федеральное государственное бюджетное образовательное учреждение высшего образования "Волгоградский государственный технический университет"

Факультет электроники и вычислительной техники

Кафедра «Системы автоматизированного проектирования и поискового конструирования»

**Лабораторная работа № 2: Визуализация данных c помощью R**

1. **Цель работы**

* Освоить основные распространенные типы графиков в R
* Приобрести основные навыки работы с пакетом ggplot2 в R

1. **Задачи**

* Углубить и закрепить знания по основным типам графикам.
* Научить строить основные типы графики в R
* Совершенствовать навыки самостоятельной работы.

1. **План выполнения работы**

Хотя пакет ggplot2 - пакет к более сложным и красивым графикам для визуализации данных в R, но это не единственный способ сделать графики в R. Для быстрого изучения данных иногда полезно использовать базовые функции в R. они устанавливаются по умолчанию с R и не требуется установить дополнительные пакеты.

Эта лаб помогает вам усвоить навыки по построению различных типов графиков в R c использованием базовых функций и пакета ggplot2.

**3.1 Scatter Plot (**диаграмма рассеяния)

- Для построения Scatter Plot ( Рис.1) воспользуемся plot():

*plot(mtcars$wt, mtcars$mpg)*

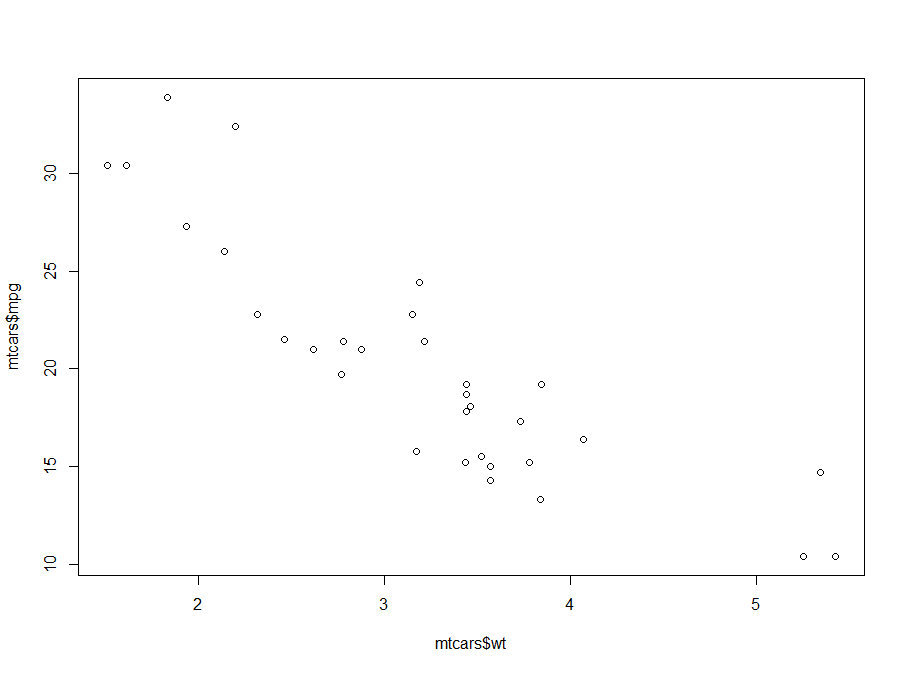


Рис 1. Scatter plot c базовой функцией plot

- С пакетом ggplot2, вы также можете получить похожий результат использованием функции qplot() (рис 2):

*library(ggplot2)*

*qplot(mtcars$wt, mtcars$mpg)*

- Если два вектора в одном фрейме данных, вы можете использовать следующий синтаксиc:

*qplot(wt, mpg, data=mtcars)*

*# это эквиваленто:*

*ggplot(mtcars, aes(x=wt, y=mpg)) + geom\_point()*

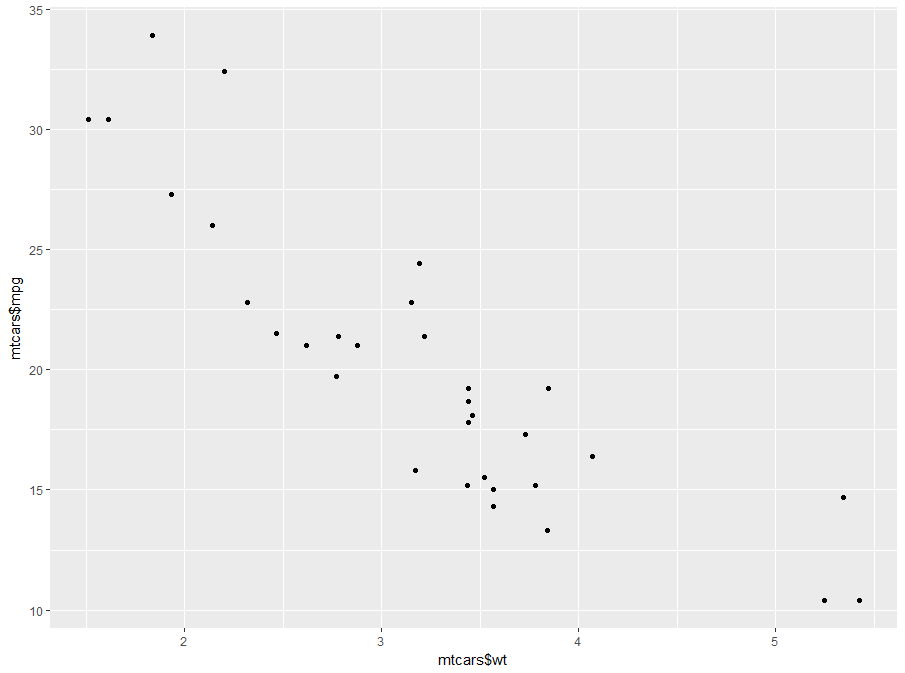


Рис 2. Scatter plot с qplot() из ggplot2

**3.2 Line Graph (Линейный график** )

- Для построения Line Graph воспользуемся plot() и указать тип графика type = “l” ( Рис.3, слева):

*plot(pressure$temperature, pressure$pressure, type="l")*

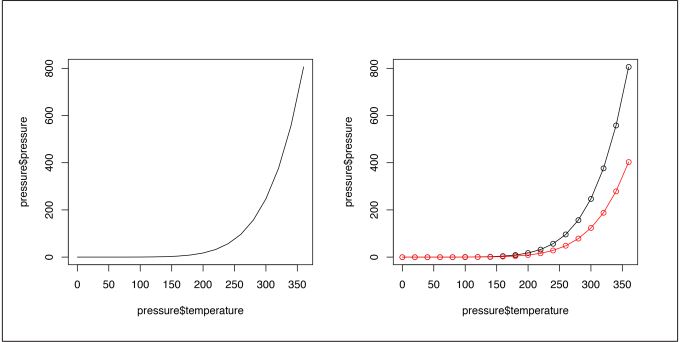


Рис 3. Line graph c базовой функцией, справа: с точками и другой линией

- Для добавления точек и/или линий (Рис.3б справа), сначала вызываем функцию plot() для первой линии, и потом добавляем точки с использованием points() и добавляем другие линии с lines():

*plot(pressure$temperature, pressure$pressure, type="l")  
points(pressure$temperature, pressure$pressure)  
lines(pressure$temperature, pressure$pressure/2, col="red")  
points(pressure$temperature, pressure$pressure/2, col="red")*

- с ggplot2, вы также можете получить похожий результат использованием функции qplot() c geom = “line” (рис. 4):

*library(ggplot2)  
qplot(pressure$temperature, pressure$pressure, geom="line")*

- Если два вектора в одном фрейме данных, вы можете использовать следующий синтаксиc:

*qplot(temperature, pressure, data=pressure, geom="line")*# This is equivalent to: *ggplot(pressure, aes(x=temperature, y=pressure)) + geom\_line()*# Lines and points together *qplot(temperature, pressure, data=pressure, geom=c("line", "point"))*# Equivalent to: *ggplot(pressure, aes(x=temperature, y=pressure)) + geom\_line() + geom\_point()*

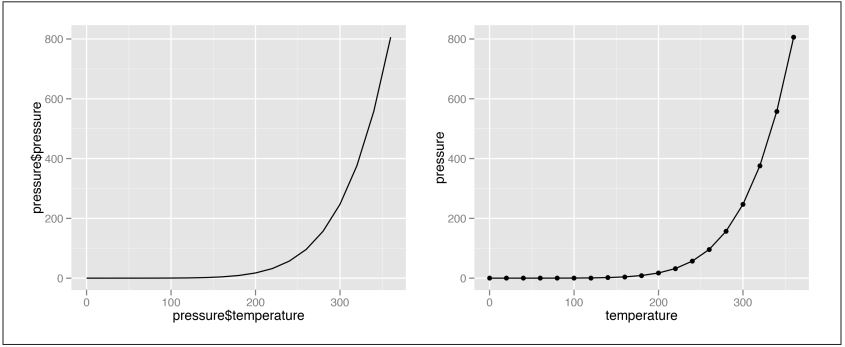
****

Рис 4. Line graph c qplot() из ggplot2, справа: с добавлением точек

**3.3 Bar Graph (столбчатая диаграмма)**

- для построения Bar Graph (рис. 5) воспользуемся barplot() и передаем ему вектор значений и вектор надписей:

*barplot(BOD$demand, names.arg=BOD$Time)*

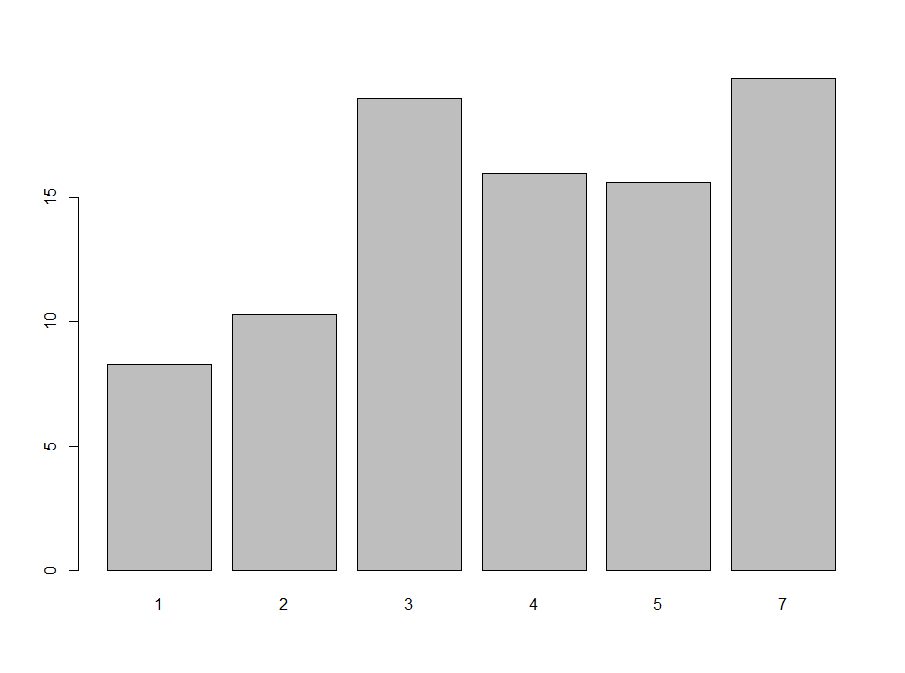


Рис 5. Bar Graph c базовой функцией barplot()

- C пакетом ggplot2 , вы также можете получить похожий результата использованием qplot(). Обратите внимание на разницу в результате, когда переменная x- непрерывная и когда она дискретная :

library(ggplot2)  
qplot(BOD$Time, BOD$demand, geom="bar", stat="identity")  
# Convert the x variable to a factor, so that it is treated as discrete  
qplot(factor(BOD$Time), BOD$demand, geom="bar", stat="identity")

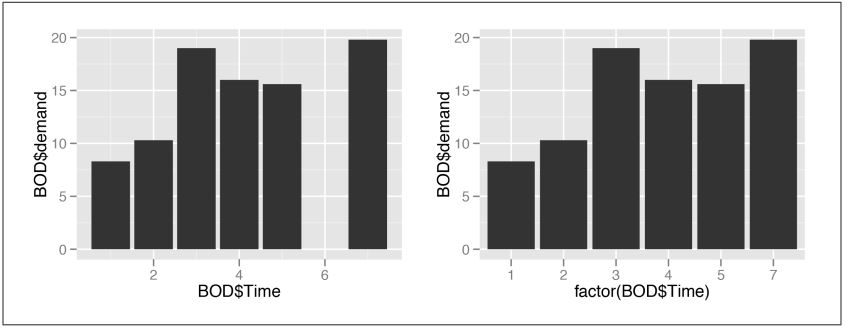
****

Рис 6. Bar Graph c qplot() c непрерывной переменной x, справа: c переменной, преобразующей в фактор (здесь нет надписи 6)

- Если ветор в фрейме данных, вы можете использовать следующий синтаксис:

*qplot(Time, demand, data=BOD, geom="bar", stat="identity")*# This is equivalent to: *ggplot(BOD, aes(x=Time, y=demand)) + geom\_bar(stat="identity")*

**3.4 Histogram (гистограмма)**

- Для построения Histogram (рис.7) воспользуемся hist():

*hist(mtcars$mpg)*# Specify approximate number of bins with breaks *hist(mtcars$mpg, breaks=10)*

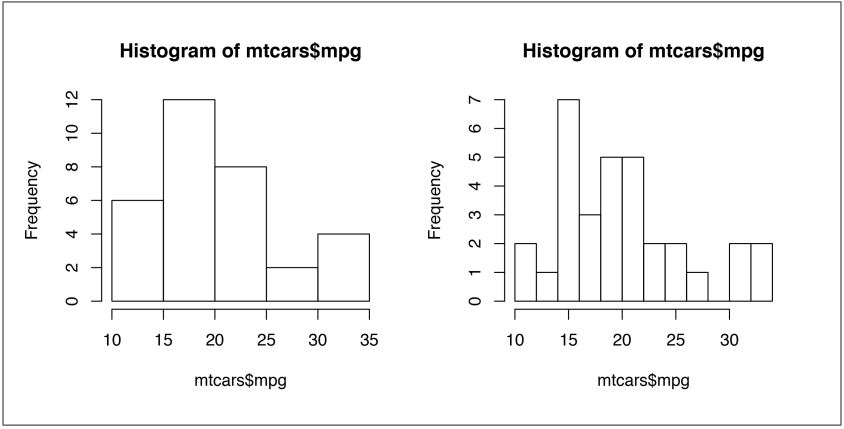
****

Рис 7. Histogram с базовой функцией, справа: с более ящиками

- С пакетом ggplot2 вы также можете получить похожий результат использованием qplot() (рис. 8):

*qplot(mtcars$mpg)*

- Если вектор в фрейме данных вы можете использовать следующий синтаксис:

*library(ggplot2)  
qplot(mpg, data=mtcars, binwidth=4)*# This is equivalent to: *ggplot(mtcars, aes(x=mpg)) + geom\_histogram(binwidth=4)*

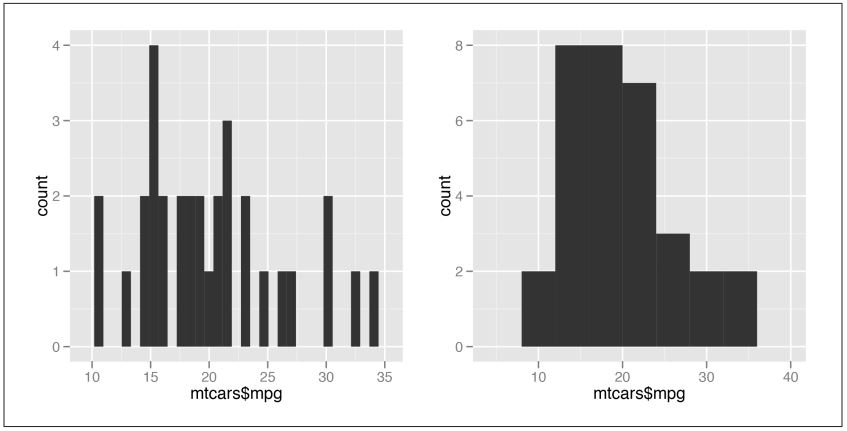
**

Рис 8. Histogram с qplot () из ggplot2 c шириной ящика по умолчанию, справа: с широками ящиками.

**3.5 Box Plot (ящики с усами)**

- Для построения box plot (рис. 9) воспользуемся plot(). Если x – фактор, то функция plot() автоматически строит box plot:

*plot(ToothGrowth$supp, ToothGrowth$len)*

-Если 2 вектора в одном фрейме данных, вы можете использовать следующий синтаксис (рис. 9):

*# Formula syntax  
boxplot(len ~ supp, data = ToothGrowth)  
# Put interaction of two variables on x-axis  
boxplot(len ~ supp + dose, data = ToothGrowth)*

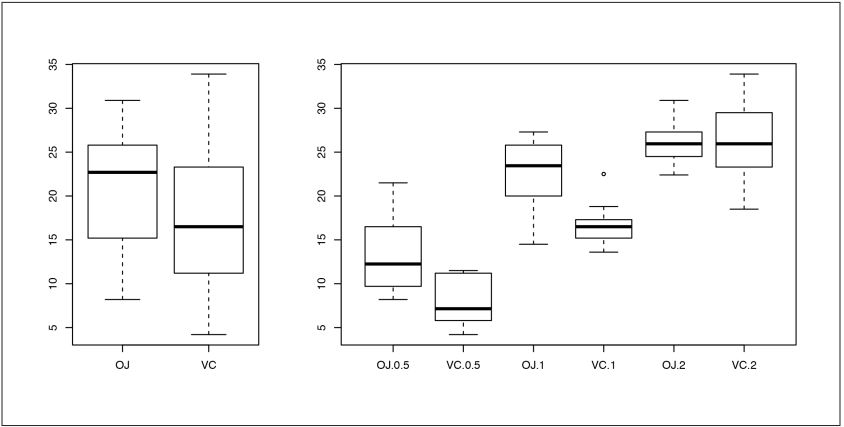
**

Рис 9. Слева: box plot c базовой функцией, справа: с несколькими группирующими переменными

- С пакетом ggplot2 вы также можете получить похожий результат использованием qplot() (рис. 10) c geom = “boxplot”:

*library(ggplot2)  
qplot(ToothGrowth$supp, ToothGrowth$len, geom="boxplot")*

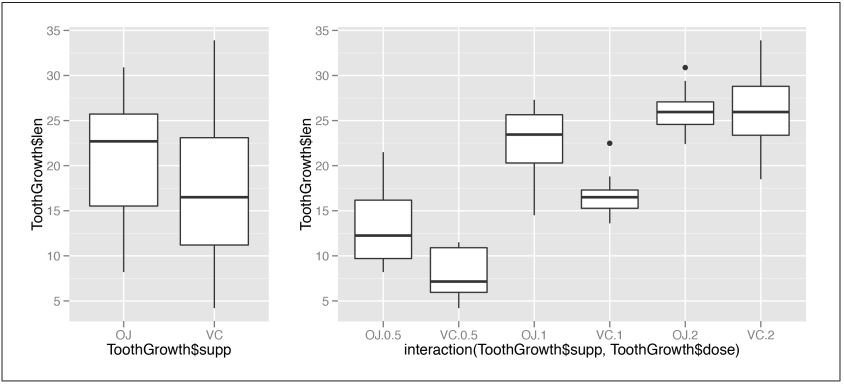
**

Рис 10. Слева: box plot c qplot() из ggplot2, справа: с несколькими группирующими переменными

-Если 2 вектора в одном фрейме данных, вы можете использовать следующий синтаксис:

*qplot(supp, len, data=ToothGrowth, geom="boxplot")*# This is equivalent to: *ggplot(ToothGrowth, aes(x=supp, y=len)) + geom\_boxplot()*