Практическая работа №10

Построение графики в пакете ggplot2

Построение графики с пакетом ggplot2 не очень сильно отличается от использования стандартных графических возможностей R, однако степень возможной вариации и детализации результата действительно значительно выше по сравнению с классическим вариантом графики из пакета graphics::.

Установка демонстрационного скрипта:

```
# Скачивание ggplot2 в случае необходимости
if ("ggplot2" %in% rownames(installed.packages()) == FALSE) {
  install.packages("ggplot2") }
library(ggplot2)

# Существующий набор данных
df <- faithful
```

Графика в ggplot2 оперирует понятиями “холст” и “слои”. Холст в ggplot2 определяется указанием функции ggplot() в теле скрипта. При выполнении данной функции на выходе мы получаем открытый графический девайс R и пустой лист, на котором ничего не имеется.

Для определения того, что нужно отрисовать на графике необходимо указать в качестве аргументов data — фрейм данных, по которым будет строиться график. В аргументе mapping мы должны указать фукнцию aes(), в которой мы определим оси и размерность измерений.

Пример определения данных для графика в ggplot2:

```
ggplot(data = df, mapping = aes(eruptions, waiting))
```

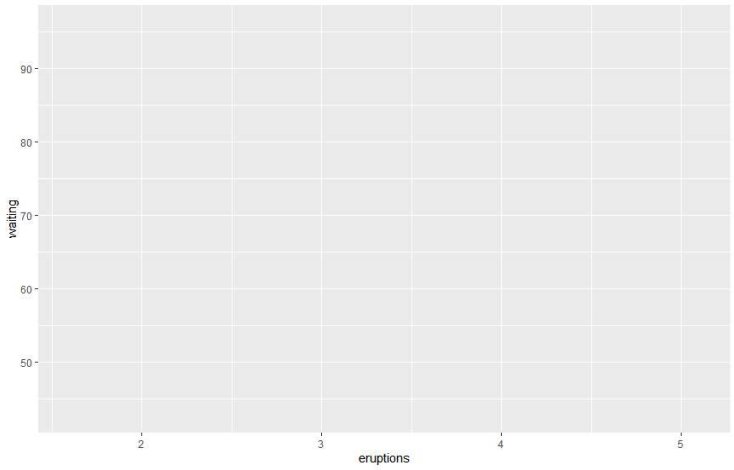

Рисунок 1 Пустой холст графика. Видны оси, видна сетка определения осей

Для того, чтобы на данный холст добавить маркеров или другой графики необходимо указать требуемый слой для построения из предиктивы geom_. Данные функции, начинающиеся с geom_ определяют слои, которые мы хотим нанести на график:

```
ggplot(data = df, mapping = aes(eruptions, waiting)) +
  geom_point()
```

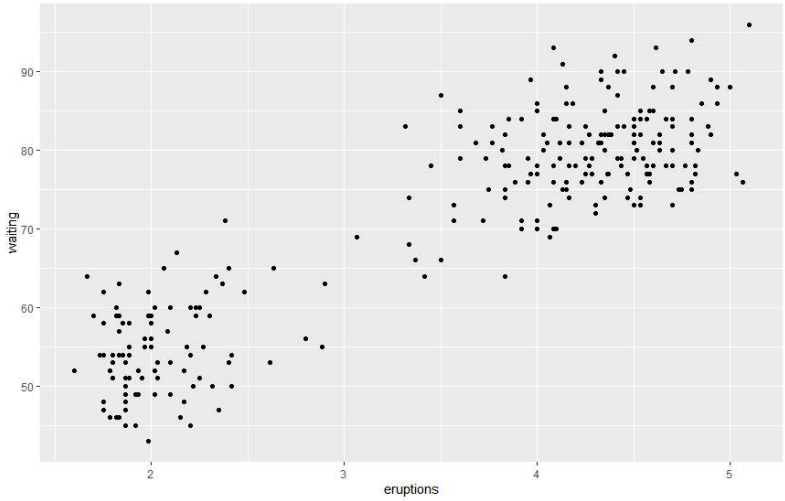

Рисунок 2 На холст графика нанесён слой точек

Слои и холсты в ggplot2 можно очень долго изменять, добиваясь безупречных результатов графического отображения:

```
ggplot(data = df, mapping = aes(eruptions, waiting)) +
  geom_point(color = kmeans(x = df, centers = 2)$cluster) +
  geom_density2d() +
  theme_bw() +
  labs(title = "Время ожидания между извержениями и продолжительность извержения",
       subtitle = "Старый Верный гейзер, штат Вайоминг, США.",
       x = "Продолжительность извержения, минуты",
       y = "Ожидание до следующего извержения, минуты") +
  scale_x_continuous(breaks = seq(min(df$eruptions), max(df$eruptions), 0.25)) +
  scale_y_continuous(breaks = seq(min(df$waiting), max(df$waiting), 5))
```

На рисунке 63 представлен пример графика.

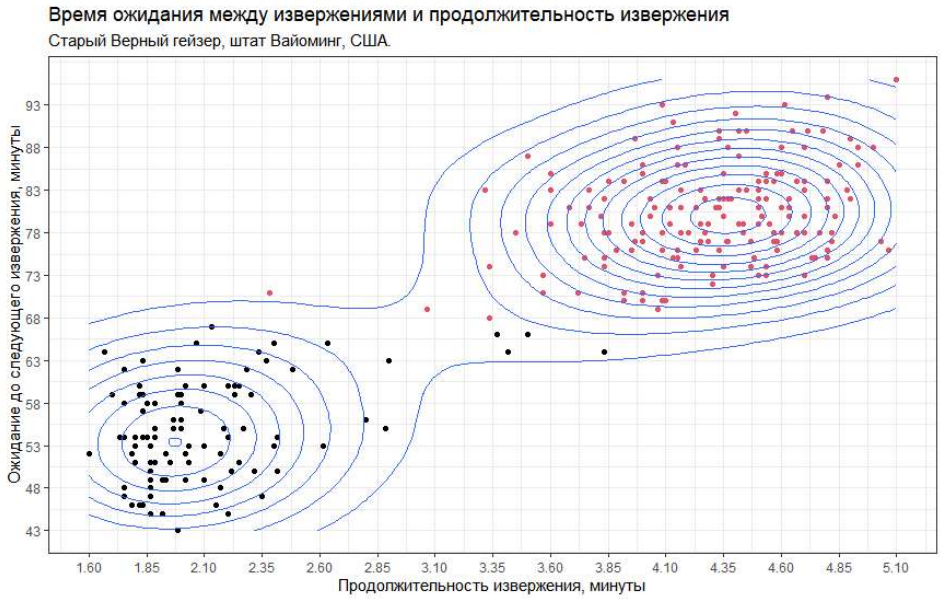

Рисунок 3 Красивая графика это просто

Отрисовка временного ряда

Чтобы научиться настраивать графику с помощью библиотеки ggplot2 необходимо несколько раз проделать некоторые базовые вещи чтобы понять суть и концепции данного инструмента. После нескольких повторений данная процедура будет занимать минимум времени и будет приносить максимум пользы.

Для того чтобы полностью погрузиться в работу над графикой, нам необходимо проделать весь процесс в рамках поставленных условий, это поможет нам не сбиться с маршрута и не наплодить лишних графических деталей.

К графикам временного ряда в анализе предъявляются некоторые требования, которые помогают процессу анализа:

- график должен быть квадратным;
- график временного ряда должен занимать всю полезную площадь поверхности рисования;
- график временного ряда должен быть отражён в виде линий с маркерами, где каждая точка будет различимой на фоне линий;
- подписи к графику должны быть понятными, единицы измерения величин единообразно определимыми.

После того, как мы определились с ограничениями к графику временного ряда, необходимо определиться с рядом данных. Воспользуемся скриптом загрузки финансовых данных из самостоятельной работы №6. Будем изучать графику временных рядов вместе с фреймом данных df.

Квадратный график. Сохранение рисунков определённого размера в R

производится путём сохранения содержимого графического устройства сессии в виде изображения одного из популярных форматов. Одним из самых популярных форматов является формат .png. Покажем, каким образом можно в R сохранять отдельные файлы изображений в формате png:

```
# Открытие png устройства
png(filename = "timeseries.png", width = 800, height = 800, units = "px")
# Функции графика
dev.off() # Сохранить в папке
```

Функция png() и dev.off() окаймляют отрисовку графика. Аргументы функции png() в комментариях не нуждаются.

Полезная площадь рисования. Для разбирательств с полезной площадью графика давайте попробуем хотя бы построить обычный график ряда. В функции ggplot() необходимо указать источник данных и оси координат, чтобы определить холст, также чтобы определить линии и точки необходимо добавить также некоторые слои через “+”:

```
# Открытие png устройства
png(filename = "timeseries.png", width = 800, height = 800, units = "px")
ggplot(data = df, mapping = aes(x = 1:nrow(df), y = df[[4]])) +
  geom_line() +
  geom_point()
dev.off() # Сохранить в папке
```

В качестве абсцисс мы указали перечисления торговых дней с момента начала торгов по акциям, до собственно конца строк в таблиц. В качестве ординат была взята четвёртая колонка фрейма данных. Функции geom_line() и geom_point() позволили отобразить точки на пересечении данных абсцисс-ординат, а также протянуть между точками прямые линии.

В результате выполнения, у нас возник в папке проекта файл “timeseries.png”, который выглядит следующим образом:

На рисунке 64 показан обычный график временного ряда.

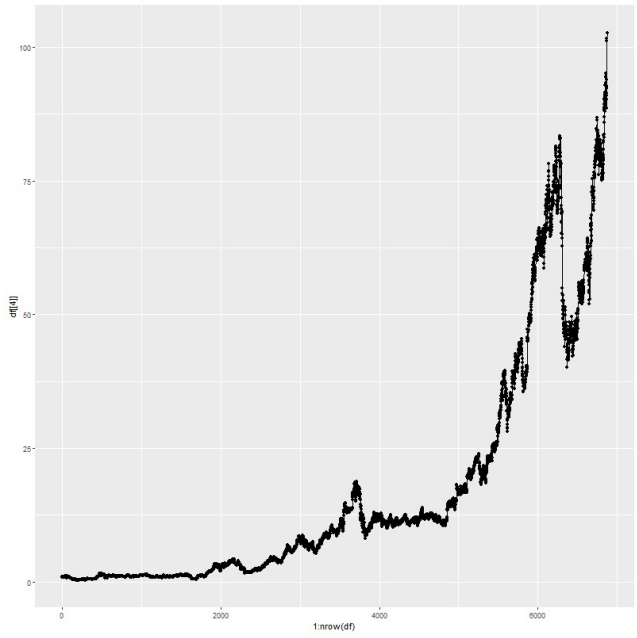

Рисунок 4 Обычный график временного ряда

Полезная площадь рисования графика определяется сеткой на которой график рисуется. Подрезать границы графика в случае образования пустого места можно с помощью coord_cartesian(), функция принимает границы рисования по x и y в аргументы xlim и ylim соответственно:

```
# Открытие png устройства
png(filename = "timeseries.png", width = 800, height = 800, units = "px")

ggplot(data = df, mapping = aes(x = 1:nrow(df), y = df[[4]])) +
  geom_line() +
  geom_point() +
  coord_cartesian(xlim = c(1, nrow(df)), ylim = range(df[[4]]))

dev.off() # Сохранить в папке
```

Различимые точки. В соответствующих слоях для отображения геометрии рисунка можно настроить некоторые особенности этих слоёв: размер, цвет, толщина, штрих-пунктир, прозрачность и т.д.

Изменим на нашем графике толщину линий и маркеров для соблюдения третьего требования:

```
ggplot(data = df, mapping = aes(x = 1:nrow(df), y = df[[4]])) +
  geom_line(lwd = I(0.5), lty = 1) +
  geom_point(cex = I(0.5)) +
  coord_cartesian(xlim = c(1, nrow(df)), ylim = range(df[[4]]))
```

В аргументах geom_ мы использовали аргументы lwd, lty и cex. Это стандартные для линий и точек аргументы графики. Аргумент lwd отвечает за

толщину линии, значение I(0.5) обозначает толщину вдвое меньшую стандартной. Аргумент lty – тип штрих-пунктира линии, определён целыми числами от 1 до 5. Аргумент cex – стандартный параметр точек, обозначает величину точек.

Подписи к графику. В библиотеке ggplot2 подписи к графикам ставятся через функцию labs():

```
ggplot(data = df, mapping = aes(x = 1:nrow(df), y = df[[4]])) +
  geom_line(lwd = I(0.5), lty = 1) +
  geom_point(cex = I(0.5)) +
  coord_cartesian(xlim = c(1, nrow(df)), ylim = range(df[[4]])) +
  labs(title = "График цен акций компании Activision-Blizzard",
       subtitle = paste("Данные от", min(rownames(df))),
       x = "Торговые дни от начала торгов, отсчитанные с 1",
       y = "Цены акций в долл.США")
```

Добавление деталей. Цвет холста – дело вкуса. Поменять цвет холста в ggplot2 на белый можно с помощью theme_bw().

Изменить дискретизация подписей координатной сетки можно с помощью scale_ _continuous()

```
# Открытие png устройства
  png(filename = "timeseries.png", width = 800, height = 800, units = "px")

  ggplot(data = df, mapping = aes(x = 1:nrow(df), y = df[[4]])) +
    geom_line(lwd = I(0.5), lty = 1) +
    geom_point(cex = I(0.5)) +
    coord_cartesian(xlim = c(1, nrow(df)), ylim = range(df[[4]])) +
    labs(title = "График цен акций компании Activision-Blizzard",
         subtitle = paste("Данные от", min(rownames(df))),
         x = "Торговые дни от начала торгов, отсчитанные с 1",
         y = "Цены акций в долл.США") +
    scale_x_continuous(breaks = seq(1, nrow(df), 365)) +
    scale_y_continuous(breaks = seq(0, 100, 10)) +
    theme_bw()

dev.off() # Сохранить в папке
```

На рисунке 65 показан график временного ряда по требованиям.

Рисунок 5 График временного ряда по требованиям

Для отображения гистограммы ряда необходимо произвести некоторые изменения в коде, поменять слои и варианты отображения:

```
# Открытие png устройства
png(filename = "histogram.png", width = 600, height = 600, units = "px")

ggplot(data = df, mapping = aes(df[[4]])) +
  geom_histogram(bins = 40, color = "black", fill = "grey") +
  geom_freqpoly(bins = 40, color = "red4", lwd = I(1.1)) +
  labs(title = "Гистограмма распределения данных статического разреза",
       subtitle = "Временной ряд цен акций Activision Blizzard",
       x = "Данные цен акций в долл.США",
       y = "Количество измерений в промежутке") +
  scale_x_continuous(n.breaks = 15) +
  scale_y_continuous(n.breaks = 15) +
  theme_bw()

dev.off() # Сохранить в папке
```

В данном скрипте выбраны следующие слои для отображения гистограммы. Для построения столбцов был использован слой geom_histogram(). Для построения линии, соединяющей середины отрезка geom_freqpoly(). Остальные параметры были использованы ранее.

Для определения данных в aes() для гистограммы необходимы только сами данные без оси абсцисс.

На рисунке 66 показана гистограмма распределения измерений по естественным границам значений ряда.

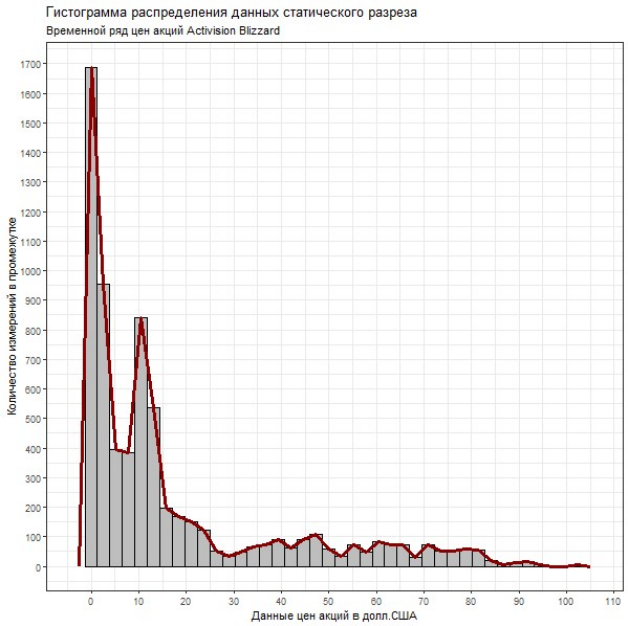

Рисунок 6 Гистограмма распределения измерений по естественным границам значений ряда

Для отображения вероятностей и ядровой функции распределения используются хитрые приёмы, которые просто лучше запомнить:

```
# Открытие png устройства
png(filename = "density_histogram.png", width = 600, height = 600, units = "px")

ggplot(data = df, mapping = aes(df[[4]])) +
  geom_histogram(aes(y = ..density..),
                 binwidth=density(df[[4]])$bw,
                 color = "black",
                 fill = "grey") +
  geom_density(fill="red", alpha = 0.2)+
  labs(title = "Гистограмма и ядровая функция распределения данных",
       subtitle = "Временной ряд цен акций Activision Blizzard",
       x = "Данные цен акций в долл.США",
       y = "Вероятность попадания значения в промежуток") +
  scale_x_continuous(n.breaks = 15) +
  scale_y_continuous(n.breaks = 15) +
  theme_bw()

dev.off() # Сохранить в папке
```

На рисунке 67 показан результат выполнения данных команд – файл с визуализацией:

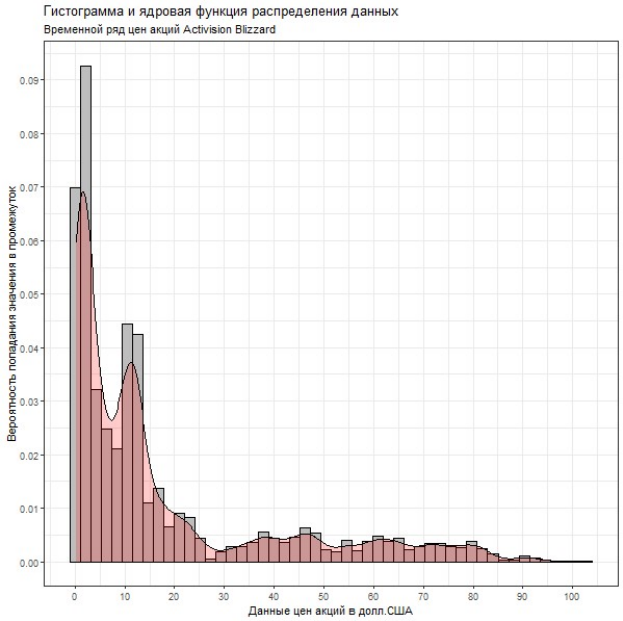

Рисунок 7 Гистограмма вероятностей и ядерная функция распределения для данных статического разреза

Самостоятельная работа №10

Часть 1

Задание 1.

Все графики строятся с помощью библиотеки ggplot2.

1. Загрузите файл demography.csv. В нём содержатся данные по населению Белгородской и Калужской областей за 2016 год (источник – Росстат).

   Переменные:

   - region: название региона;
   - district: название района;
   - empl_total: численность занятого населения;
   - A-O: занятость по отраслям (как на сайте Росстата: сельское хозяйство);
   - popul_total: численность населения;
   - urban_total: численность городского населения;
   - rural_total: численность сельского населения;
   - wa_total: численность трудоспособного населения;
   - wa_female: численность трудоспособного населения (женский пол);
   - wa_male: численность трудоспособного населения (мужской пол);
   - ret_total: численность пенсионеров;

- ret_female: численность пенсионеров (женский пол);
- ret_male: численность пенсионеров (мужской пол);
- young_total: численность населения, моложе трудоспособного возраста;
- young_female: численность населения, моложе трудоспособного возраста (женский пол);
- young_male: численность населения, моложе трудоспособного возраста (мужской пол);
- X18_19 – X70_plus: численность населения по возрастным группам.

2. Создайте переменную young_share – процент населения возраста, моложе трудоспособного. Создайте переменную trud_share – процент населения трудоспособного возраста и old_share – процент населения возраста, старше трудоспособного.
3. Постройте гистограмму для доли трудоспособного населения в процентах. Измените цвет гистограммы, добавьте *rugs*. Добавьте вертикальную линию, которая отчерчивает медианное значение доли трудоспособного населения в процентах.
4. Постройте сглаженные графики плотности распределения для доли трудоспособного населения в процентах по регионам (два графика в одной плоскости). Настройте цвета и прозрачность заливки. По графикам плотности определите, имеет ли смысл для визуализации распределения доли трудоспособного населения строить скрипичные диаграммы (*violin plot*). Если да, постройте их (так же по группам). Если нет, постройте ящики с усами.
5. Постройте диаграмму рассеяния для переменных young_share и old_share. Можно ли сказать, что чем больше процент молодого населения (моложе трудоспособного населения), тем меньше процент пожилых людей (старше трудоспособного возраста)? Поменяйте цвет и тип маркера для точек.
6. Создайте переменную male_share – доля мужского населения в районе/городе (в процентах). Создайте переменную male, которая принимает значение 1, если доля мужчин в муниципальном районе/городе больше доли женщин, и значение 0 – во всех остальных случаях.

7. Постройте пузырьковую диаграмму (*bubble plot*) для переменных young_share и old_share, учитывая информацию о доле мужчин в районе и о том, преобладают ли мужчины в районе или нет.

Постройте столбиковую диаграмму (*bar plot*), которая показывала бы, сколько в базе данных районов Белгородской области, а сколько – Калужской.

Часть 2

Задание 1

В данном задании нужно работать со встроенной в R базой данных по автомобилям mtcars. Загружать ее по ссылке не нужно, достаточно набрать ее название (data = mtcars). Например, чтобы посмотреть на базу, можно просто воспользоваться View(mtcars).

Постройте с помощью библиотеки ggplot2 пузырьковую диаграмму (bubble plot), которая

- показывала бы связь между показателями *Gross horsepower* (hp) и *Weight* (wt);
- учитывала бы информацию о числе цилиндров у автомобиля (cyl);
- учитывала бы информацию о типе коробки передач – автоматическая или нет (am); сделайте так, чтобы легенда графика была корректной и информативной + пусть точки, соответствующие автомобилям с автоматической коробкой передач, будут зеленого цвета ("green"), а с ручной – красного ("red").

Подпишите оси (дайте им более вразумительные названия). Добавьте название (заголовок) графика.

Задание 2

Работая с той же базой mtcars, воспроизведите следующий график:

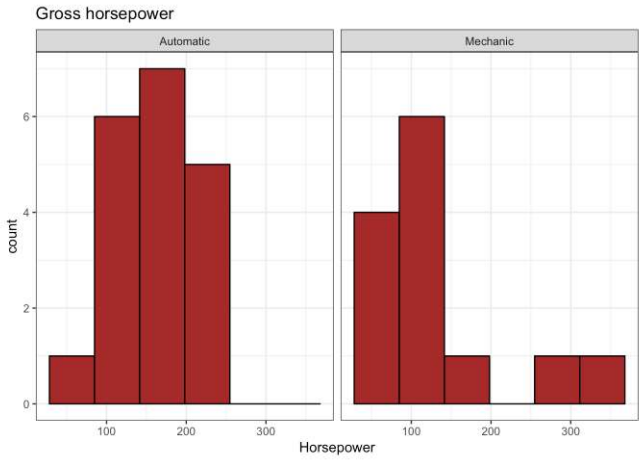

Рисунок 8 Пример графика

Подсказка: цвет – “brown”, 0 – автоматическая коробка передач, 1 –

ручная (am) число столбцов (bins) можно определить по графику.

Задание 3

В данном задании нужно работать с базой sleep, встроенной в R.

Постройте «ящики с усами» в пределах одной области для графика, которые иллюстрировали бы распределение переменной extra по группам испытуемых. Поменяйте базовые цвета заливки графиков, добавьте подписи к осям и заголовок графика.

Часть 3

Визуализация данных временного ряда COVID19

В данной работе для проработки изложенного материала предлагается визуализировать данные временного ряда, полученные ранее в ходе выполнения самостоятельной работы №3 (предобработка данных Covid19).

Задание

Проделать все шаги визуализации данных временного ряда, рассмотренные в практической работе №10 применительно к 3 временным рядам различных стран (на выбор) из фрейма данных полученного в самостоятельной работе №3.