

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Национальный исследовательский ядерный университет «МИФИ»  
**Обнинский институт атомной энергетики –**  
филиал федерального государственного автономного образовательного учреждения высшего  
образования «Национальный исследовательский ядерный университет «МИФИ»  
**(ИАТЭ НИЯУ МИФИ)**

Отделение Интеллектуальные кибернетические системы

Отчёт по лабораторной работе №3  
По курсу: «Информационные системы и технологии»

Выполнил:  
студент гр. ИС-М18  
Ермаков В.Е.

Проверил:  
д.т.н., профессор  
Сальников Н.Л.

**Обнинск 2018**

## **1 Задание**

Найти и описать средства визуализации данных и построить различные графики (временной ряд, гистограмма, тепловая карта и т.д.). Средства: библиотеки java, js, python, R и другие варианты со свободной лицензией.

## **2 Цель работы**

Найти удобное средство визуализации данных. Научиться им пользоваться и научить других.

### 3 Ход работы

В процесс поиска подходящего инструмента и языка я остановился на языке программирования R и пакете *ggplot2*.

В качестве примера создадим таблицу *example* с двумя столбцами. В первом столбце будут храниться коды уровней гипотетического фактора Factor (три уровня: A, B, и C). Для каждого из этих уровней сгенерируем по 300 значений.

```
example = data.frame(  
  factor = rep(c("A", "B", "C"), each = 300),  
  var = c(rep(1:10, 30),  
          rep(1:25, 12),  
          rep(1:5, 60))  
)
```

#### 3.1 График распределения плотности

Для начала построим простой график распределения плотности вероятности (см. рис 1).

```
p1 = ggplot(example, aes(x = var, fill = factor)) +  
  geom_density()
```

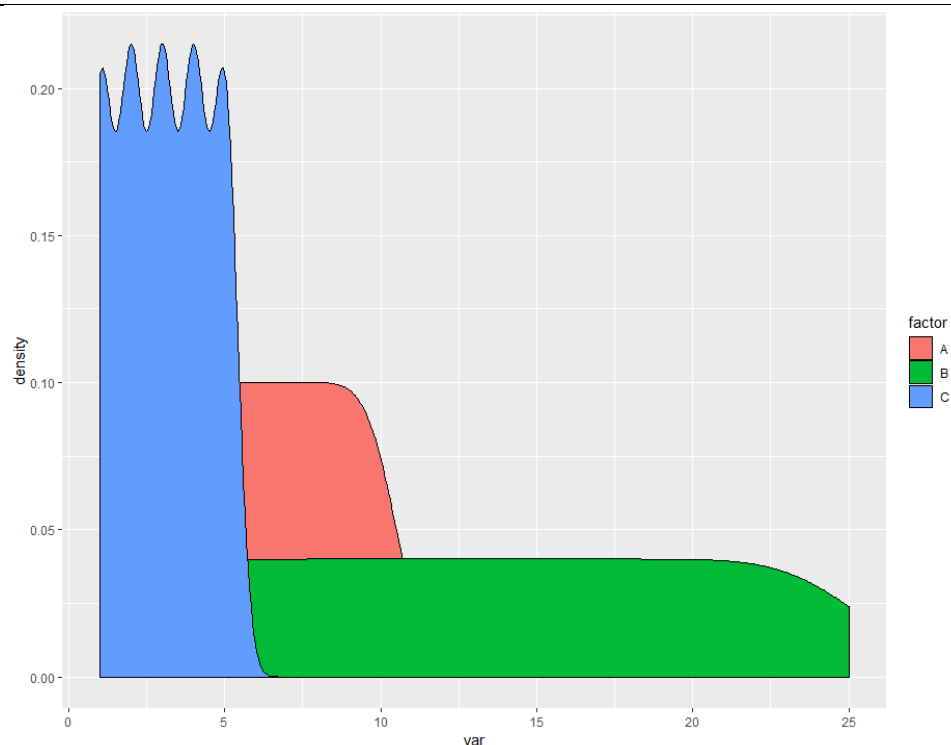


Рисунок 1. Обычный график распределения плотности

Добавим прозрачности с помощью параметра *alpha*. Результат на рисунке 2.

```
p2 = ggplot(example, aes(x = var, fill = factor)) +  
  geom_density(alpha = 1/2)
```

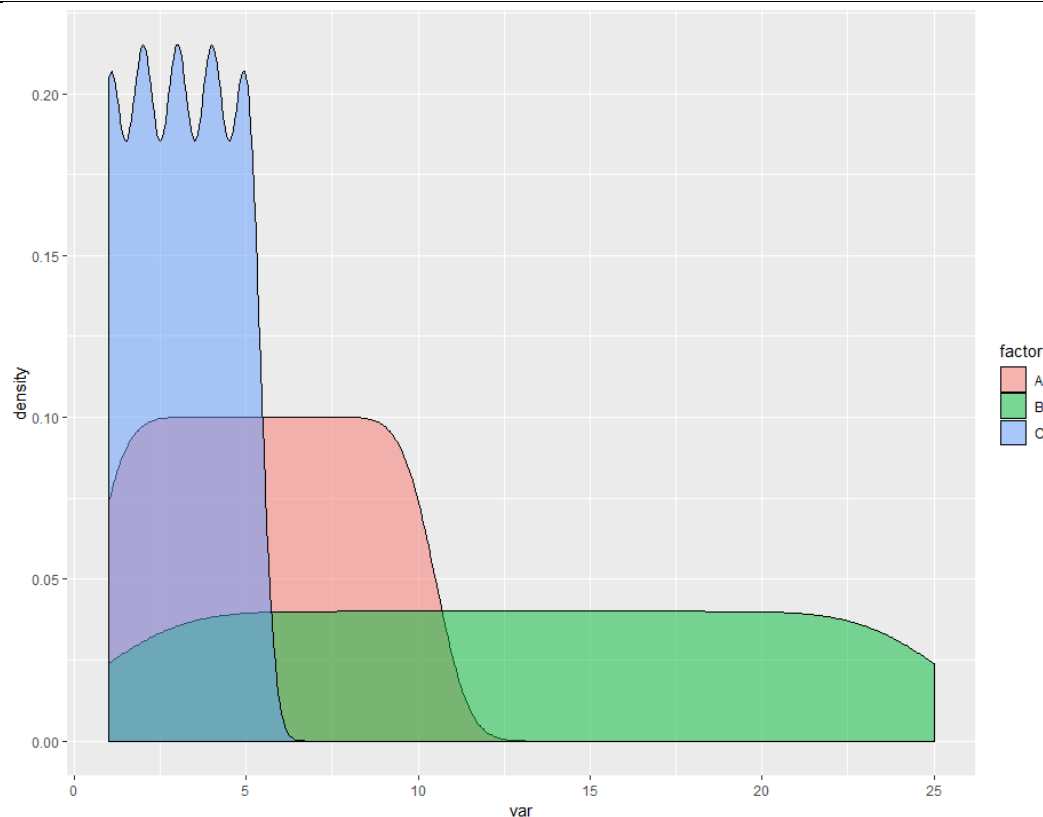


Рисунок 2. График распределения плотности с прозрачностью

Если хотим, чтобы графики наложились друг на друга (сложили свои значения), то воспользуемся параметром *position* в значении "stack" (см. рисунок 3).

```
p3 = ggplot(example, aes(x = var, fill = factor)) +  
  geom_density(alpha = 1/2, position = "stack")
```

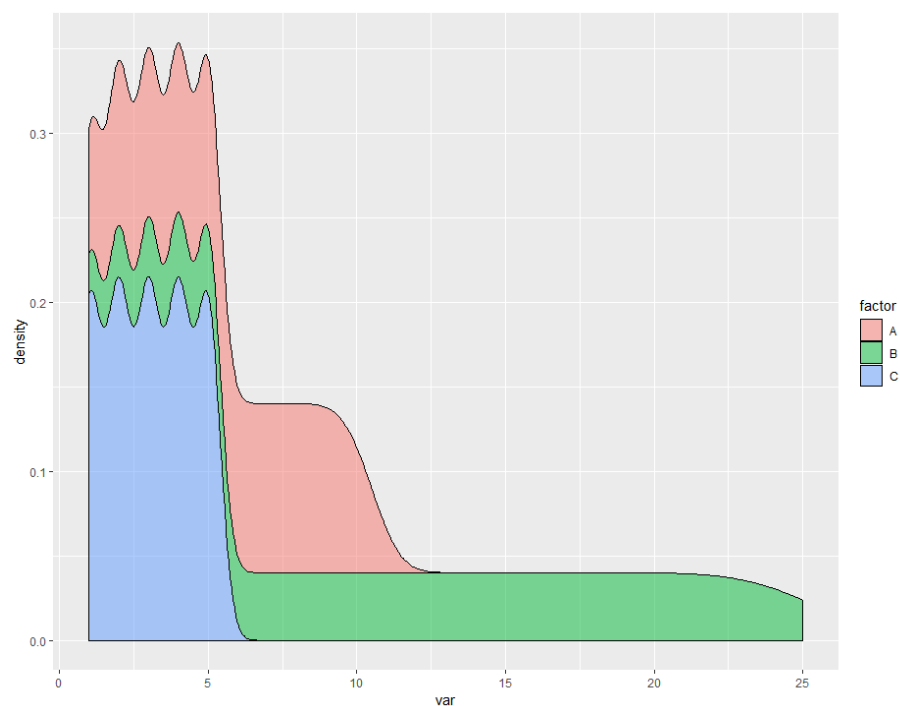


Рисунок 3. График распределения плотности с прозрачностью и наложением

Если хотим, чтобы графики наложились друг на друга и заполнили пространство графика, то для параметра *position* вместо "stack" нужно указать "fill"). Смотрите пример на рисунке 4.

```
p4 = ggplot(example, aes(x = var, fill = factor)) +  
  geom_density(position = "fill")
```

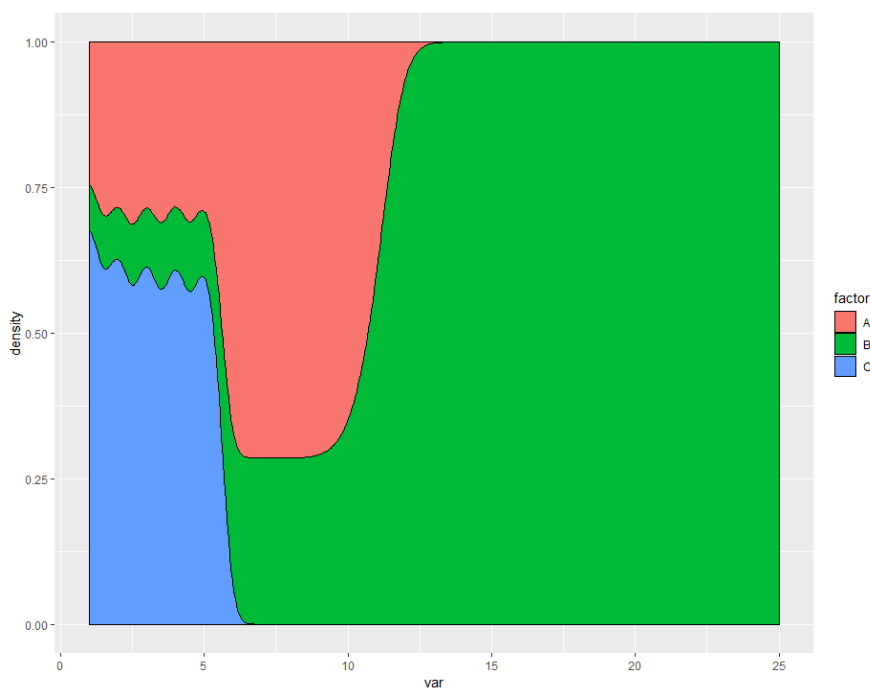


Рисунок 4. График распределения плотности с прозрачностью и наложением

А вот так легко с помощью пакета *gridExtra* можно объединить несколько графиков на одном холсте (см. рисунок 5).

```
library(gridExtra)
grid.arrange(p1, p2, p3, p4)
```

Результат смотрите на рисунке 5.

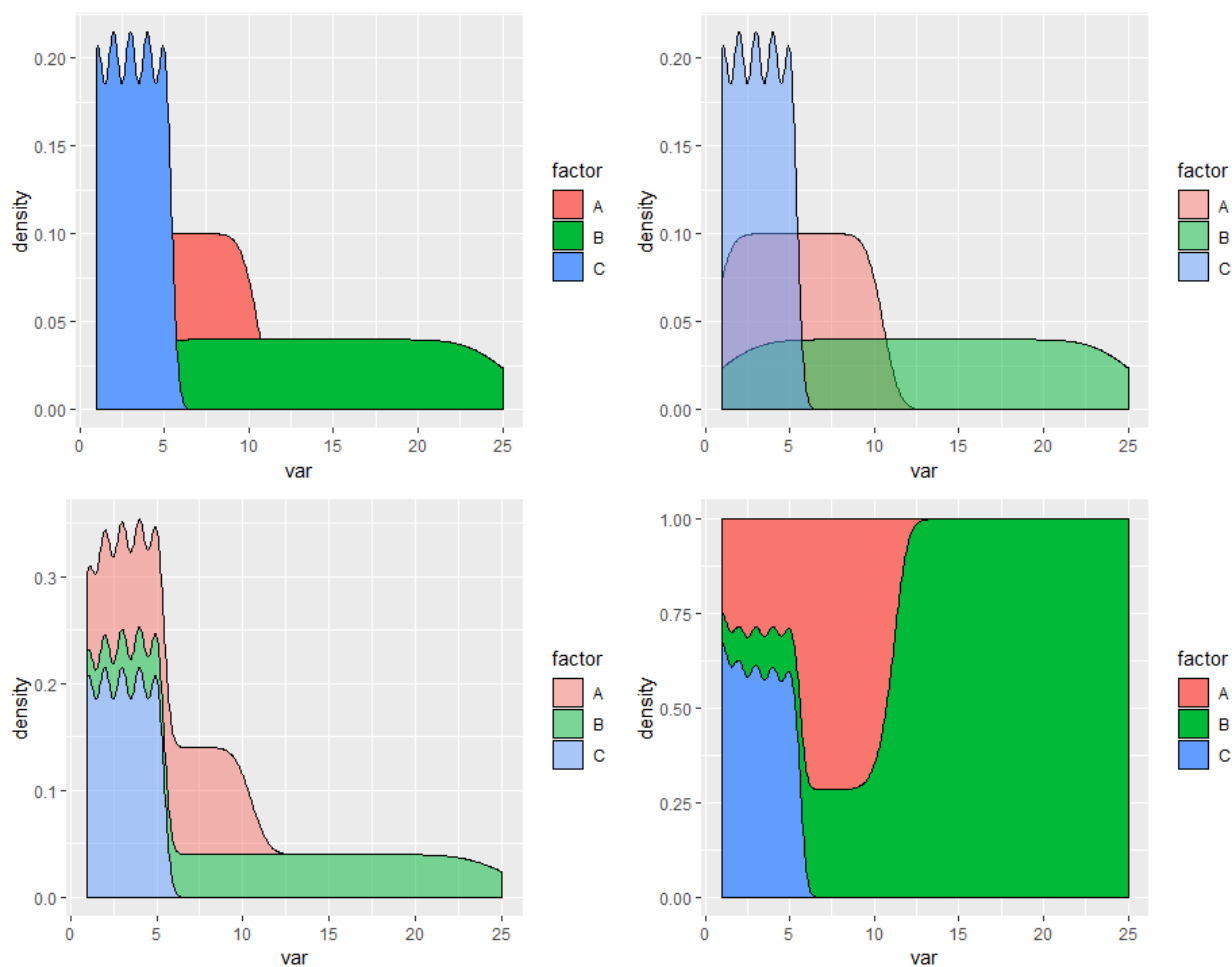


Рисунок 5. Объединение всех четырёх графиков

### 3.2 Box plot

Перед любыми статистическими расчётами полезно выполнить разведочный анализ данных и "рассмотреть" структуру данных на графиках.

Пример ниже нарисует box-plot диаграмму (см. рисунок 6).

```
ggplot(data = example, aes(x = factor, y = var)) + geom_boxplot()
```

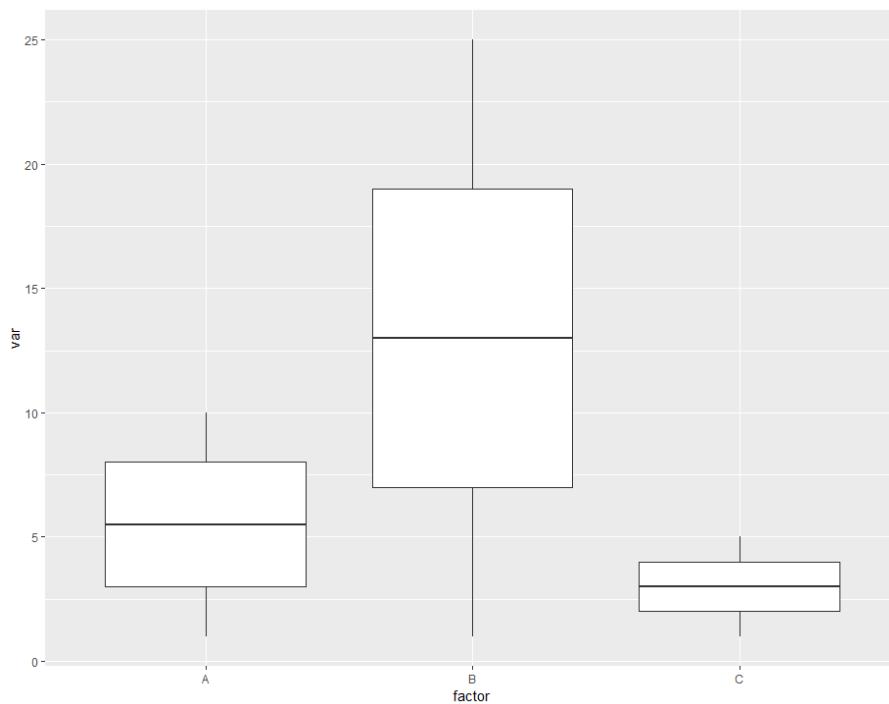


Рисунок 6. Вох-plot диаграмма

Однако наши экспериментные данные не лучшим образом подходят для этого типа графиков. Давайте рассмотрим на более подходящих данных.

В качестве тестовых данных возьмём встроенный набор "*HSAUR2*", который имеет следующую структуру:

```
# 'data.frame': 40 obs. of 3 variables:
# $ source : Factor w/ 2 levels "Beef","Cereal": 1 1 1 1 1 1 1 1 1 ...
# $ type : Factor w/ 2 levels "High","Low": 2 2 2 2 2 2 2 2 2 ...
# $ weightgain: int 90 76 90 64 86 51 72 90 95 78 ...
```

Результат работы блока кода ниже можно наблюдать на рисунке 7.

```
library(HSAUR2)
data(weightgain)
str(weightgain)

ggplot(data = weightgain, aes(x = type, y = weightgain)) +
  geom_boxplot(aes(fill = source))
```



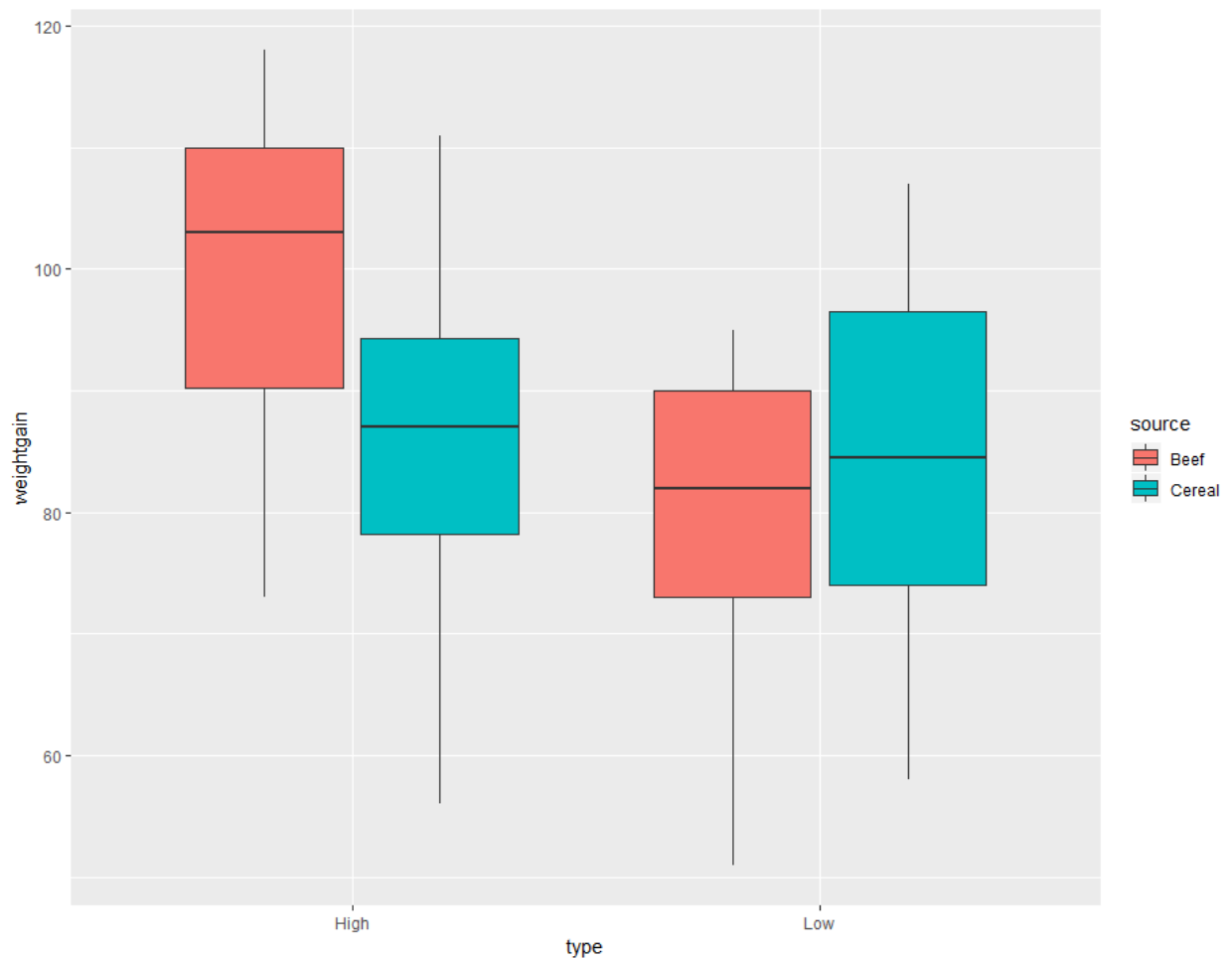


Рисунок 7. Box-plot диаграмма

### 3.3 Тепловые карты

Ещё один полезный вид визуализации – тепловые карты. Для этого нам понадобится дополнительно библиотека *tidyverse*.

*Tidyverse* – включает в себя целый ряд полезных библиотек, включая *ggplot2*, по этому можно подключить только её. Вот их список

```
## ✔ ggplot2 3.0.0      ✔ purrr 0.2.5
## ✔ tibble 1.4.2       ✔ dplyr 0.7.6
## ✔ tidyr 0.8.1        ✔ stringr 1.3.1
## ✔ readr 1.1.1        ✔ forcats 0.3.0
```

С помощью следующего примера вы сможете нарисовать тепловую карту (см. рисунок 8).

```
library(tidyverse)
```

```

## make data
dat <- matrix(rnorm(100, 3, 1), ncol=10)

## reshape data (tidy/tall form)
dat2 <- dat %>%
  tbl_df() %>%
  rownames_to_column('Var1') %>%
  gather(Var2, value, -Var1) %>%
  mutate(
    Var1 = factor(Var1, levels=1:10),
    Var2 = factor(gsub("V", "", Var2), levels=1:10)
  )

## plot data
ggplot(dat2, aes(Var1, Var2)) +
  geom_tile(aes(fill = value)) +
  geom_text(aes(label = round(value, 1))) +
  scale_fill_gradient(low = "white", high = "red")

```

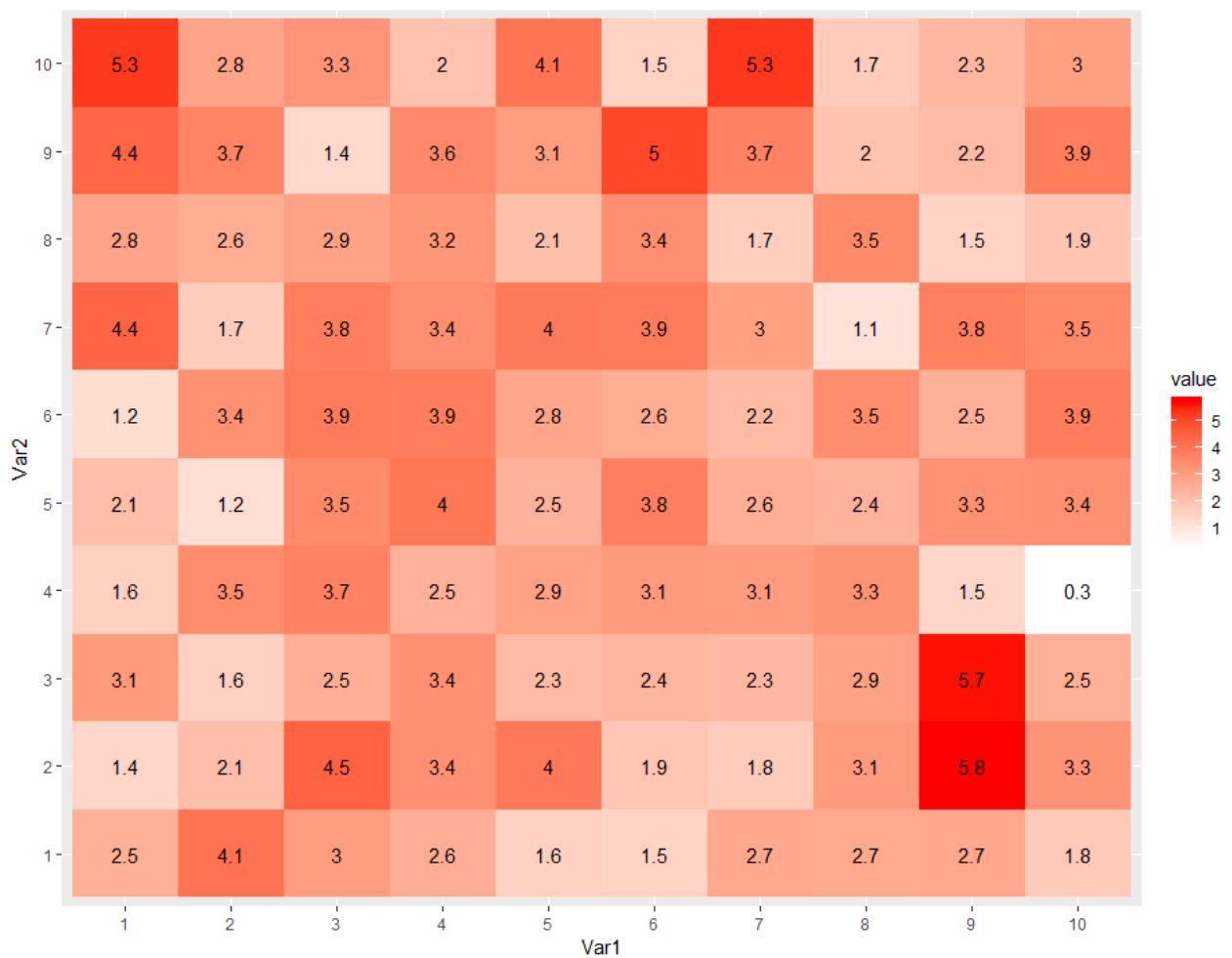


Рисунок 8. Тепловая карта

### 3.4 Корреляционная матрица

Корреляционная матрица (коррелограмма) позволяет исследовать корреляцию нескольких непрерывных переменных, присутствующих в одном и том же dataframe. Это удобно реализовать с помощью пакета *corrplot*.

В качестве тестовых данных возьмём встроенный набор "mtcars" и выполним кусок кода ниже. Результат его работы смотрите на рисунке 9.

```
# devtools::install_github("kassambara/ggcorrplot")
library(ggplot2)
library(ggcorrplot)

data(mtcars)
corr <- round(cor(mtcars), 1)

ggcorrplot(corr, hc.order = TRUE,
            type = "lower",
            lab = TRUE,
            lab_size = 3,
            method="circle",
            colors = c("tomato2", "white", "springgreen3"),
            title="Correlogram of mtcars",
            ggtheme=theme_bw)
```

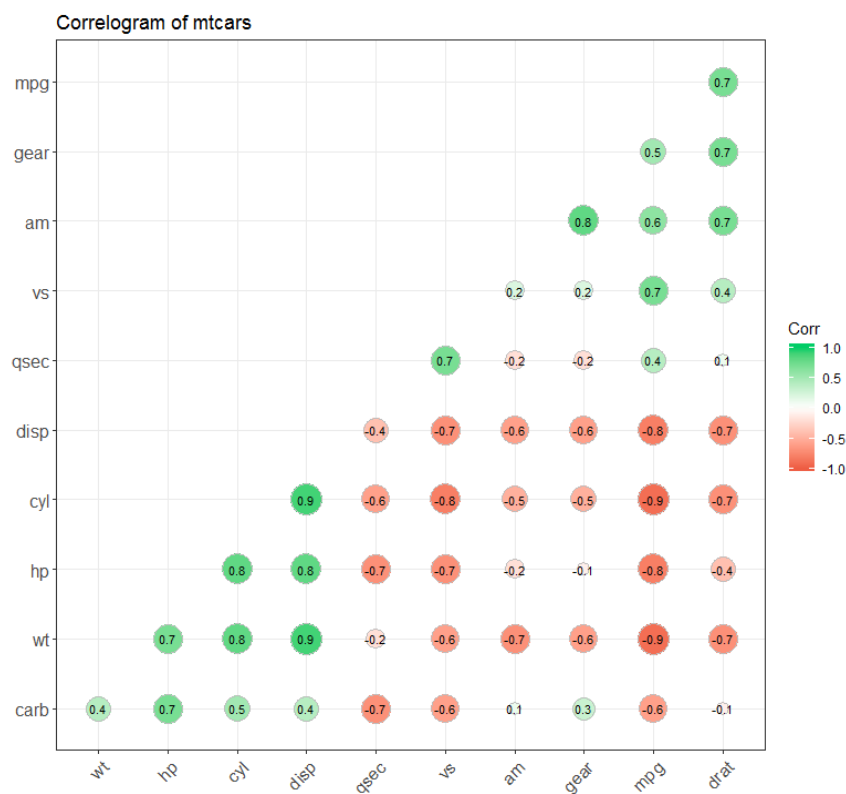


Рисунок 9. Корреляционная матрица

## ВЫВОДЫ

Я познакомился с большим количеством всевозможных способов использования пакета `ggplot2` для языка программирования R. Отобразил в этом отчёте наиболее интересные, по моему мнению. Указал в источниках материалы, где можно найти больше интересных пример использования этого графического пакета.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. С.Э. Мастицкий, В.К. Шитиков - СТАТИСТИЧЕСКИЙ АНАЛИЗ И ВИЗУАЛИЗАЦИЯ ДАННЫХ С ПОМОЩЬЮ R (2014)
2. <https://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html>
3. <https://ggplot2.tidyverse.org/reference/index.html#section-plot-basics>