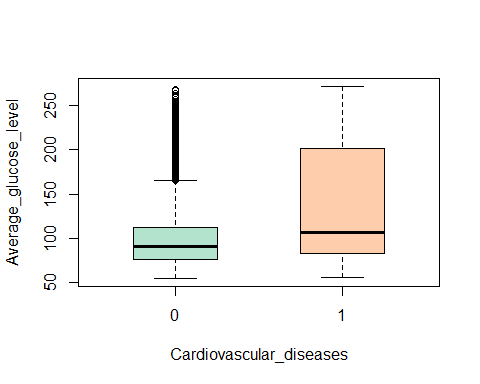
# Для переменной Average\_glucose\_level  
mypalette <- brewer.pal(4, "Paired")  
boxplot(Average\_glucose\_level ~ Gender, cleaned\_data, boxwex = 0.5, col = mypalette)



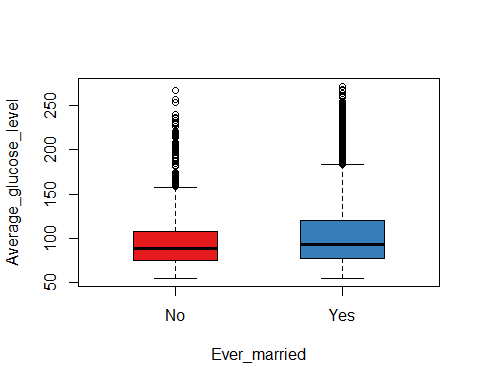
mypalette1 <- brewer.pal(4, "Pastel1")  
boxplot(Average\_glucose\_level ~ Hypertension, cleaned\_data, boxwex = 0.5, col = mypalette1)



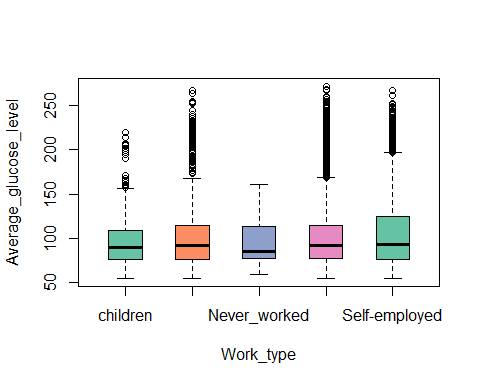
mypalette2 <- brewer.pal(4, "Pastel2")  
boxplot(Average\_glucose\_level ~ Cardiovascular\_diseases, cleaned\_data, boxwex = 0.5, col = mypalette2)



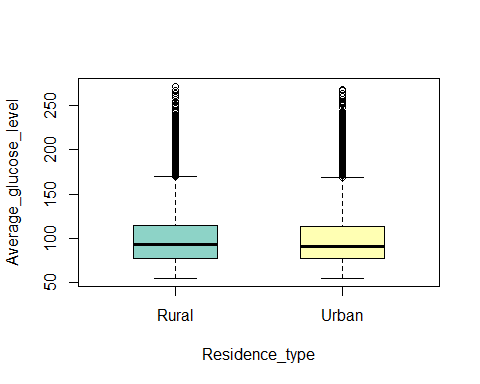
mypalette3 <- brewer.pal(4, "Set1")  
boxplot(Average\_glucose\_level ~ Ever\_married, cleaned\_data, boxwex = 0.5, col = mypalette3)



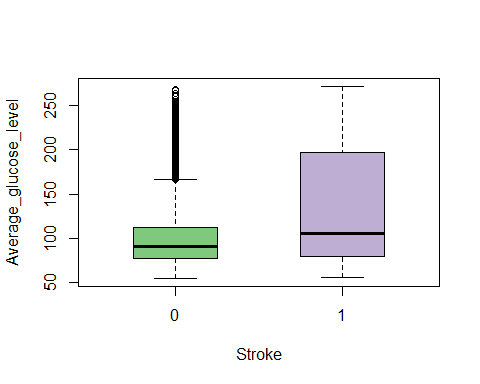
mypalette4 <- brewer.pal(4, "Set2")  
boxplot(Average\_glucose\_level ~ Work\_type, cleaned\_data, boxwex = 0.5, col = mypalette4)



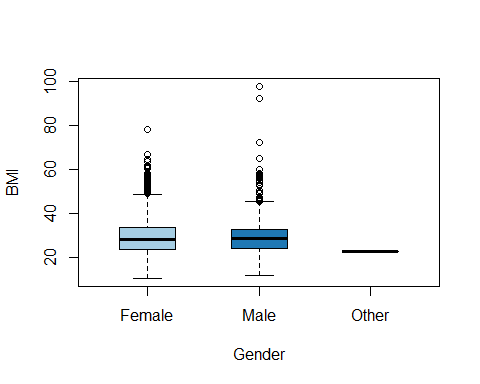
mypalette5 <- brewer.pal(4, "Set3")  
boxplot(Average\_glucose\_level ~ Residence\_type, cleaned\_data, boxwex = 0.5, col = mypalette5)



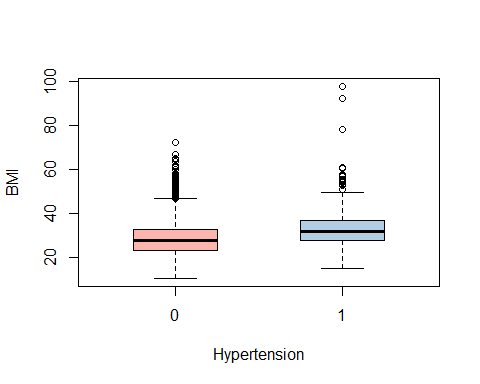
mypalette6 <- brewer.pal(4, "Accent")  
boxplot(Average\_glucose\_level ~ Stroke, cleaned\_data, boxwex = 0.5, col = mypalette6)



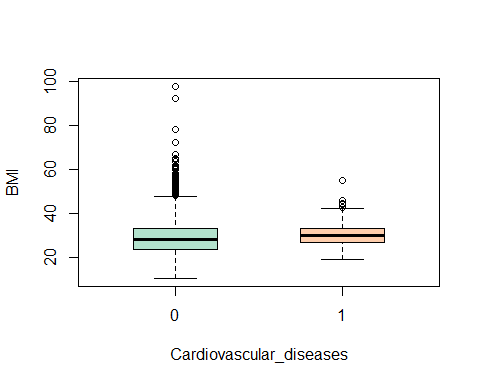
# Для переменной BMI  
mypalette <- brewer.pal(4, "Paired")  
boxplot(BMI ~ Gender, cleaned\_data, boxwex = 0.5, col = mypalette)



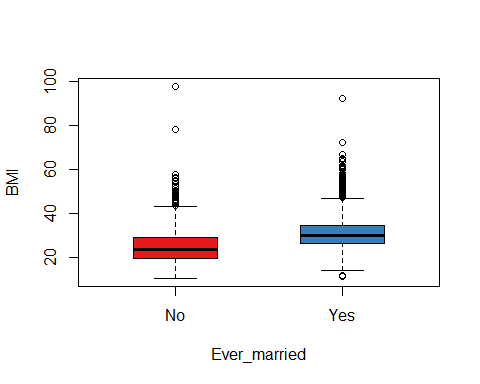
mypalette1 <- brewer.pal(4, "Pastel1")  
boxplot(BMI ~ Hypertension, cleaned\_data, boxwex = 0.5, col = mypalette1)



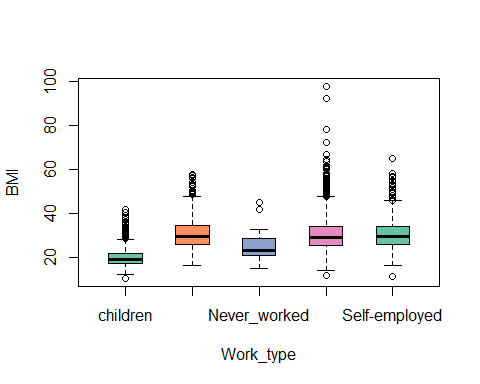
mypalette2 <- brewer.pal(4, "Pastel2")  
boxplot(BMI ~ Cardiovascular\_diseases, cleaned\_data, boxwex = 0.5, col = mypalette2)



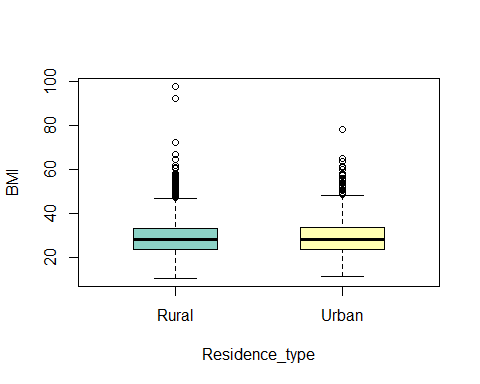
mypalette3 <- brewer.pal(4, "Set1")  
boxplot(BMI ~ Ever\_married, cleaned\_data, boxwex = 0.5, col = mypalette3)



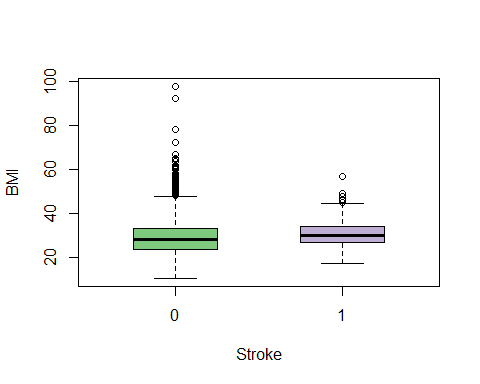
mypalette4 <- brewer.pal(4, "Set2")  
boxplot(BMI ~ Work\_type, cleaned\_data, boxwex = 0.5, col = mypalette4)



mypalette5 <- brewer.pal(4, "Set3")  
boxplot(BMI ~ Residence\_type, cleaned\_data, boxwex = 0.5, col = mypalette5)



mypalette6 <- brewer.pal(4, "Accent")  
boxplot(BMI ~ Stroke, cleaned\_data, boxwex = 0.5, col = mypalette6)



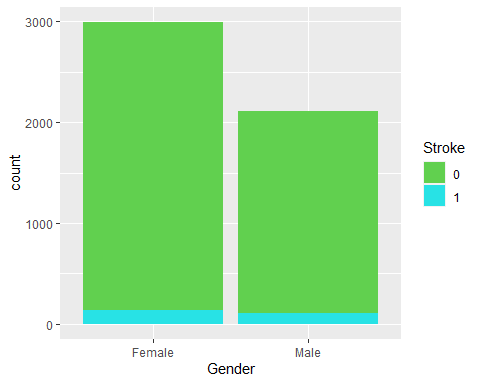
# ggplot(data = cleaned\_data, mapping = aes(x = Stroke, y = Age)) +  
# geom\_boxplot() +   
# geom\_beeswarm(size = 0.5, cex = 0.4, shape = 21, stroke = 0.1) +  
# theme\_bw()

## Категориальные переменные

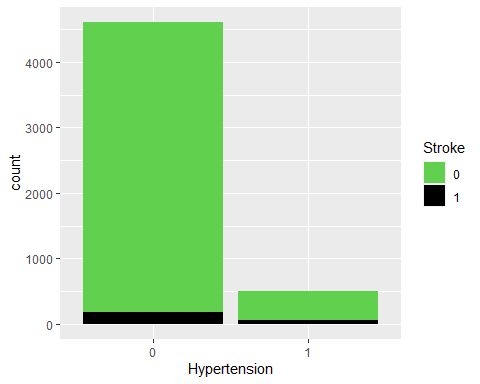
1. Сделайте подходящие визуализации категориальных переменных. Обоснуйте, почему выбрали именно этот тип.

Категориальные переменные хорошо описываются гистограммами, мозичными сюжетами и блочными диаграммами. В данном случае и распределение каждой переменной и частоты по 2 переменным на одном графике (например, с учетом переменной Stroke) удобно посмотреть по гистограммам в geom\_bar.

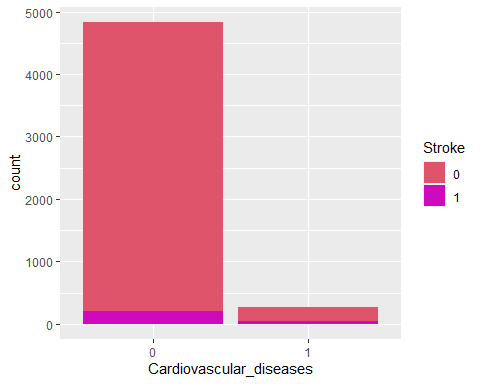
cleaned\_data %>%   
 filter(Gender != "Other") %>%  
 ggplot() +  
 geom\_bar(aes(x = Gender, fill = Stroke)) +  
 scale\_fill\_manual(values = c("99CC00", "445FAA"))



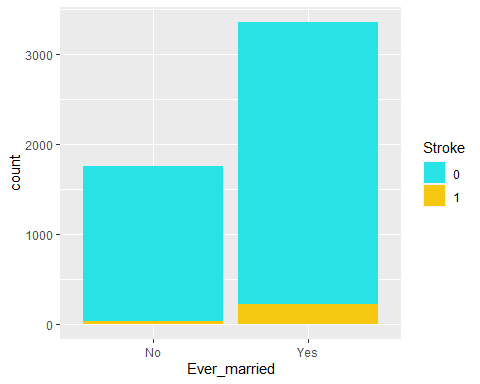
cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Hypertension, fill = Stroke)) +  
 scale\_fill\_manual(values = c("99AD00", "1E55AA"))



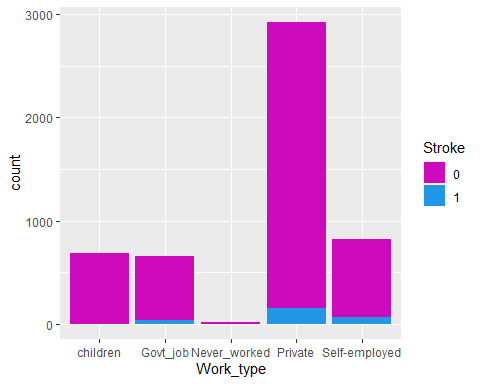
cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Cardiovascular\_diseases, fill = Stroke)) +  
 scale\_fill\_manual(values = c("994E00", "334CAA"))



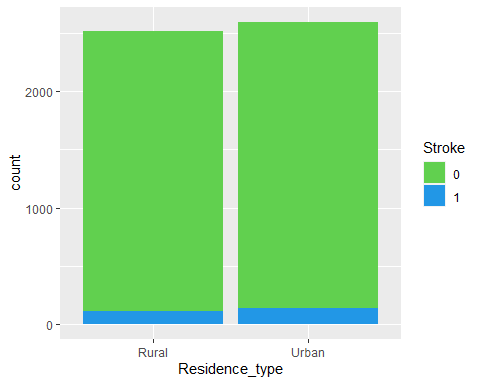
cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Ever\_married, fill = Stroke)) +  
 scale\_fill\_manual(values = c("93EFF0", "495F0A"))



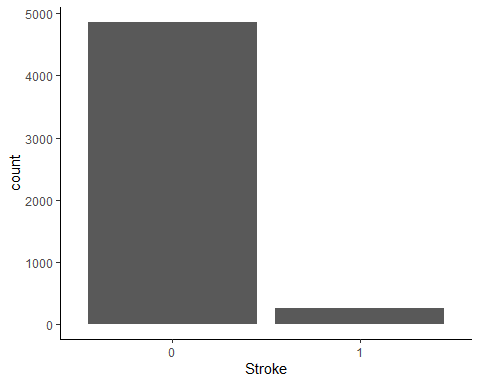
cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Work\_type, fill = Stroke)) +  
 scale\_fill\_manual(values = c("78DC00", "4CCFAA"))



cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Residence\_type, fill = Stroke)) +  
 scale\_fill\_manual(values = c("99CF00", "4BBFAA"))



cleaned\_data %>%   
 ggplot() +  
 geom\_bar(aes(x = Stroke)) +  
 theme\_classic()



# 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? Для всех количественных переменных p-value менее 0.05, что позволяет отвергнуть нулевые гипотезы, что распределение этих переменных не отличается от нормального.

# Поскольку функция теста Шапиро-Уилка не разрешает задавать на проверку векторы более чем из 5000 значений, пробуем создать серию выборок по 5000 значений из каждой количественной переменной и в каждой выборке проверить нормальность распределения, подсчитав количество результатов с p-value менее 0.05:   
  
# Для переменной Age  
df1 <- data.frame(n\_test = 1:100, `p-value` = NA)  
for(i in 1:100) {df1$p[i] = shapiro.test(sample(cleaned\_data$Age, 5000))[2]}  
print(paste0("Age: p-value = ", sum(df1$p < 0.05))) # Все значения p-value оказались менее 0.05 - распределение переменной Age отлично от нормального.

## [1] "Age: p-value = 100"

# Можно также попробовать убрать выбросы (по правилу 3 сигм остается более 5000 значений, поэтому пробуем другой вариант поиска выбросов - менее quantile(0.25) - 1.5 \* IQR или более quantile(0.75) + 1.5 \* IQR:  
n\_outliers <- cleaned\_data %>%  
 filter(between(Age, quantile(Age, 0.25, na.rm = TRUE) - 1.5 \* IQR(Age, na.rm = TRUE), quantile(Age, 0.75, na.rm = TRUE) + 1.5 \* IQR(Age, na.rm = TRUE)) & between(Average\_glucose\_level, quantile(Average\_glucose\_level, 0.25, na.rm = TRUE) - 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE), quantile(Average\_glucose\_level, 0.75, na.rm = TRUE) + 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE)) & between(BMI, quantile(Age, 0.25, na.rm = TRUE) - 1.5 \* IQR(BMI, na.rm = TRUE), quantile(BMI, 0.75, na.rm = TRUE) + 1.5 \* IQR(BMI, na.rm = TRUE)))  
  
shapiro.test(n\_outliers$Age) # Значение p-value менее 0.05 - распределение переменной Age отлично от нормального.

##   
## Shapiro-Wilk normality test  
##   
## data: n\_outliers$Age  
## W = 0.97031, p-value < 2.2e-16

# Для переменной Average\_glucose\_level  
df2 <- data.frame(n\_test = 1:100, `p-value` = NA)  
for(i in 1:100) {df2$p[i] = shapiro.test(sample(cleaned\_data$Average\_glucose\_level, 5000))[2]}  
print(paste0("Average\_glucose\_rate: p-value = ", sum(df2$p < 0.05))) # Все значения p-value оказались менее 0.05 - распределение переменной BMI отлично от нормального.

## [1] "Average\_glucose\_rate: p-value = 100"

# Исключение выбросов:  
n\_outliers <- cleaned\_data %>%  
 filter(between(Age, quantile(Age, 0.25, na.rm = TRUE) - 1.5 \* IQR(Age, na.rm = TRUE), quantile(Age, 0.75, na.rm = TRUE) + 1.5 \* IQR(Age, na.rm = TRUE)) & between(Average\_glucose\_level, quantile(Average\_glucose\_level, 0.25, na.rm = TRUE) - 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE), quantile(Average\_glucose\_level, 0.75, na.rm = TRUE) + 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE)) & between(BMI, quantile(BMI, 0.25, na.rm = TRUE) - 1.5 \* IQR(BMI, na.rm = TRUE), quantile(BMI, 0.75, na.rm = TRUE) + 1.5 \* IQR(BMI, na.rm = TRUE)))  
  
shapiro.test(n\_outliers$Average\_glucose\_level) # Значение p-value менее 0.05 - распределение переменной BMI отлично от нормального.

##   
## Shapiro-Wilk normality test  
##   
## data: n\_outliers$Average\_glucose\_level  
## W = 0.95188, p-value < 2.2e-16

# Для переменной BMI  
df3 <- data.frame(n\_test = 1:100, `p-value` = NA)  
for(i in 1:100) {df3$p[i] = shapiro.test(sample(cleaned\_data$BMI, 5000))[2]}  
print(paste0("BMI: p-value = ", sum(df3$p < 0.05))) # Все значения p-value оказались менее 0.05 - распределение переменной BMI отлично от нормального.

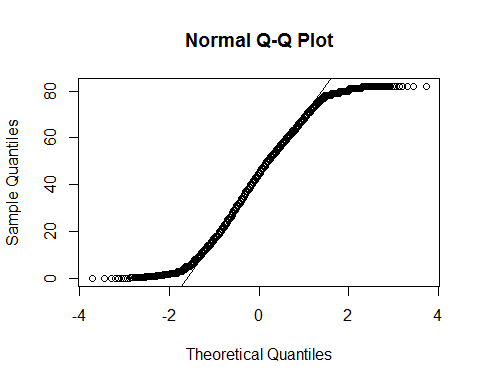
## [1] "BMI: p-value = 100"

# Исключение выбросов:  
n\_outliers <- cleaned\_data %>%  
 filter(between(Age, quantile(Age, 0.25, na.rm = TRUE) - 1.5 \* IQR(Age, na.rm = TRUE), quantile(Age, 0.75, na.rm = TRUE) + 1.5 \* IQR(Age, na.rm = TRUE)) & between(Average\_glucose\_level, quantile(Average\_glucose\_level, 0.25, na.rm = TRUE) - 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE), quantile(Average\_glucose\_level, 0.75, na.rm = TRUE) + 1.5 \* IQR(Average\_glucose\_level, na.rm = TRUE)) & between(BMI, quantile(BMI, 0.25, na.rm = TRUE) - 1.5 \* IQR(BMI, na.rm = TRUE), quantile(BMI, 0.75, na.rm = TRUE) + 1.5 \* IQR(BMI, na.rm = TRUE)))  
  
shapiro.test(n\_outliers$BMI) # Значение p-value менее 0.05 - распределение переменной BMI отлично от нормального.

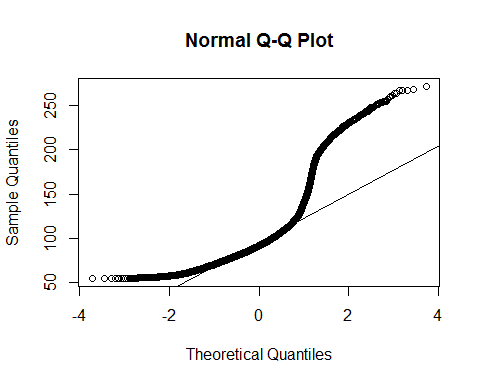
##   
## Shapiro-Wilk normality test  
##   
## data: n\_outliers$BMI  
## W = 0.98846, p-value < 2.2e-16

1. Постройте для каждой количественной переменной QQ-плот. Отличаются ли выводы от теста Шапиро-Уилка? Какой метод вы бы предпочли и почему? Выводы от теста Шапиро-Уилка не отличаются - распределение всех количественных переменных (Age, Average\_glucose\_level, BMI) отличаются от нормальных. Графический метод позволяет проще оценить распределение, но не всегда дает возможность сделать точный вывод (без знания p-value). Также QQ-plot не ограничивает длину переменной.

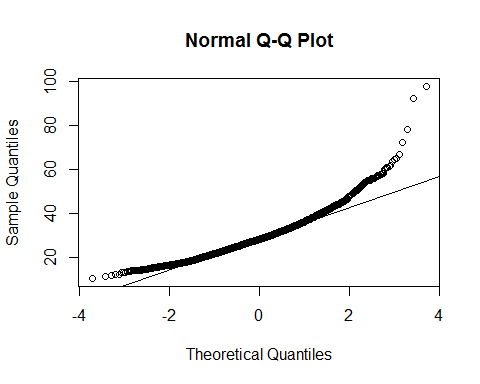
qqnorm(cleaned\_data$Age, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", plot.it = TRUE)  
qqline(cleaned\_data$Age)



qqnorm(cleaned\_data$Average\_glucose\_level, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", plot.it = TRUE)  
qqline(cleaned\_data$Average\_glucose\_level)



qqnorm(cleaned\_data$BMI, xlab = "Theoretical Quantiles", ylab = "Sample Quantiles", plot.it = TRUE)  
qqline(cleaned\_data$BMI)



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

**Напишите текст здесь**

Тест Колмогорова-Смирнова. Ограничения: выборка должна быть была достаточно большой. При сопоставлении двух эмпирических распределений необходимо, чтобы n1,2 ≥ 50. Переменные должны быть упорядочены по нарастанию или убыванию какого-либо признака.

Тест Шапиро-Франсия, тест Андерсона-Дарлинга.

## Сравнение групп

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

# Для количественных переменных Age, Average\_glucose\_level, BMI, с учетом большого количества наблюдений (несколько тысяч для группы Stroke=0 и несколько сотен для группы Stroke=1) целесообразно использовать параметрический метод - асимптотический z-критерий для разности средних (можно рассчитывать на его нормальное распределение). Нулевая гипотеза - H0: 𝜇1 = 𝜇2 - среднее значение переменной (Age, Average\_glucose\_level, BMI) в популяциях лиц с инсультом и без инсульта не различаются.  
  
group\_A <- cleaned\_data$Age[cleaned\_data$Stroke == 0]  
group\_B <- cleaned\_data$Age[cleaned\_data$Stroke == 1]  
z\_test\_Age <- z.test(x = group\_A, y = group\_B, alternative = "two.sided", sigma.x =sd(group\_A), sigma.y = sd(group\_B))  
print("z\_test\_Age: ")

## [1] "z\_test\_Age: "

z\_test\_Age

##   
## Two-sample z-Test  
##   
## data: group\_A and group\_B  
## z = -29.686, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -27.45717 -24.05613  
## sample estimates:  
## mean of x mean of y   
## 41.97154 67.72819

# Группы без инсульта и с инсультом статистически значимо отличаются по возрасту, 95% CI между группами [-27.45; -24.06].  
  
group\_A <- cleaned\_data$Average\_glucose\_level[cleaned\_data$Stroke == 0]  
group\_B <- cleaned\_data$Average\_glucose\_level[cleaned\_data$Stroke == 1]  
z\_test\_Average\_glucose\_level <- z.test(x = group\_A, y = group\_B, alternative = "two.sided", sigma.x =sd(group\_A), sigma.y = sd(group\_B))  
print("z\_test\_Average\_glucose\_level: ")

## [1] "z\_test\_Average\_glucose\_level: "

z\_test\_Average\_glucose\_level

##   
## Two-sample z-Test  
##   
## data: group\_A and group\_B  
## z = -6.9824, p-value = 2.902e-12  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -35.53844 -19.96001  
## sample estimates:  
## mean of x mean of y   
## 104.7955 132.5447

# Группы без инсульта и с инсультом статистически значимо отличаются среднему уровню глюкозы, 95% CI между группами [-35.54; -19.96].   
  
group\_A <- cleaned\_data$BMI[cleaned\_data$Stroke == 0 & !is.na(cleaned\_data$BMI)]  
group\_B <- cleaned\_data$BMI[cleaned\_data$Stroke == 1 & !is.na(cleaned\_data$BMI)]  
z\_test\_BMI <- z.test(x = group\_A, y = group\_B, alternative = "two.sided", sigma.x =sd(group\_A), sigma.y = sd(group\_B))  
print("z\_test\_BMI: ")

## [1] "z\_test\_BMI: "

z\_test\_BMI

##   
## Two-sample z-Test  
##   
## data: group\_A and group\_B  
## z = -3.6404, p-value = 0.0002722  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -2.5356192 -0.7608369  
## sample estimates:  
## mean of x mean of y   
## 28.82306 30.47129

# Группы без инсульта и с инсультом статистически значимо отличаются по ИМТ, 95% CI между группами [-2.56; -0.76].  
  
# Для категориальных переменных (Gender, Hypertension, Cardiovascular\_diseases, Ever\_married, Work\_type, Residence\_type) может быть использован критерий хи-квадрат Пирсона (переменные категориальные, независимые). Нулевая гипотеза - H0: переменная Stroke и переменная Gender (далее, соответственно, Hypertension, Cardiovascular\_diseases, Ever\_married, Work\_type, Residence\_type) независимы. Дополнительно проверим наличие разреженных таблиц сопряжения между выбранными переменными:  
  
  
lapply(X = c("Gender", "Hypertension", "Cardiovascular\_diseases", "Ever\_married", "Work\_type", "Residence\_type"), function(i) {  
 cleaned\_data %>% select(Stroke, all\_of(i)) %>% table()  
 }  
)

## [[1]]  
## Gender  
## Stroke Female Male Other  
## 0 2853 2007 1  
## 1 141 108 0  
##   
## [[2]]  
## Hypertension  
## Stroke 0 1  
## 0 4429 432  
## 1 183 66  
##   
## [[3]]  
## Cardiovascular\_diseases  
## Stroke 0 1  
## 0 4632 229  
## 1 202 47  
##   
## [[4]]  
## Ever\_married  
## Stroke No Yes  
## 0 1728 3133  
## 1 29 220  
##   
## [[5]]  
## Work\_type  
## Stroke children Govt\_job Never\_worked Private Self-employed  
## 0 685 624 22 2776 754  
## 1 2 33 0 149 65  
##   
## [[6]]  
## Residence\_type  
## Stroke Rural Urban  
## 0 2400 2461  
## 1 114 135

# С уччетом полученных значений Для переменной Gender и Work\_type целесообразно применить вариант критерия хи-квадрат с Монте-Карло.  
  
lapply(X = c("Hypertension", "Cardiovascular\_diseases", "Ever\_married", "Residence\_type"), function(i) {  
 cleaned\_data %>%   
 select(Stroke, all\_of(i)) %>%  
 table %>%  
 chisq.test(.) %>%  
 broom::tidy() %>%  
 mutate(var = paste0("Stroke vs ", i), .before = "statistic")  
 }  
)

## [[1]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <int> <chr>   
## 1 Stroke vs Hypertension 81.6 1.66e-19 1 Pearson's Chi-squared tes~  
##   
## [[2]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <int> <chr>   
## 1 Stroke vs Cardiovascular\_diseases 90.3 2.09e-21 1 Pearson's Chi-~  
##   
## [[3]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <int> <chr>   
## 1 Stroke vs Ever\_married 58.9 1.64e-14 1 Pearson's Chi-squared tes~  
##   
## [[4]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <int> <chr>   
## 1 Stroke vs Residence\_type 1.08 0.298 1 Pearson's Chi-squared te~

lapply(X = c("Gender", "Work\_type"), function(i) {  
 cleaned\_data %>%   
 select(Stroke, all\_of(i)) %>%  
 table %>%  
 chisq.test(., simulate.p.value = TRUE, B = 1000) %>%  
 broom::tidy() %>%  
 mutate(var = paste0("Stroke vs ", i), .before = "statistic")  
 }  
)

## [[1]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <lgl> <chr>   
## 1 Stroke vs Gender 0.473 0.578 NA "Pearson's Chi-squared test with~  
##   
## [[2]]  
## # A tibble: 1 x 5  
## var statistic p.value parameter method   
## <chr> <dbl> <dbl> <lgl> <chr>   
## 1 Stroke vs Work\_type 49.2 0.000999 NA "Pearson's Chi-squared test ~

# Мы можем отвергнуть нулевую гипотезу о независимости переменных Stroke и Hypertension, Stroke и Cardiovascular\_diseases, Stroke и Ever\_married, Stroke и Work\_type.

# 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.

## Моделирование

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