УДК 61:004.92:621.39
DOI: http://dx.doi.org/10.11603/mie.1996-1960.2017.2.7886

ГРАМАТИКА МОВИ ГРАФІЧНОЇ ВІЗУАЛІЗАЦІЇ МЕДИЧНИХ ДАНИХ В ПАКЕТІ GGPLOT2

В. П. Марценюк, І. Є. Андрущак[1], А. І. Банадига[2]

Університет Бєльсько-Бяли, Польща
[1]Луцький національний технічний університет
[2]ДВНЗ «Тернопільський державний медичний університет імені І. Я. Горбачевського МОЗ України»

У роботі показано застосування парадигми граматики графіки пакету ggplot2 на прикладі візуалізації даних медичних досліджень. Цей підхід пропонує гнучкий інструмент для побудови багатошарових і багатопанельних графіків на основі даних клініко-лабораторних досліджень.

Ключові слова: комп'ютерна графіка, візуалізація даних, медичні наукові дослідження, R, ggplot2.

GRAMMAR OF LANGUAGE OF GRAPHICAL VISUALIZATION OF MEDICAL DATA IMPLEMENTED IN PACKAGE GGPLOT2

V. P. Martsenyuk, I. Ye. Andrushchak[1], A. I. Banadyha[2]

University of Bielsko-Biala, Poland
[1]Lutsk National Technical University
[2]SHEI "I. Ya. Gorbachevsky Ternopil state medical university of MH of Ukraine"

Application of package ggplot2 graphics grammar paradigm is shown on the example of medical researches data visualization. This approach offers a flexible instrument for the construction of multi-layered and multipanel charts on the basis of clinical and laboratory researches data.

Key words: computer graphics, rendering data, medical research, R, ggplot2.

ГРАММАТИКА ЯЗЫКА ГРАФИЧЕСКОЙ ВИЗУАЛИЗАЦИИ МЕДИЦИНСКИХ ДАННЫХ В ПАКЕТЕ GGPLOT2

В. П. Марценюк, И. Е. Андрущак[1], А. И. Банадыга[2]

Университет Бельсько-Бялы, Польша
[1]Луцкий национальный технический университет
[2]ГВУЗ «Тернопольский государственный медицинский университет имени И. Я. Горбачевского МЗ Украины»

В работе показано применение парадигмы грамматики графики пакета ggplot2 на примере визуализации данных медицинских исследований. Данный подход предлагает гибкий инструмент для построения многослойных и многопанельных графиков на основе данных клинико-лабораторных исследований.

Ключевые слова: компьютерная графика, визуализация данных, медицинские научные исследования, R, ggplot2.

Введение. Современные научные исследования в медицине и биологии неразрывно связаны с графическим представлением и визуализацией данных [4]. Визуализация данных — это значительно больше, чем просто набор технологий для рисования графиков. Это на самом деле является способом мышления [2, 3].

Для того чтобы развивать этот «способ мышления», необходимы инструменты, позволяющие экспериментировать с представлением данных.

Можно выделить три основных вида инструментов, используемых для создания графиков.

1. Программы типа GIMP, Adobe Illustrator или Inkscape. Они позволяют построить произвольную информационную графику. Воображение разработчика практически ничем не ограниченно, даже данными, которые нужно представить. Можно построить произвольный график, пользуясь произвольными формами.
   GIMP (https://www.gimp.org) — это бесплатная программа для растровой графики, Inskape (https://inkscape.org) — для векторной графики. Данные инструменты не создают соединения между данными и элементами графика. Именно «сам график» решает: что, как и почему должно быть представлено. Таким образом, именно график «заботится» о целостности и согласованности данных для представления. При более сложных историях здесь легко что-то упустить.
2. Программы типа Calc, Excel, Tableau. Позволяют быстро создавать графики с помощью набора шаблонов. Быстрота является здесь ключевым аргументом. Пользователь может молниеносно «собрать» шаблон, указать столбцы данных, которые параметризуют шаблон, и получить готовый график.
   Этот подход позволяет быстро создавать графики, но является ограниченным. Доступных шаблонов может существовать достаточно много, но если не имеется подходящего шаблона для нашей истории (либо модели данных), то мы не сможем ни просто, ни быстро построить нового.
   Как правило, доступные шаблоны не способны изложить сложных историй. Вместо одного содержательного графика создаются наборы из простых графиков, которые, правда, можно быстро строить и сопоставлять.
3. Библиотеки языков программирования, такие, как ggplot2. Они опираются на сравнительно небольшой набор элементов, которые можно гибко складывать между собой в сложные графики. Гибкость является здесь определяющим словом. Складывание элементов на многих слоях позволяет представлять сложные истории.

Грамматика построения графиков разработана Hadley Wickham в работах [5, 6] и реализована в библиотеке ggplot2 для среды R. Эта грамматика опирается на более общую грамматику, которую создал Leland Wilkinson (описана в книге [7]). В свою очередь, грамматика, представленная Wilkinson, базируется на результатах Jacques Bertin [1]. Здесь используются также результаты, полученные в очень многих разнообразных дисциплинах: от картографии, изобразительного искусства, через исследования восприятия и когнитивистику до лингвистики, математики и статистики. Визуализацию информации можно по праву считать междисциплинарным направлением.

Цель работы — представить подходы к визуализации медицинских данных на основе грамматики языка ggplot2.

Материалы и методы исследования. Графики проектируются для того, чтобы представлять истории, записанные в данных. Термин «история» встречается в работах [4, 6] и имеет самое общее значение, которое соответствует с точки зрения анализа данных понятию «модели данных». Истории могут быть простыми (например, «в 2017 году увеличилось финансирование здравоохранения Украины по сравнению с 2016 годом») или сложными («финансирование связано с реформированием здравоохранения, созданием госпитальных округов, развитием первичного уровня медицинской помощи»). Чем сложнее история, тем больше усилий нужно приложить, чтобы ее правильно и доступно представить.

Графики создаются для того, чтобы представлять зависимости, присутствующие в данных. Следовательно, можно рассматривать графики как рассуждения, которые описывают зависимости. На основе каких правил строятся такие рассуждения? Ключевыми являются два аспекта.

1. Если мы употребим достаточно богатый язык, то с помощью такого одного рассуждения / одним графиком мы сможем

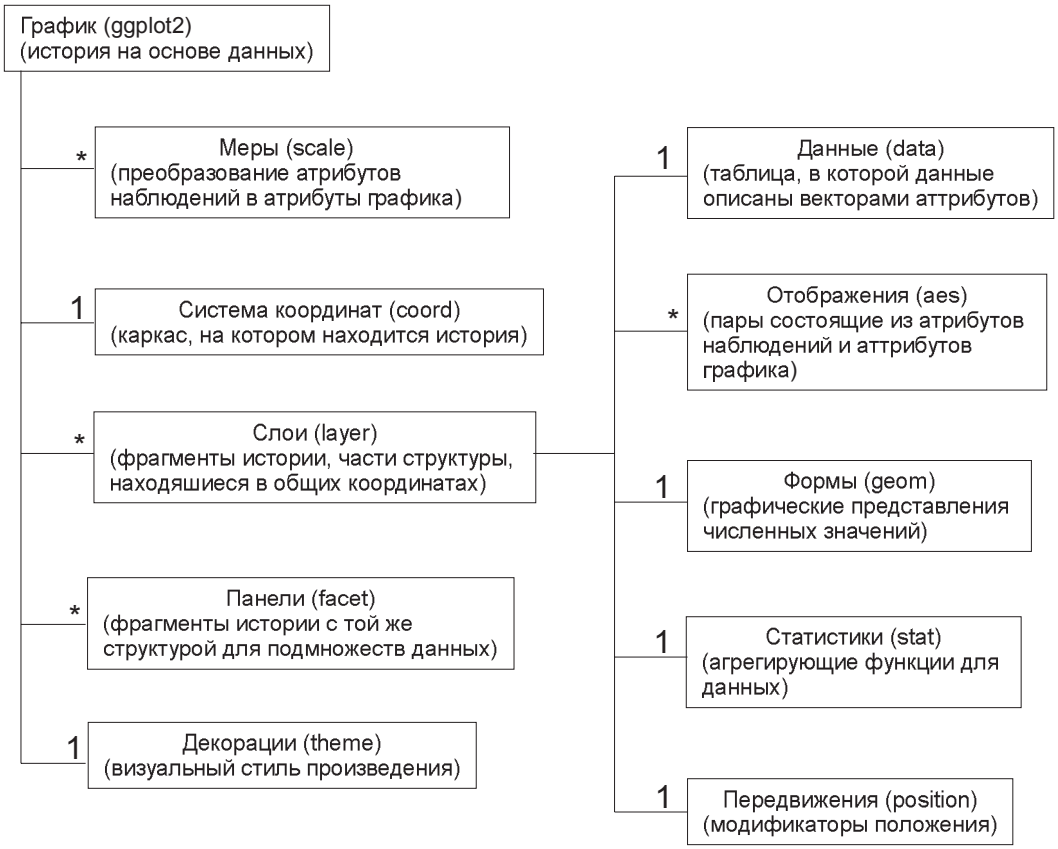

Рис. 1. Грамматика языка визуализации данных ggplot2

представить сложную историю. Если мы воспользуемся простым языком, то для передачи того же содержания нам потребуется много рассуждений / графиков. Более того, мы увеличиваем риск, что определенных мыслей с помощью такого простого языка нам не удастся выразить, либо мы останемся непонятыми.

2. Нет причин полагать, что умение чтения графиков является прирожденным. У многих возникают трудности с интерпретацией даже простых графиков, это указывает на то, что такое умение является приобретаемым. Уделяя время работе с графиками, мы изучаем правила их конструирования и интерпретации.

Пакет ggplot2 является одним из наиболее «продвинутых» инструментов для создания статистических графиков. «Продвинутость» не означает, что можно быстро сделать в нем график, а также то, что доступными являются много шаблонов графиков. Заметим: конструкция пакета настолько эластична, что можно с ним реализовать практически любую статистическую графику.

Эта гибкость получается за счет того, что структура графика опирается на способ, которым мы думаем и читаем графики. Глядя на график, мы не видим в нем набор отрезков и окружностей, а видим коллекции объектов, по-своему похожих или разных. Следовательно, создавая график, мы не должны думать о том, где и как нарисовать отрезок, а о том, как элементы графика должны представлять данные.

Грамматика графики (т. е. языка визуализации данных) в самом широком аспекте описана в работе [5].

Грамматика, реализованная в пакете ggplot2, позволяет строить графики в соответствии со структурой, представленной на рисунке 1. Она состоит из многих элементов. Далее поочередно обсудим их применение с целью визуализации данных клинико-лабораторных исследований.

Результаты и их обсуждение. Приведенные ниже примеры опираются на наборы данных клинико-лабораторных исследований панкреатита.

```
investigations = read.table(«D:/My_doc/Banadyha/BanEng1.csv», header = TRUE, sep
= «;»)
head(investigations)
```

```
  Pain Vomiting Bloating Flatulence Mayo_Robsons_symptom Voskresensky_symptom
1    1        1        1          1                    1                    0
2    1        1        0          0                    0                    0
3    1        0        0          0                    0                    0
4    1        1        1          1                    0                    1
5    1        1        0          0                    1                    1
6    1        1        1          1                    1                    0

  Kert_symptom Cardiovascular_dysfunct. Respiratory_dysfunct. Amylase Diastase
1            1                        0                    0     22.4      242
2            1                        0                    0     25.4      275
3            0                        0                    0    124.8     1128
4            1                        0                    0     89.0      573
5            1                        0                    0    115.0     1153
6            1                        0                    0     87.7      120

  Glucose Bilirubin    ALT  AST Leukocytes Hemoglobin Total_protein Group
1    6.48     37.30 111.00 99.0        7.2        122         70.00     1
2    9.67     14.50  34.00 48.0        3.8        128         64.00     1
3    5.09     11.28  13.28  7.6        5.1        117         73.00     1
4    5.30     16.90  64.00 87.0        9.6        141         67.00     1
5    5.32     16.92  42.00 37.0        9.0        124         70.28     1
6    8.60     47.20  18.60 19.3        8.0        146         74.30     1
```

Минимальное определение графика в пакете ggplot2 состоит, по крайней мере, из трех элементов.

1. Функция ggplot() создает основу графика. Здесь декларируются общие параметры для других элементов графика. Декларация может быть пустой, но обычно здесь указывается набор данных (ниже investigations) и отображения (ниже функция aes()).
2. Функции geom_/stat_ создают последовательные слои представления данных, называемые далее геометриями (в англоязычных источниках употребляется термин «геом»). Ниже используется функция geom_point(), которая создает слой с употреблением точек.
3. Оператор + соединяет описания последовательных элементов графика.

На рисунке 2 построен график, представляющий с помощью точек информацию о зависимости логарифмических значений амилазы и диастазы для пациентов с панкреатитом:

```
library(ggplot2)
ggplot(investigations,
aes(x=log(Amylase),
y=log(Diastase))) + geom_point()
```

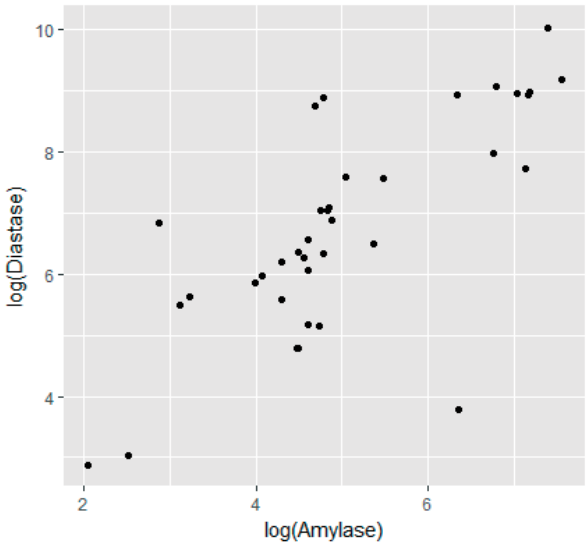

Рис. 2. Точечный график

Определение отображения. Графики представляют собой наборы объектов, которые описываются с помощью графических атрибутов. Отображения определяют, какие атрибуты графиков соответствуют каким переменным из набора данных.

Отображения описываются внутри функции aes() (сокращение от aesthetic). Они являются парами вида *графический атрибут = название переменной.*

Для каждого типа геометрии (слои графика) определено, какие графические атрибуты могут представлять данные. Список атрибутов, которые можно использовать для геометрии geom_point, находится по адресу http://docs.ggplot2.org/current/geom_point.html. В этом случае обязательные атрибуты x и y — координаты точек. На графике рисунка 3 мы определяем также отображения для атрибутов: цвет (color) и форма (shape):

```
library(ggplot2)
investigations = read.table(«D:/
My_doc/Banadyha/BanEng1.csv»,
header = TRUE, sep = «;») %>%
mutate(Group=factor(Group))
ggplot(investigations,
aes(x=log(Amylase), y=log(Diastase),
color=Group, shape=Group)) + geom_
point()
```

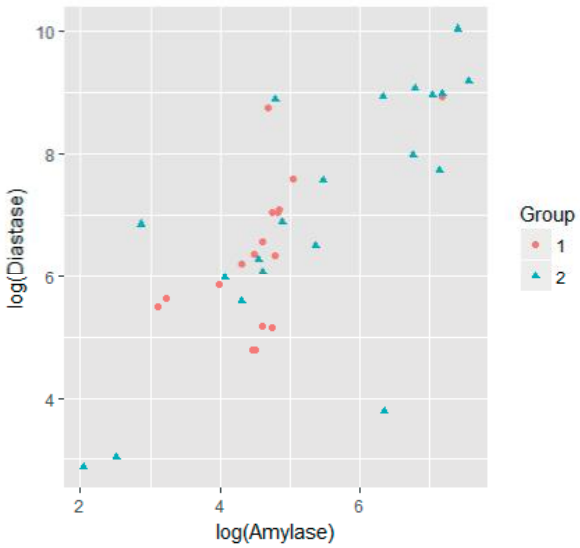

Рис. 3. Использование отображений aes() для определения атрибутов color, shape

В этом примере, определяя отображение shape=Group, мы устанавливаем, чтобы формы точек отвечали группам пациентов. Мы, однако, не определяем, какая форма должна быть для какой группы.

Способ отображения библиотека ggplot2 выбирает на основании типа переменной (factor / численная / логическая) и числа уровней, которые должны быть представлены.

Например, на предыдущем графике мы представляли группы пациентов с помощью цветов. Цвета подбирались так, чтобы можно было легче различить отдельные группы. Однако нет никакого предопределенного порядка между группами пациентов.

На следующем примере мы отображаем цвет на количественную переменную — глюкозу. Здесь уже имеет существует упорядоченность значений, и она отображается с помощью шкалы цветов — от голубого до черного:

```
ggplot(investigations,
aes(x=log(Amylase), y=log(Diastase),
color=Glucose, size=Glucose)) +
geom_point()
```

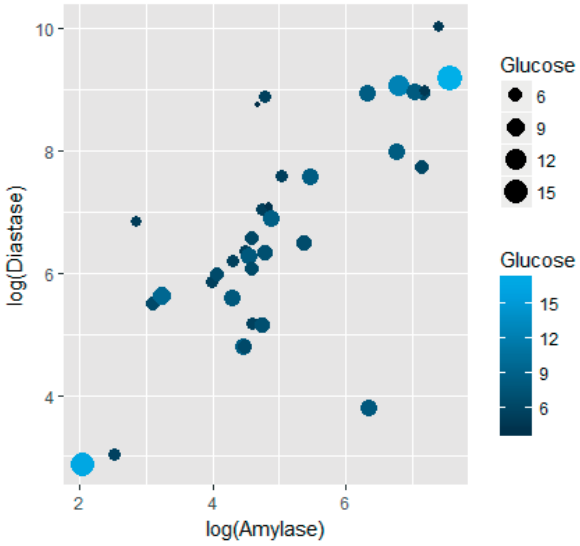

Рис. 4. Использование в отображении aes() количественной переменной для атрибутов color, size

Определение геометрий. Геометрии определяют наборы форм, которые представляют данные. Это могут быть точки (геометрия geom_point()), линии, прямоугольники, области и фигуры практически произвольного вида.

Список доступных на данный момент геометрий в наличии на http://docs.ggplot2.org/current/. Пакет ggplot2 имеет также доступные механизмы для создания произвольных новых геометрий.

Далее представлены примеры геометрии:

- geom_dotplot (погруппированные точки)

```
ggplot(investigations, aes(x
= Group, y = Amylase)) +geom_
```

```
dotplot(binaxis = «y», stackdir =
«center»)
```

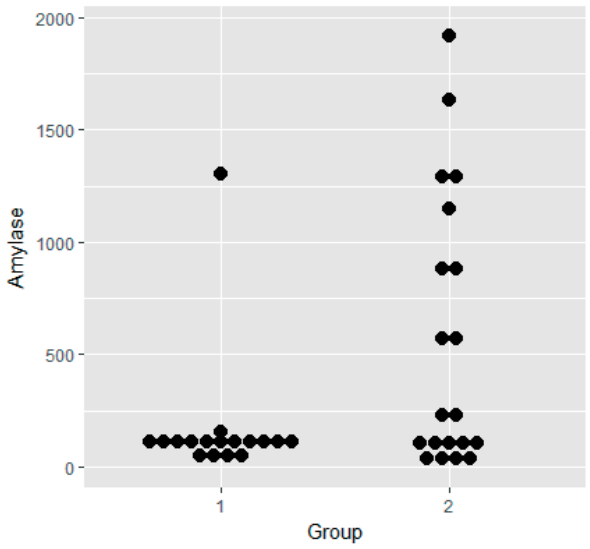

Рис. 5. Геометрия geom_dotplot

- geom_violin (скрипки):

```
ggplot(investigations, aes(x =
Group, y = Amylase, color=Group,
fill=Group)) + geom_violin()
```

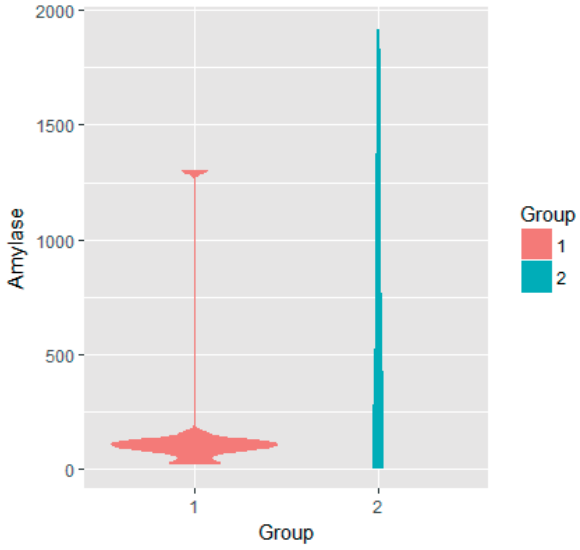

Рис. 6. Геометрия «скрипки»

- geom_line (линии):

```
library(tidyr)
investigations %>%
gather(rate, values, Amylase,
Diastase) %>%
group_by(Group, rate) %>%
summarise(values = mean(values,
na.rm=TRUE)) %>%
ggplot(aes(x = rate, y = values,
group=Group, color=Group)) +
geom_line(size=2) +
geom_point(size=4)
```

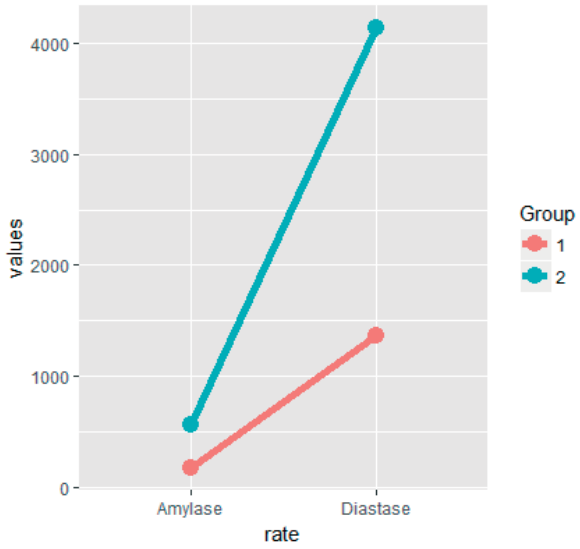

Рис. 7. Геометрия «линии»

Создание многослойных графиков. Создание сложной и богатой на содержание графики возможно в ggplot2 благодаря складыванию слоев. Все слои существуют в рамках общей системы координат графика. Благодаря этому объекты можно легче сравнивать между слоями. Это дает большие возможности для построения многослойных графиков с дополняющимися контентами.

Добавление последующего слоя происходит с помощью добавления оператором + следующей геометрии. Далее приведен пример графика с тремя слоями (рис. 8):

```
library(ggrepel)
ggplot(investigations,
aes(x=log(Amylase), y=log(Diastase),
label=Total_protein)) + geom_point()
+ geom_smooth(se=FALSE, size=3) +
geom_text_repel(data=investigations
[c(21,25,33),], color=»red»)
```

Это, в свою очередь, слой точек, слой кривой тренда и слой со значениями общего белка для некоторых пациентов.

Слои могут дополняться. На переднем плане находится линия с трендом, точки играют второстепенную роль на втором плане. Надписями отмечены значения общего белка для наиболее крайних точек.

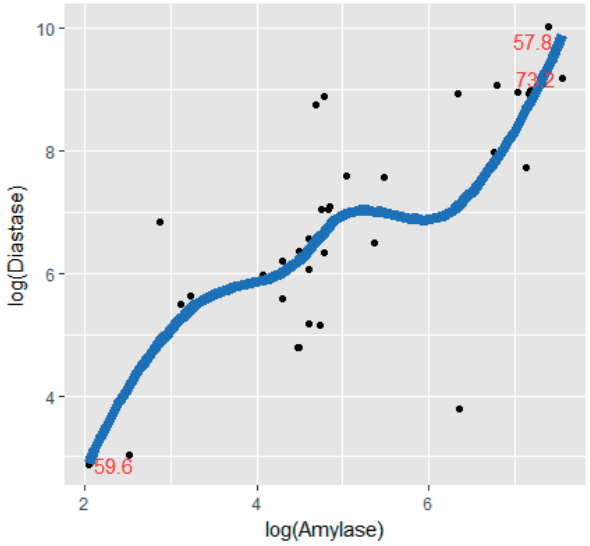

Рис. 8. Многослойный график

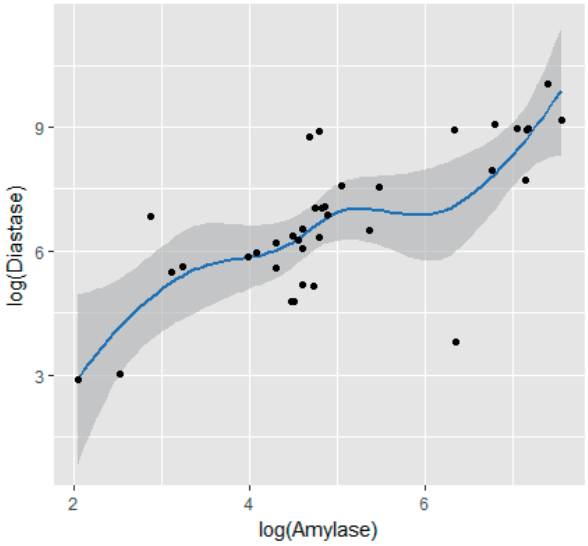

Рис. 9. Построение статистики тренда stat_smooth()

Построение статистик. Обычно используются табличные данные, в которых переменные представлены в столбцах, а отдельные наблюдения – в строках.

Однако мы не всегда хотим, чтобы каждая строка была представлена на графике. В определенных ситуациях вместо представления всех строк отдельно предпочтительным является рассчитать по ним какую-то статистику и ее представить на графике.

Такая статистика может характеризовать зависимость, заключенную в данных, и быть показательным дополнением для презентации отдельных точек.

Слои со статистикой можно создавать, как используя функцию stat_ (список таких функций находится здесь: http://docs.ggplot2.org/current/), так и через функции geom_, в которых определяется аргумент stat.

Статистику можно параметризовать. Например, статистика stat_smooth() имеет аргумент method, который определяет, каким способом должен строиться тренд данных, статистика stat density2d() позволяет определить параметры плотности, включая ширину окна.

Далее мы представим четыре примерных статистики. Каждая из них создает отдельный слой на графике:

- статистика тренда (рис. 9):

```
ggplot(investigations,
aes(x=log(Amylase),
y=log(Diastase))) +
stat_smooth() + geom_point()
```

- статистика плотности ядра распределения (рис. 10):

```
ggplot(investigations,
aes(x=log(Amylase),
y=log(Diastase))) +
stat_density2d(h=c(10,10),
color=»grey») + geom_point()
```

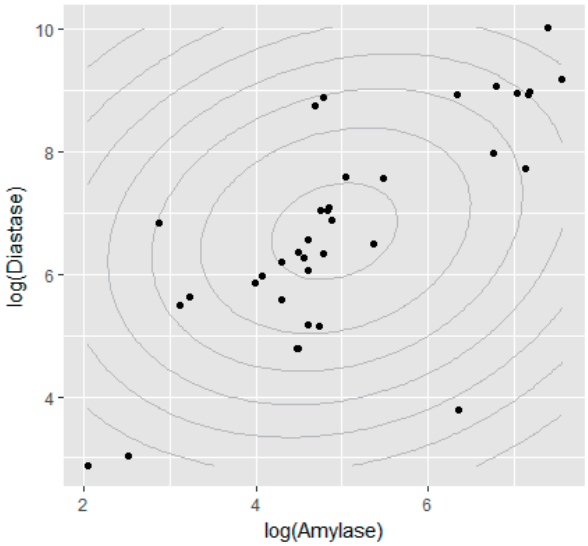

Рис. 10. Статистика плотности ядра распределения

- статистика уровней значений количественной переменной (рис. 11):

```
ggplot(investigations, aes(x=Group,
y=Amylase)) +
stat_boxplot(fill=»grey», coef = 3) +
```

```
geom_jitter(width = 0.3)
```

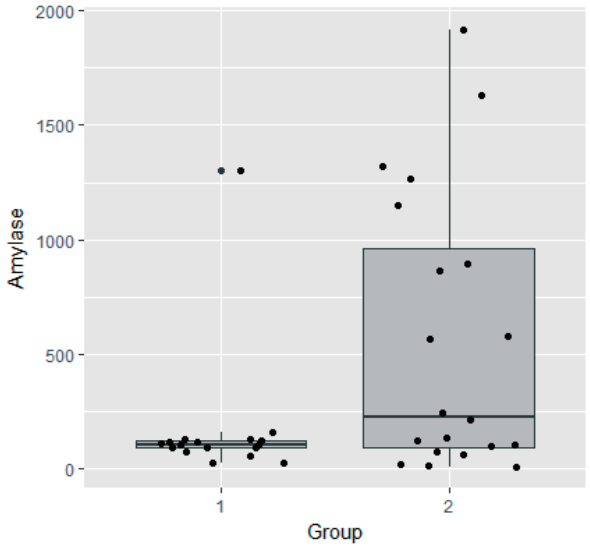

Рис. 11. Статистика уровней значений количественной переменной (амилазы)

- статистика количества наблюдений в группе значений качественной (факторной) переменной (рис. 12):

```
ggplot(investigations, aes(x=Group,
fill=Group)) +
geom_bar()
```

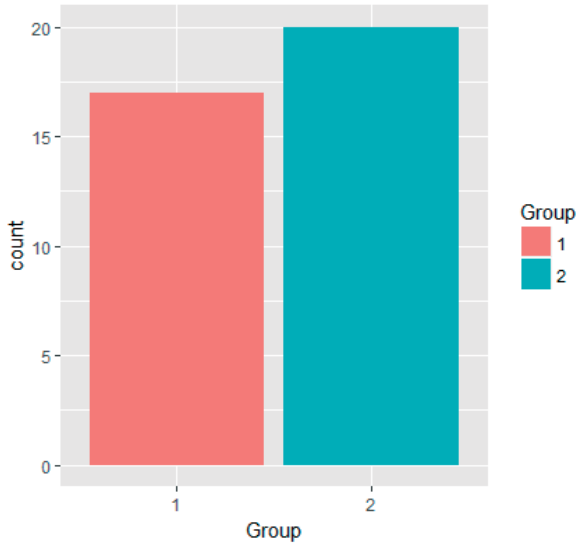

Рис. 12. Статистика количества наблюдений в группе значений качественной (факторной) переменной

Создание панелей. Одной из интересных возможностей пакета ggplot2 является представление результатов относительно подгрупп данных на соседних панелях.

У всех панелей одна и та же система координат. Благодаря этому легче сравнивать зависимости в подгруппах значений.

Создаются панели функциями facet_grid() либо facet_wrap(). Ниже представим пример, в котором каждая панель представляет отдельную группу пациентов, причем на заднем фоне панели добавлены точки, которые представляют все данные. Это облегчает анализ того, как выглядят данные показатели (амилаза, диастаза) в данной группе на фоне других групп (группы) (рис. 13):

```
ggplot(na.omit(investigations),
aes(x=log(Amylase),
y=log(Diastase))) +
stat_ellipse(color=»red4»)+
geom_point(data=investigations[,-
19],size=1,color=»grey») +
geom_point(size=2, color=»red») +
facet_wrap(~Group)
```

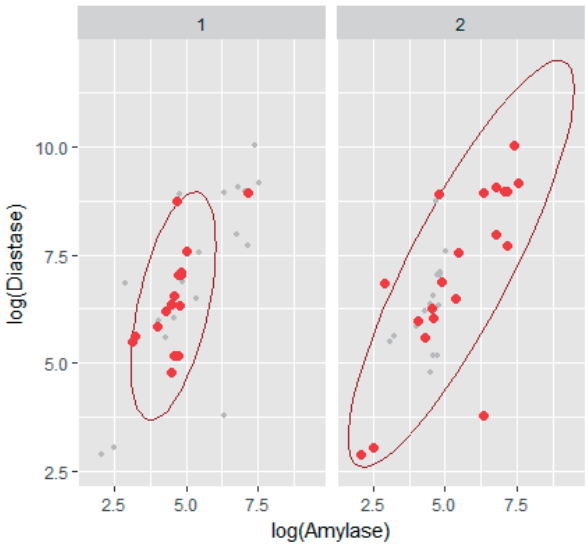

Рис. 13. Создание панелей для двух групп пациентов

Группы можно выделять также цветами, при этом используя только одну панель (рис. 14):

```
ggplot(na.omit(investigations),
aes(x=log(Amylase), y=log(Diastase),
color=Group)) +
stat_ellipse()+
geom_point(size=2)
```

Но такой график читабелен только в случае небольшого числа групп. Для большого количества групп определенно лучшим решением является использование отдельных панелей.

Изменение шкал. Описывая механизм отображений, мы указали, что достаточно определить, какая переменная должен быть отображена на

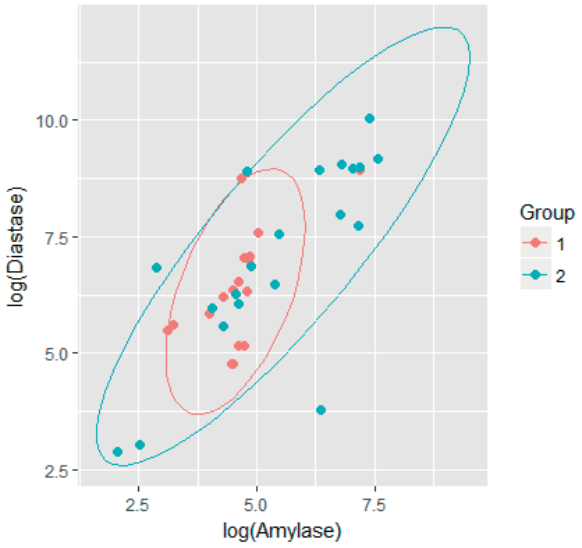

Рис. 14. Использование эллипсоидальной статистики для групп пациентов на одной панели

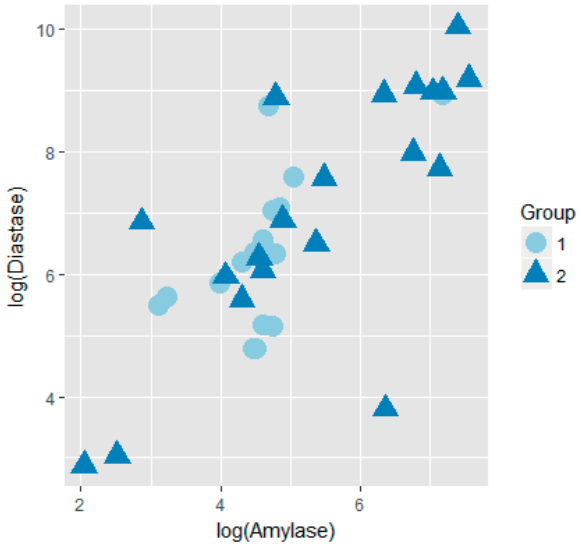

Рис. 15. Шкала для групп пациентов выбирается согласно определенной схеме

каком атрибуте, и пакет ggplot2 уже будет знать, как подобрать цвета / формы.

Однако что делать, если нам не подходит предложенная шкала цветов / форм? Решением является определение шкалы для отображения.

Функции из класса scale_атрибут_значение позволяют определить то, как должно выглядеть отображение на определенный атрибут. Мы можем вручную указать формы / цвета или мы можем указать функцию, которая за нас определит цвета / формы.

Далее мы представим три примера шкал для форм и цветов. Первый — это шкала по умолчанию (рис. 3), второй — шкала выбирается согласно определенной схеме (рис. 15):

```
pl + scale_color_brewer(type=»qual»,
palette = 3)
рис. 17:
pl + scale_shape_cleveland()
рис. 18:
pl + scale_shape_
manual(values=LETTERS)
```

И третий определяет то, какие значения должны отвечать каждой группе пациентов (рис. 16):

```
pl + scale_color_manual(values
= c(«red4», «red1», «green4»,
«green1», «grey»))
```

Модификация системы координат. На графиках данные представлены в системе координат. Эта система является каркасом для всего графика, причем по умолчанию эта система картезианская.

Однако возникают ситуации, когда мы можем захотеть этот каркас изменить. Если мы представляем отображения, то мы можем захотеть использовать другую проекцию. Мы можем захотеть поменять оси местами. Или установить, что одна из осей — логарифмическая либо в специальных единицах. Если на осях представлена одна и та же единица измерения, то мы можем пожелать, чтобы были сохранены пропорции между вертикальной и горизонтальной осями.

Все эти модификации возможны при соответствующем определении системы координат.

В пакете ggplot систему координат можно определить функцией coord_. Для одного графика определить можно только одну систему координат.

Модификация стиля графика. Кроме элементов, связанных с данными, график заключает также много графических элементов, не относящихся к данным, но также являющихся важными. Например, заглавие графика, величина описаний оси, положение легенды, цвет вспомогательных линий и тому подобное.

В пакете ggplot2 такие элементы можно доопределить двумя способами. Можно воспользоваться готовым комплектом графических установок, своеобразным каркасом. Такие каркасы доступные с помощью функций theme_.

Например, добавление к графику функции theme_ excel() влечет, что график выглядит подобно как из пакета Excel.

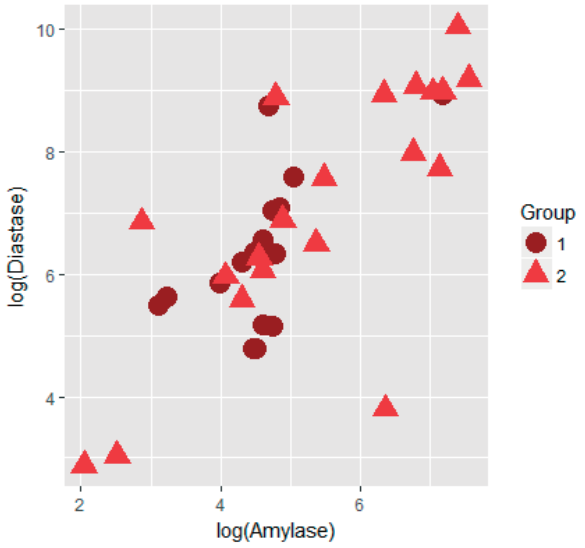

Рис. 16. Самостоятельное определение шкалы для групп пациентов

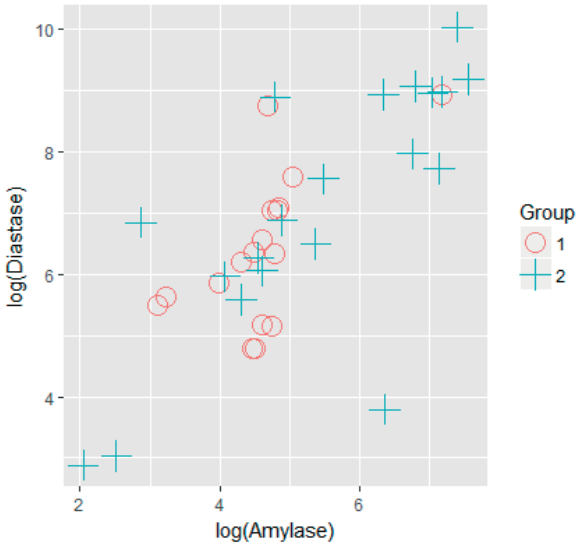

Рис. 17. Шкала для групп пациентов согласно схеме scale_shape_cleveland()

Другая возможность — это изменение отдельных элементов функцией theme(). Таким способом можно переместить легенду, изменить цвета оси, увеличить описания оси и модифицировать подобные элементы графика.

Выводы. Графическое представление данных медико-биологических исследований является важной проблемой, связанной с системным анализом и принятием решений. Это, в свою очередь, требует новых нестандартных инструментов построения, обладающих определенной гибкостью.

Такая гибкость для построения графиков может быть обеспечена с помощью специально разработанных языков программирования. Овладение языками программирования, как правило, ассоциируется с определенными трудностями, сопряженными с употреблением «технических» программных компонент.

Употребление парадигмы «грамматики графики», представленной в данной работе и реализованной в языке пакета ggplot2, позволяет получить гибкий механизм построения графиков, при этом используя общеупотребительные понятия, связанные с их контентом и внешним видом.

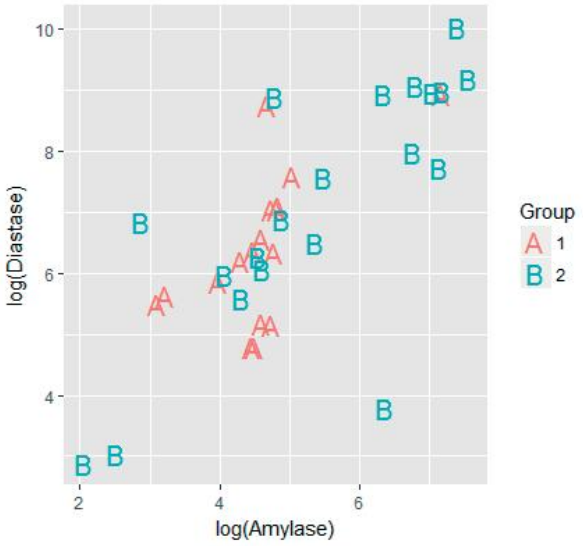

Рис. 18. Шкала для групп пациентов согласно схеме scale_shape_manual(values=LETTERS)

Литература.

1. Bertin J. Sémiologie graphique / J. Bertin. — Paris : Mouton et Gauthier-Villars, 1967. — 431 p.
2. Biecek P. Odkrywać! Ujawniać! Objaśniać! Zbiór esejów o sztuce prezentowania danych / P. Biecek. — Warszawa : Uniwersytet Warszawski, 2016. — 226 p.
3. Biecek P. Przewodnik po pakiecie R / P. Biecek. — Wrocław : Oficyna wydawnicza GiS, 2017. — (Wydanie IV rozszerzone). — 395 p.
4. Martsenyuk V. P. Information support system of medical system research / V. P. Martsenyuk, I. Ye. Andrushchak // International Journal of Medicine and Medical Research. — 2015. — Vol. 1, No. 1. — P. 63–67.
5. Wickham H. ggplot2: elegant graphics for data analysis / H. Wickham. — New York : Springer-Verlag, 2009. — 216 p.
6. Wickham H. Reshaping data with the reshape package / H. Wickham // Journal of Statistical Software. — 2007. — Vol. 21, No. 12. — P. 1–20.
7. Wilkinson L. The grammar of graphics / L. Wilkinson. — New York : Springer-Verlag, 1999. — 408 p.

Referenes.

1. Bertin, J. (1967). Sémiologie graphique. Paris: Mouton et Gauthier-Villars.
2. Biecek, P. (2016). Odkrywać! Ujawniać! Objaśniać! Zbiór esejów o sztuce prezentowania danych. Warszawa: Uniwersytet Warszawski.
3. Biecek, P. (2017). Przewodnik po pakiecie R. Wrocław: Oficyna wydawnicza GiS.
4. Martsenyuk, V. P., Andrushchak, I. Ye. (2015). Information support system of medical system research. International Journal of Medicine and Medical Research, 1(1), 63–67. doi 10.11603/ijmmr.24136077.2015.1.3285.
5. Wickham, H. (2009). ggplot2: elegant graphics for data analysis. New York: Springer-Verlag.
6. Wickham, H. (2007). Reshaping data with the reshape package. Journal of Statistical Software, 21(12), 1–20.
7. Wilkinson, L. (1999). The grammar of graphics. New York: Springer-Verlag.