Lecture1 R编程基础-1

李素科　博士

北京大学软件与微电子学院

September 11, 2025

outline

- 课程介绍
- R基本语法
- 使用ggplot2可视化数据

课程介绍

- 本课讲授一种数据分析工具，R。
- 课程总时长32学时。
- 教师联系方式:lisuke@ss.pku.edu.cn。

学习R的好处

- 掌握一门统计编程语言。
- 掌握一些编程背后的知识。

不涉及的内容

- 大数据
- 深度学习
- 软件工程

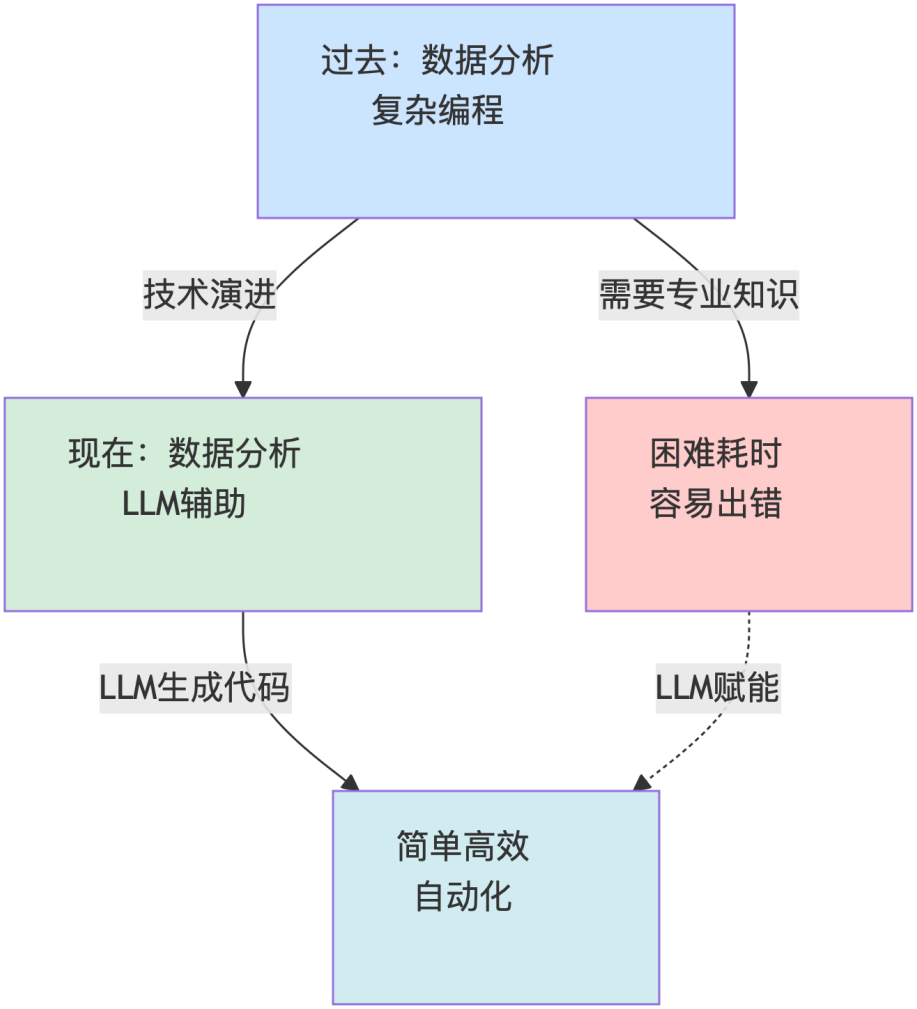
学习方式

- 开放式学习，自主学习，实验学习，验证式学习
- Learn by doing

评分方法

- 考勤+期末作业。
- 包括3次考勤，考勤是随机的，可能考勤也可能不考勤。(30%)。
- 期末作业可选：2-4个人一组做一个大作业，题目自选(和金融经济相关)。期末项目提交包括源代码和演讲PPT。(70%)。
- LLM 让我们的学习发生了根本变化。LLM可以生成代码。那我们学习什么？我们主要关注
  - 能读懂代码，不能完全相信LLM
  - 知道代码解决了什么问题，以及背后的统计知识
  - 明确我们要解决什么问题，代码充当有用的工具

数据分析范式发生变化


过去：数据分析
复杂编程
技术演进
需要专业知识
现在：数据分析
LLM辅助
困难耗时
容易出错
LLM生成代码
LLM赋能
简单高效
自动化

课程讲义

https://course.pku.edu.cn

参考书

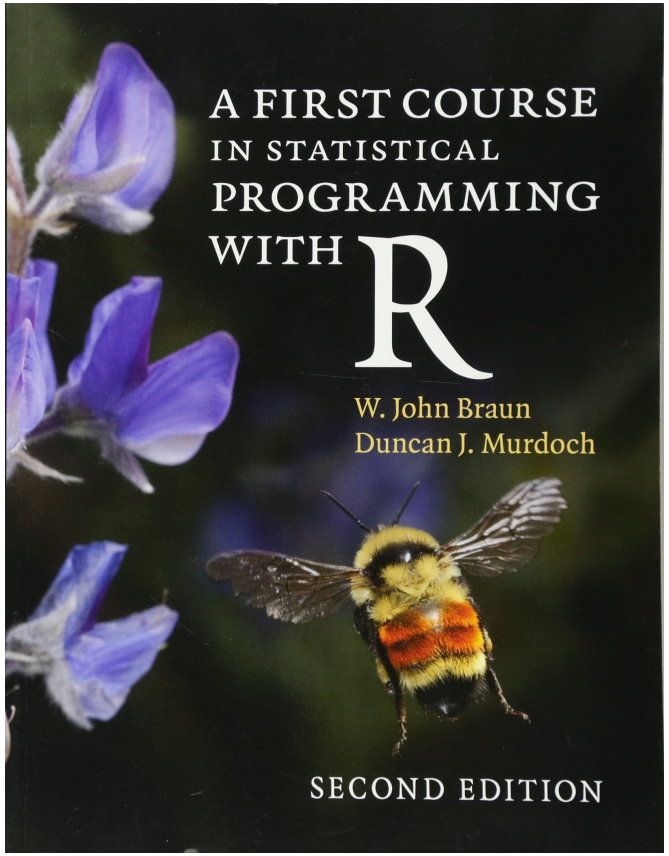


Figure: A First Course in Statistical Programming with R，W. John Braun and Duncan J. Murdoch. Cambridge University Press. Second edition 2016

参考书

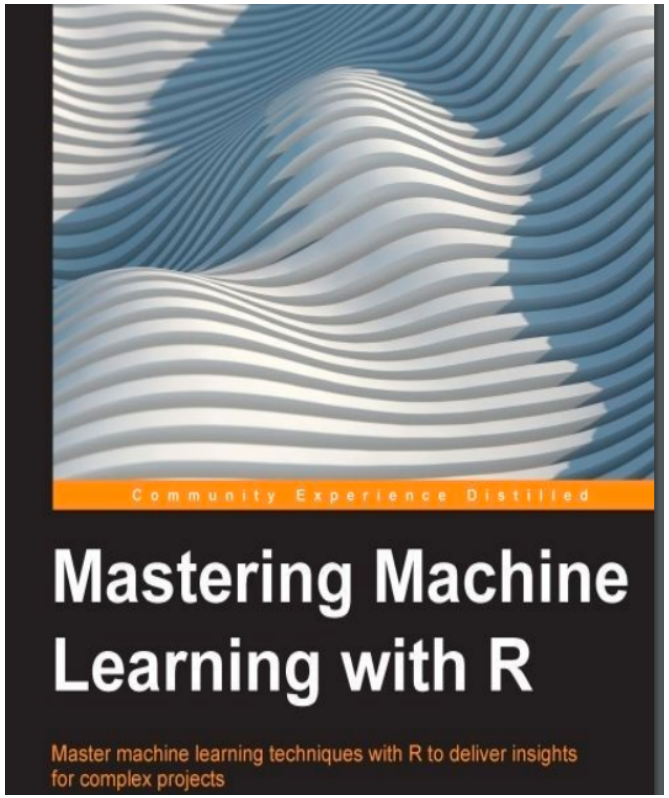


Figure: Mastering Machine Learning with R

参考书

Figure: R for Data Science Import, Tidy, Transform, Visualize, and Model Data

参考书

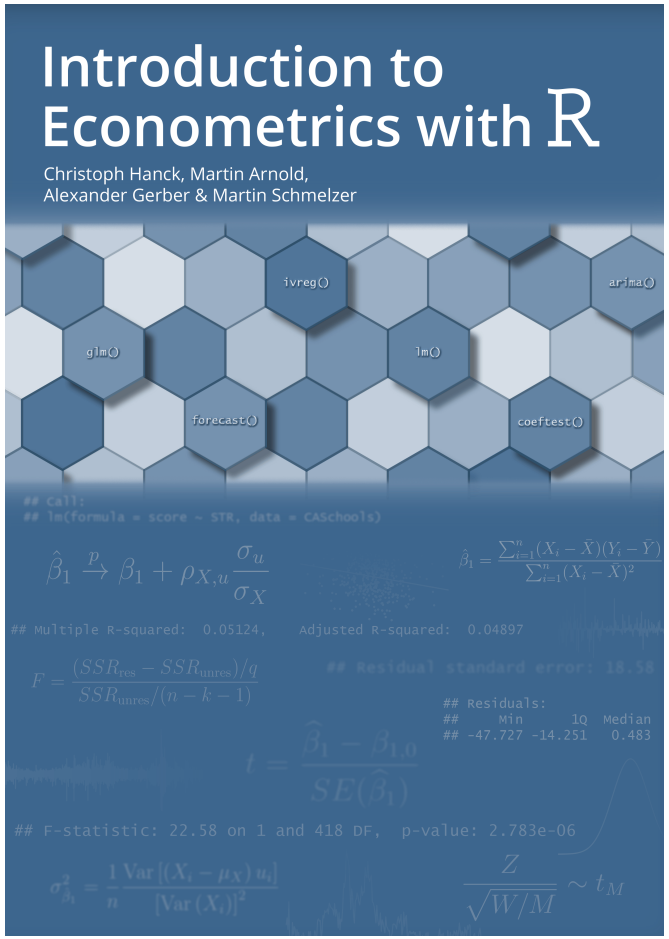


Figure: Introduction to Econometrics with R
https://www.econometrics-with-r.org/

授课内容

- Lecture1, R编程基本语法-1
- Lecture2, R编程基本语法-2
- Lecture3, 使用R计算收益率、债券相关收益率、基本统计模型
- Lecture4, 探索数据分析，单变量统计模型、多变量统计模型
- Lecture5, 使用R进行机器学习实践-1
- Lecture6, 使用R进行机器学习实践-2
- Lecture7, 使用R的回归分析
- Lecture8, 基于R的时间序列分析
- Lecture9, 基于R的算法交易、风险管理、衍生品定价
- Lecture10, 基于R的投资组合优化
- Lecture11, 学生期末作业演讲

outline

- 课程介绍
- **R基本语法**
- 使用ggplot2可视化数据

什么是R?

- R是统计计算和图形的语言和环境。它是一个GNU项目，类似于S语言和环境，S是由John Chambers及其同事在贝尔实验室（前身为AT&T，现为朗讯科技）开发。R可以被认为是S的不同实现。存在一些重要的差异，但是为S编写的许多代码在R下运行不变。
- R是由Ross Ihaka 和Robert Gentleman 于1993年开发的，当时他们都在新西兰奥克兰大学统计系工作。

什么是R?

- R提供各种统计（线性和非线性建模，经典统计检验，时间序列分析，分类，聚类.....）和图形技术，并且具有高度可扩展性。
- 如何获得R？https://www.r-project.org/

r-project

https://www.r-project.org/

[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Get Involved: Contributing
Developer Pages
R Blog

R Foundation

Foundation
Board
Members
Donors
Donate

Help With R

Getting Help

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- **R version 4.5.1 (Great Square Root)** has been released on 2025-06-13.
- **R version 4.5.0 (How About a Twenty-Six)** has been released on 2025-04-11.
- **R version 4.4.3 (Trophy Case)** (wrap-up of 4.4.x) was released on 2025-02-28.
- The **useR! 2025** conference will take place at Duke University, in Durham, NC, USA, August 8-10.
- We are deeply sorry to announce that our friend and colleague Friedrich (Fritz) Leisch has died. Read our tribute to Fritz here.
- You can support the R Foundation with a renewable subscription as a supporting member.

News via Mastodon

```
<div class="loading-spinner"></div>
```

rstudio

https://posit.co/download/rstudio-desktop/

Introducing Positron, the free code editor made for data science with Python, R, and you. EXPLORE POSITRON

posit PRODUCTS FREE & OPEN SOURCE USE CASES PARTNERS LEARN & SUPPORT ABOUT

1: Install R

RStudio requires R 3.6.0+. Choose a version of R that matches your computer's operating system.

R is not a Posit product. By clicking on the link below to download and install R, you are leaving the Posit website. Posit disclaims any obligations and all liability with respect to R and the R website.

DOWNLOAD AND INSTALL R

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR MACOS 13+

This version of RStudio is only supported on macOS 13 and higher. For earlier macOS environments, please download a previous version.

Size: 607.30 MB | SHA-256: 76E1538B | Version: 2025.05.1+513 | Released: 2025-06-05

为什么使用用R?

- R 允许与其他语言集成，如C/C++、Java、Python 等。
- R 拥有一个充满活力、不断增长的在线用户社区。
- CRAN（The Comprehensive R Archive Network）有超过8000 多个可用软件包。
- R 与平台无关，因此您可以在任何操作系统上使用它。

R的特点

- R为统计及数据科学而生。
- R简单，易学易用。
- R在科学研究领域、金融经济领域被大量地使用。
- R提供了丰富的可视输出。
- R不是为了构建软件或者系统而设计的语言。
- R的弱点：不适合开发大规模系统、不适合分析大规模数据、不适合深度学习。适合日常单机版小规模数据统计分析与可视化。

R的可视化功能

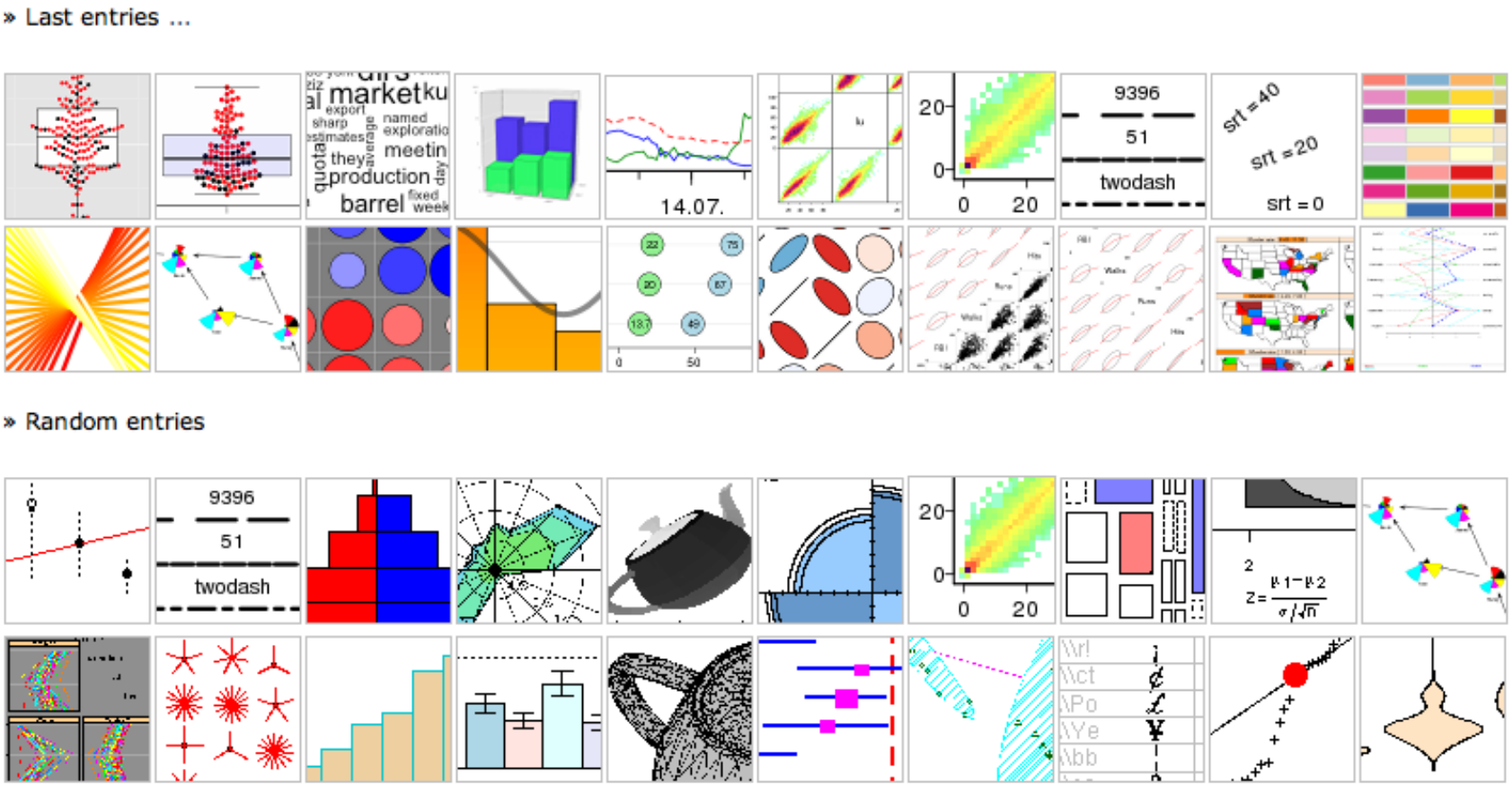


Figure: 图片来源：互联网

R的可视化功能

Figure: 图片来源：互联网

R的可视化功能

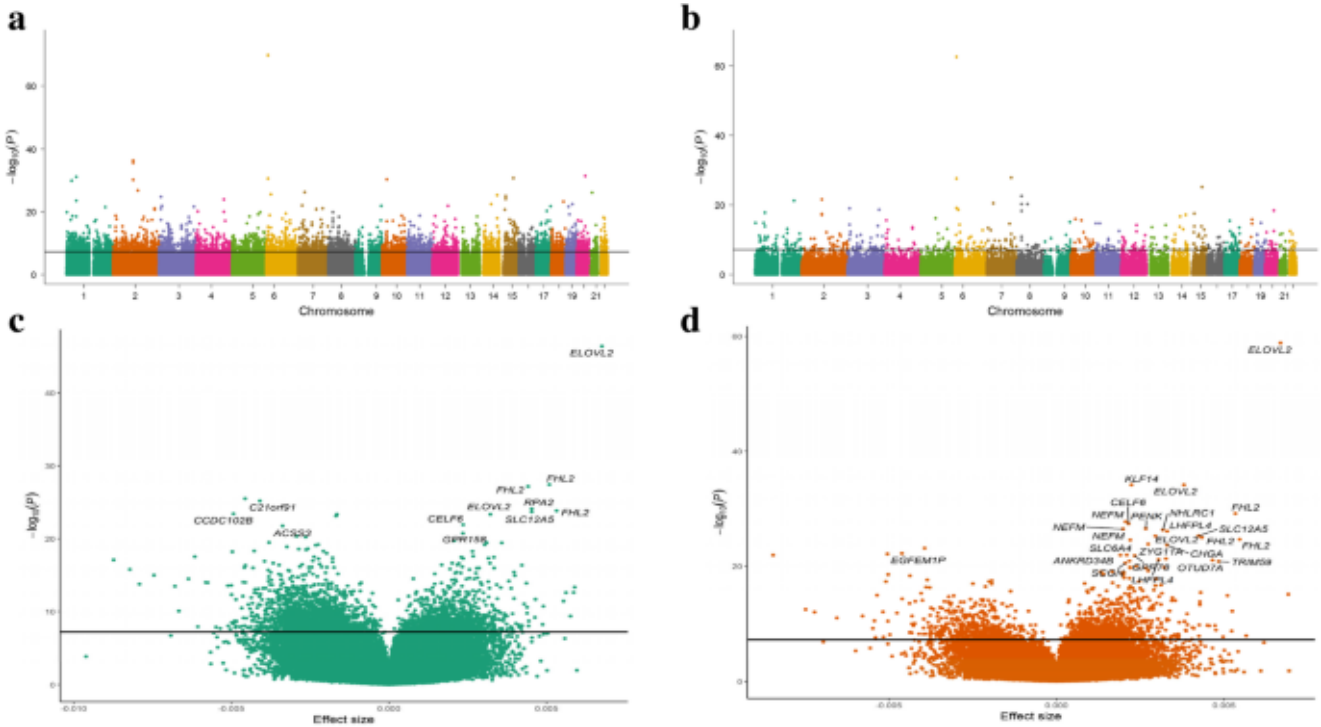


Figure: 图片来源：互联网

R的可视化功能

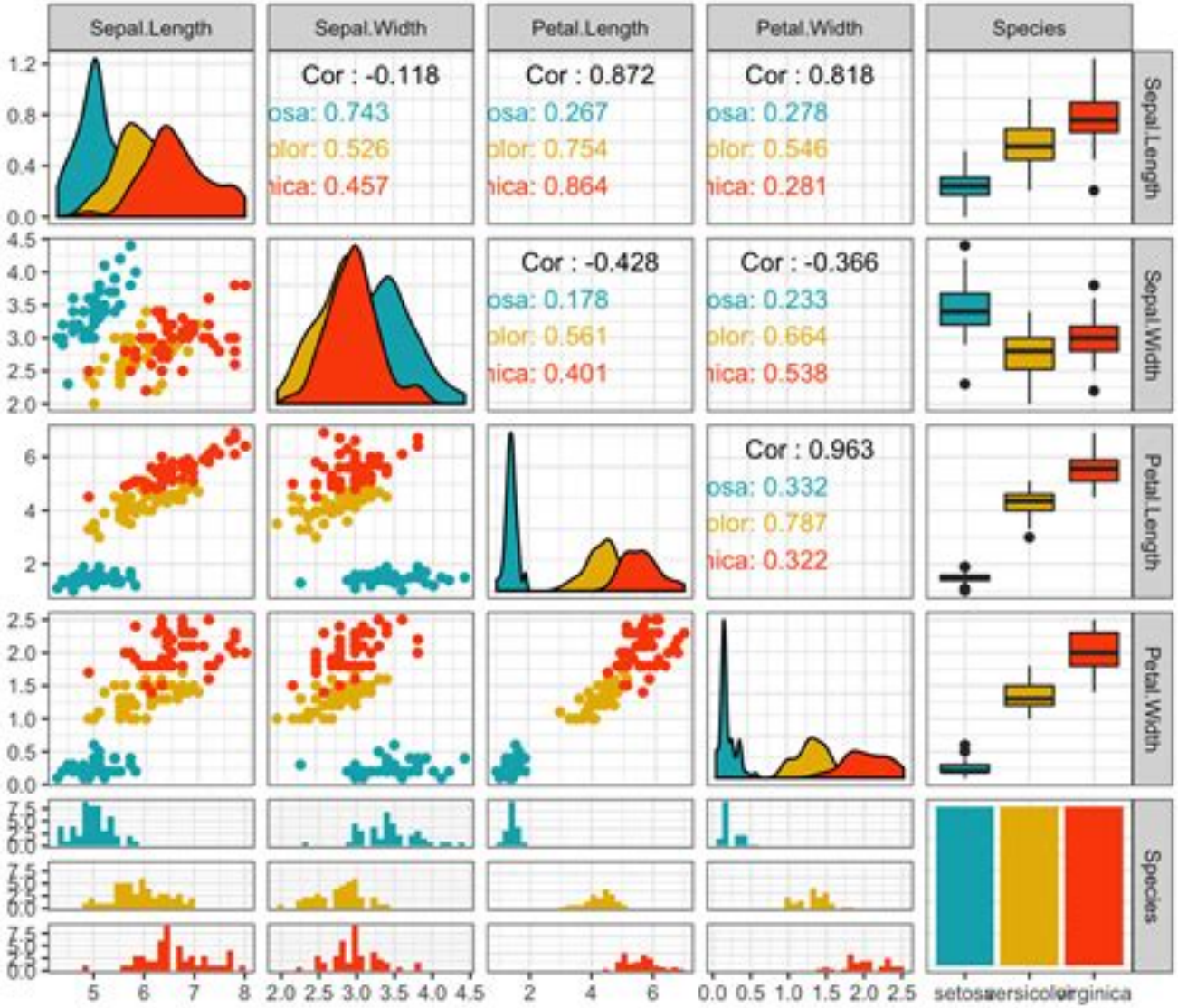


Figure: 图片来源：互联网

R的可视化功能

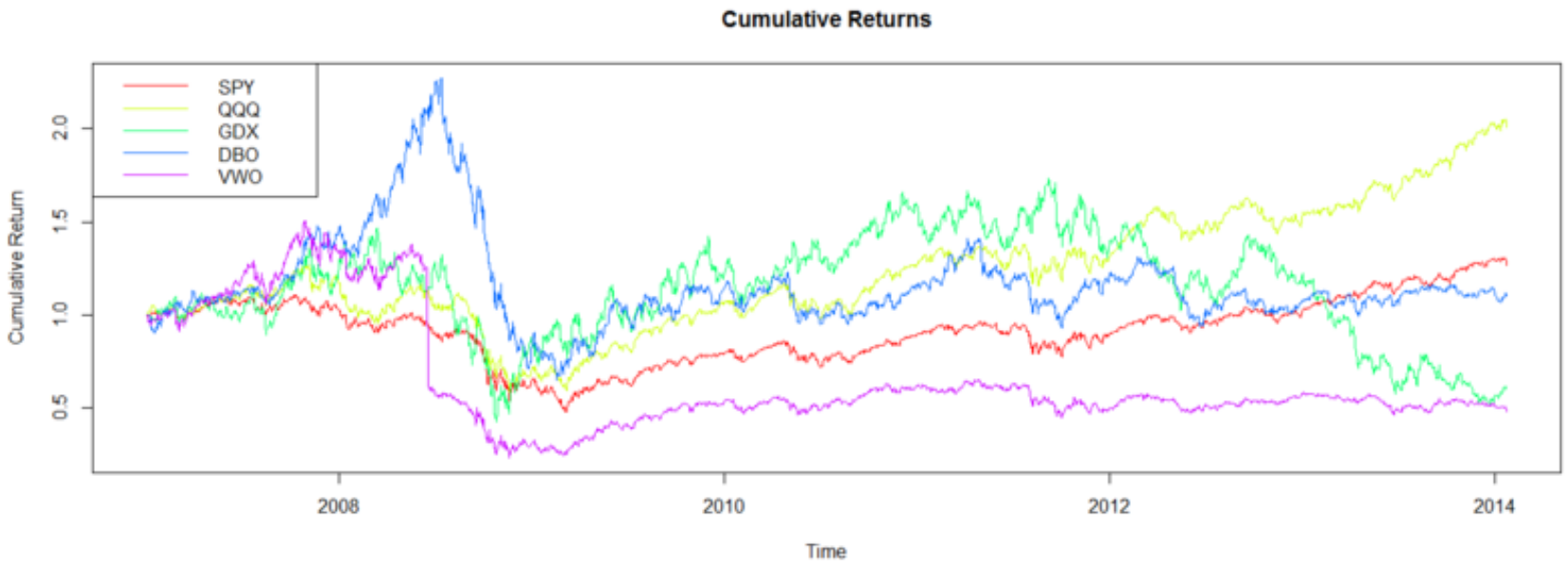


Figure: 图片来源：互联网

R的可视化功能

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Sign up for alerts RSS feed

nature › articles › article

Article | Published: 27 April 2022

ABO genotype alters the gut microbiota by regulating GalNAc levels in pigs

Hui Yang, Jinyuan Wu, ... Lusheng Huang + Show authors

Nature **606**, 358–367 (2022) | Cite this article

14k Accesses | **3** Citations | **92** Altmetric | Metrics

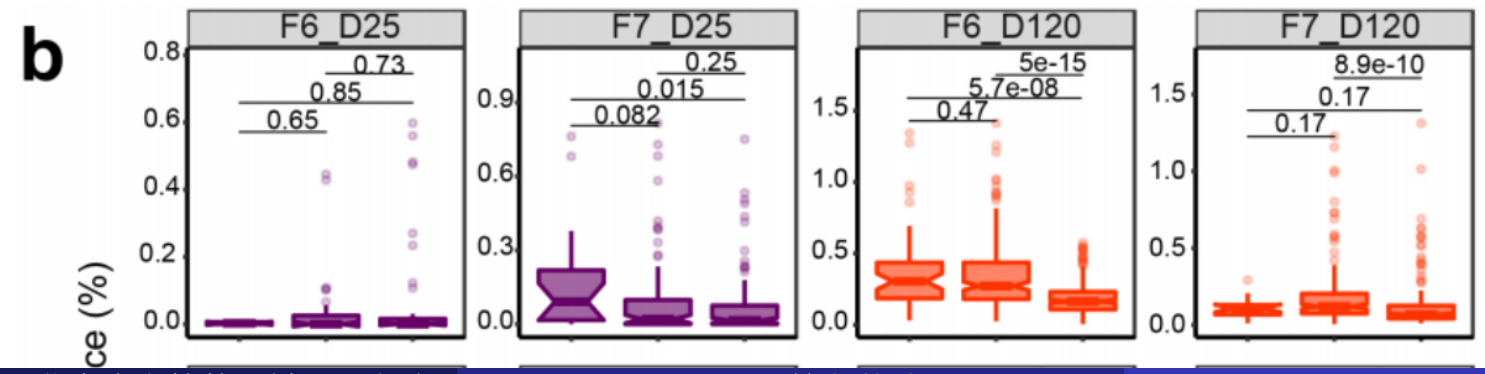
Abstract

The composition of the intestinal microbiome varies considerably between individuals and is correlated with health[1]. Understanding the extent to which, and how, host genetics contributes to this variation is essential yet has proved to be difficult, as few associations have been replicated, particularly in humans[2]. Here we study the effect of host genotype on the composition of the intestinal microbiota in a large mosaic pig population. We show that, under conditions of exacerbated genetic diversity and environmental uniformity, microbiota composition and the abundance of specific taxa are heritable. We map a quantitative trait locus affecting the abundance of Erysipelotrichaceae species and show that it is caused by a 2.3 kb deletion in the gene encoding *N*-acetyl-galactosaminyl-transferase that underpins the

Download PDF

Sections | Figures | References

Abstract
Main
A mosaic pig population to study complex traits
The intestinal microbiota of the healthy pig
Microbiota heritability in mosaic pigs
A miQTL affecting Erysipelotrichaceae species
A 3.5-million-year-old deletion in the pig *ABO* orth...
The miQTL affects caecal GalNAc concentrations
miQTL-responsive bacteria use GalNAc
GalNA
Discu
Meth



统计编程

- 以编程的方式进行统计分析。（另一种方式是点击鼠标操作软件进行数据分析。）
- 模型必须拟合数据，并显示结果。
- Microsoft Excel, SAS, SPSS,S-PLUS, R, Stata, etc。这些工具基本都提供编程接口。

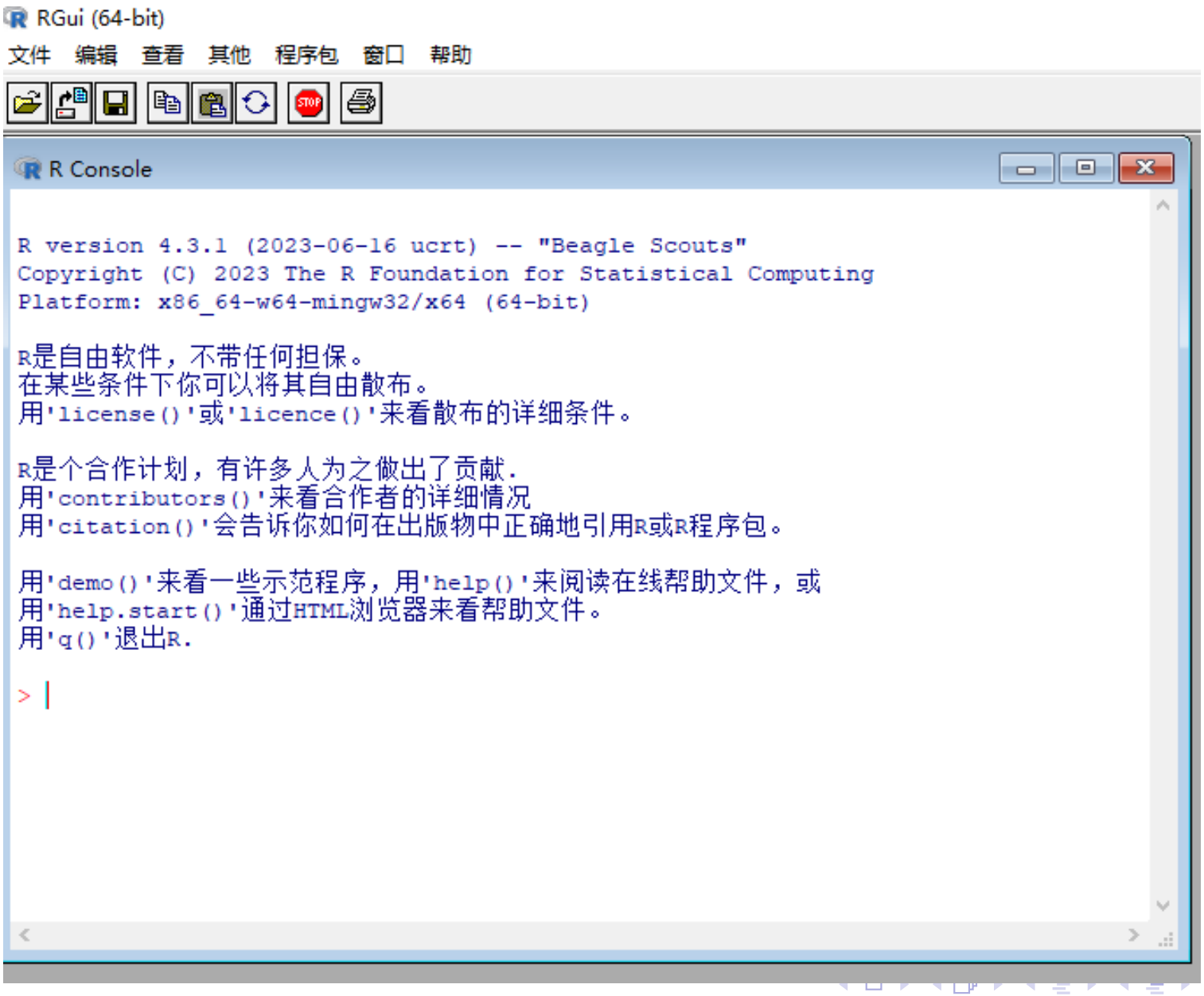
R编程要素

- 语言本身的语法结构
- 数据表达、运算及存储结构
- 支持各种算法的软件包

使用R语言的任务

- 数据的获取、读写、清洗、选择
- 模型选择、模型对比
- 相关软件包选择与使用
- 数据输出、可视化结果展示

运行界面


RGui (64-bit)
文件 编辑 查看 其他 程序包 窗口 帮助
R Console
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
R是自由软件，不带任何担保。
在某些条件下你可以将其自由散布。
用'license()'或'licence()'来看散布的详细条件。
R是个合作计划，有许多人为之做出了贡献.
用'contributors()'来看合作者的详细情况
用'citation()'会告诉你如何在出版物中正确地引用R或R程序包。
用'demo()'来看一些示范程序，用'help()'来阅读在线帮助文件，或
用'help.start()'通过HTML浏览器来看帮助文件。
用'q()'退出R.
>

R是命令驱动的语言

- 直接在界面输入命令，命令数量不受限制，输入随心所欲。
- RStudio是R的集成开发环境，www.rstudio.com

RStudio

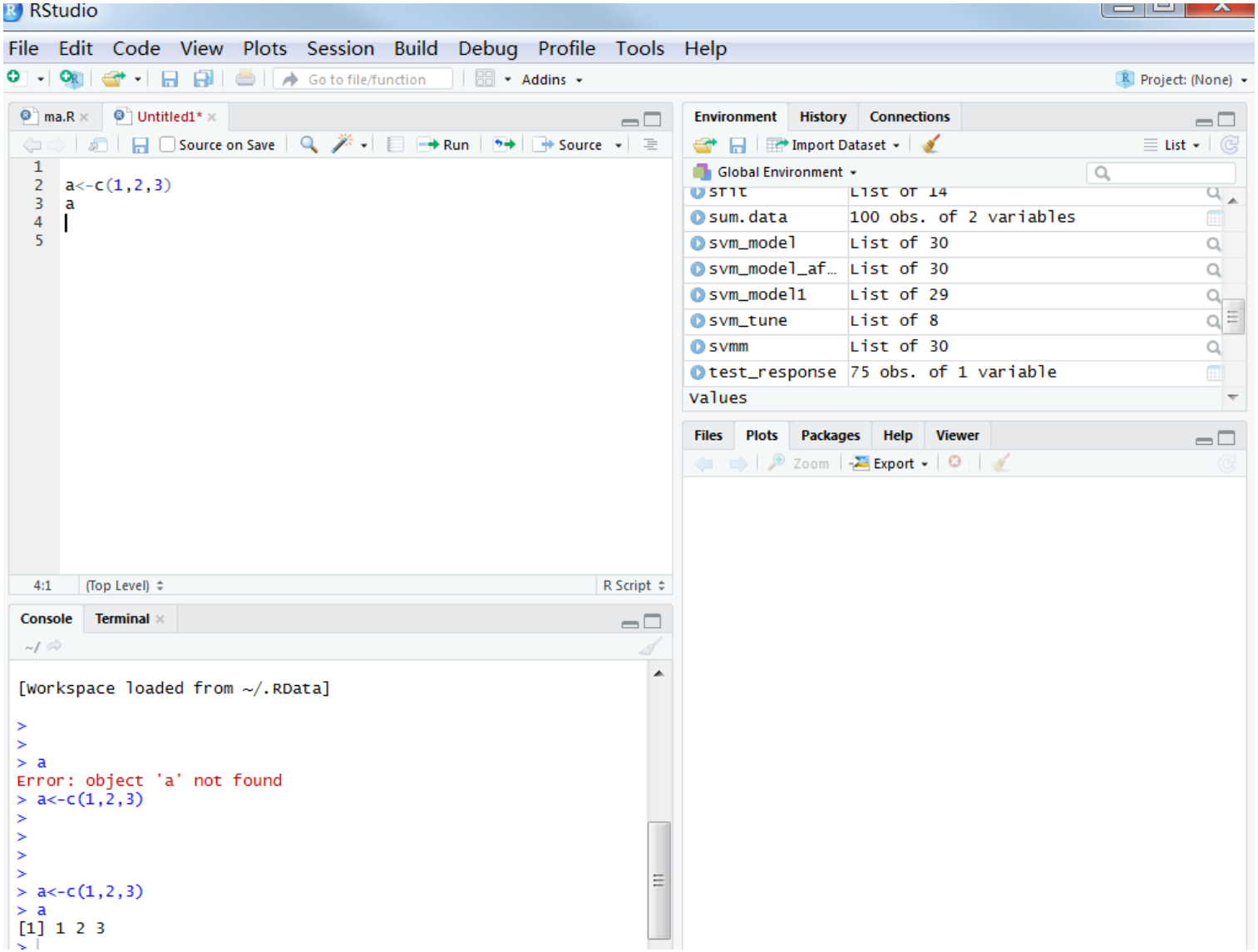


Figure: RStudio的运行界面

vscode

test.r

Rcode > test.r > ...

```r
a <- c(1,2,3)
b <- c(2,3,4)
print(a+b)
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS 4: R Interactive

```
source /Users/lisuke/Documents/llm/DataScience/venv/bin/activate

R version 4.5.0 (2025-04-11) -- "How About a Twenty-Six"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

source("/Users/lisuke/Documents/llm/DataScience/Rcode/test.r", encoding = "UTF-8")
> source /Users/lisuke/Documents/llm/DataScience/venv/bin/activate
Error: object 'Users' not found
> source("/Users/lisuke/Documents/llm/DataScience/Rcode/test.r", encoding = "UTF-8")
[1] 3 5 7
> source("/Users/lisuke/Documents/llm/DataScience/Rcode/test.r", encoding = "UTF-8")
[1] 3 5 7
>
```

vscode

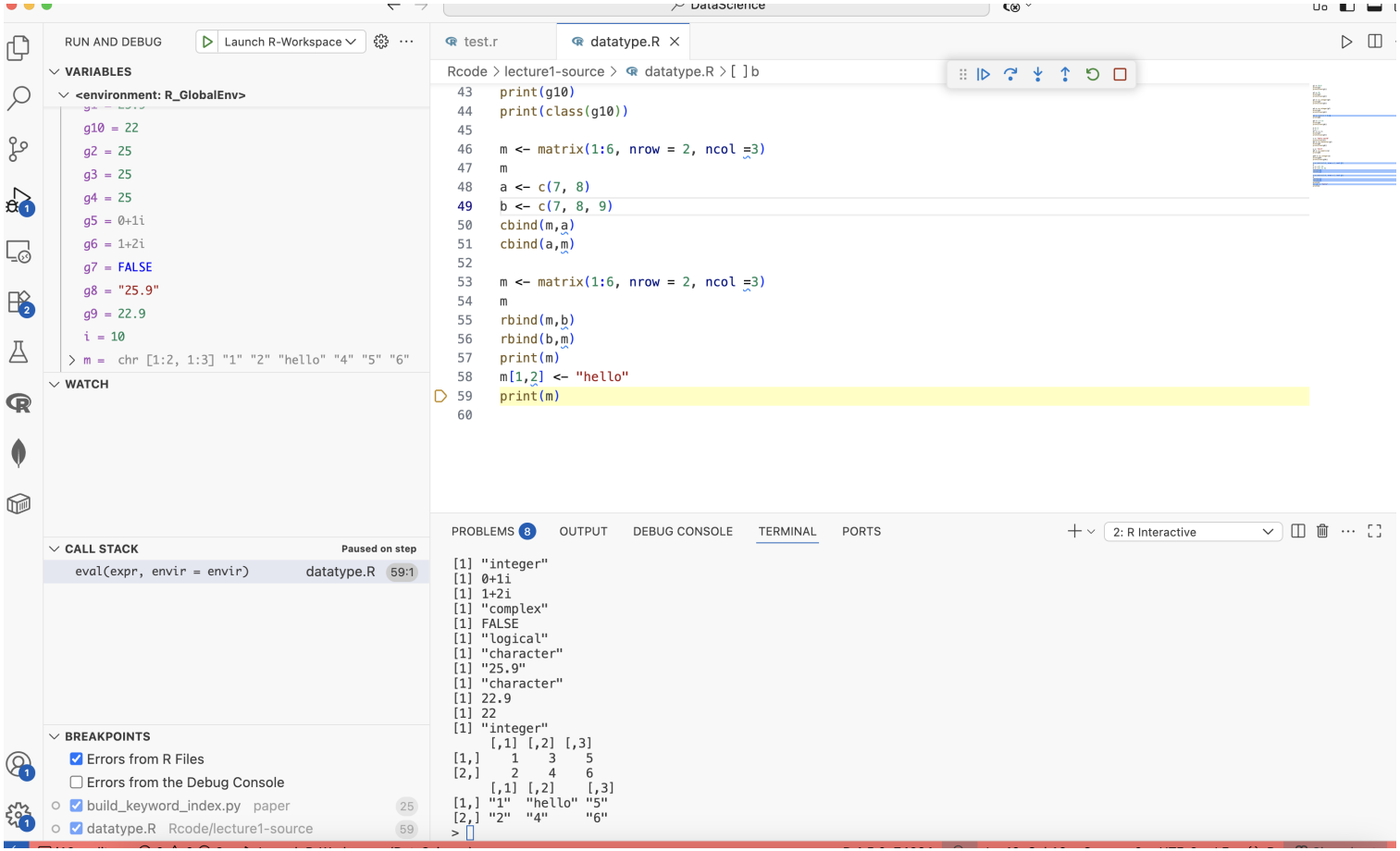


Figure: vscode的运行界面

R/Finance Conference

RFinance

Home | About | Call for Presenters | Sponsors | Code of Conduct | Diversity | Archive

Applied Finance with R

R/Finance is back, live and person for 2022! Join us in Chicago this Spring for the 14th anniversary of the conference that brings together some of the best people and conversation in quantitative finance – covering R (or Python or Julia!), portfolio construction, statistics, quant ideas and more.

We hope to see you this June 3rd and 4th at the University of Illinois at Chicago:

R/Finance 2022 Call For Papers

There is still time to submit a paper or abstract to speak. We are especially interested in lightning talks at this point - if you have an interesting project, problem, or solution, feel free to submit your abstract to the link above.

We will update this page when registration links open.

Looking forward to seeing everyone in Chicago in June!

Best,

R/Finance Committee

Figure: http://www.rinfance.com

观察某些R内置的数据

输入:data()

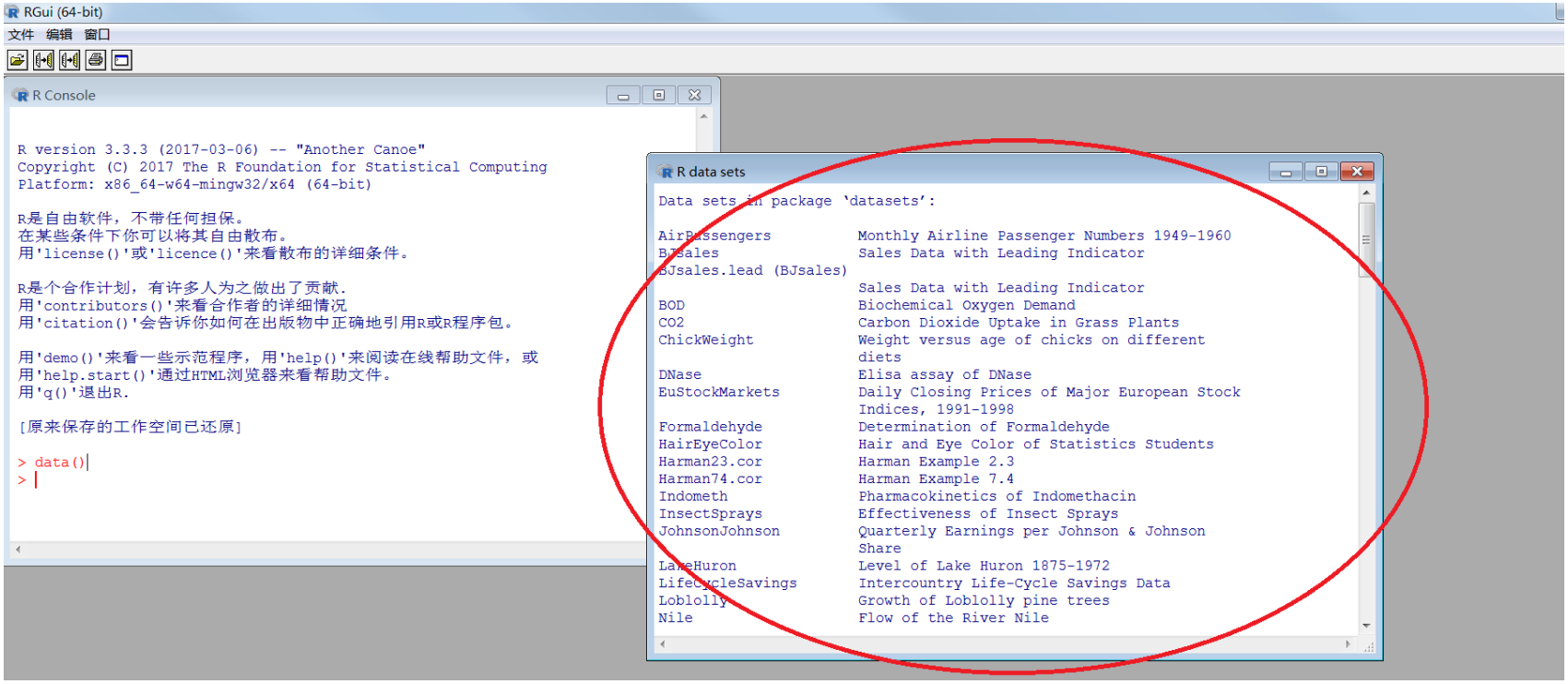
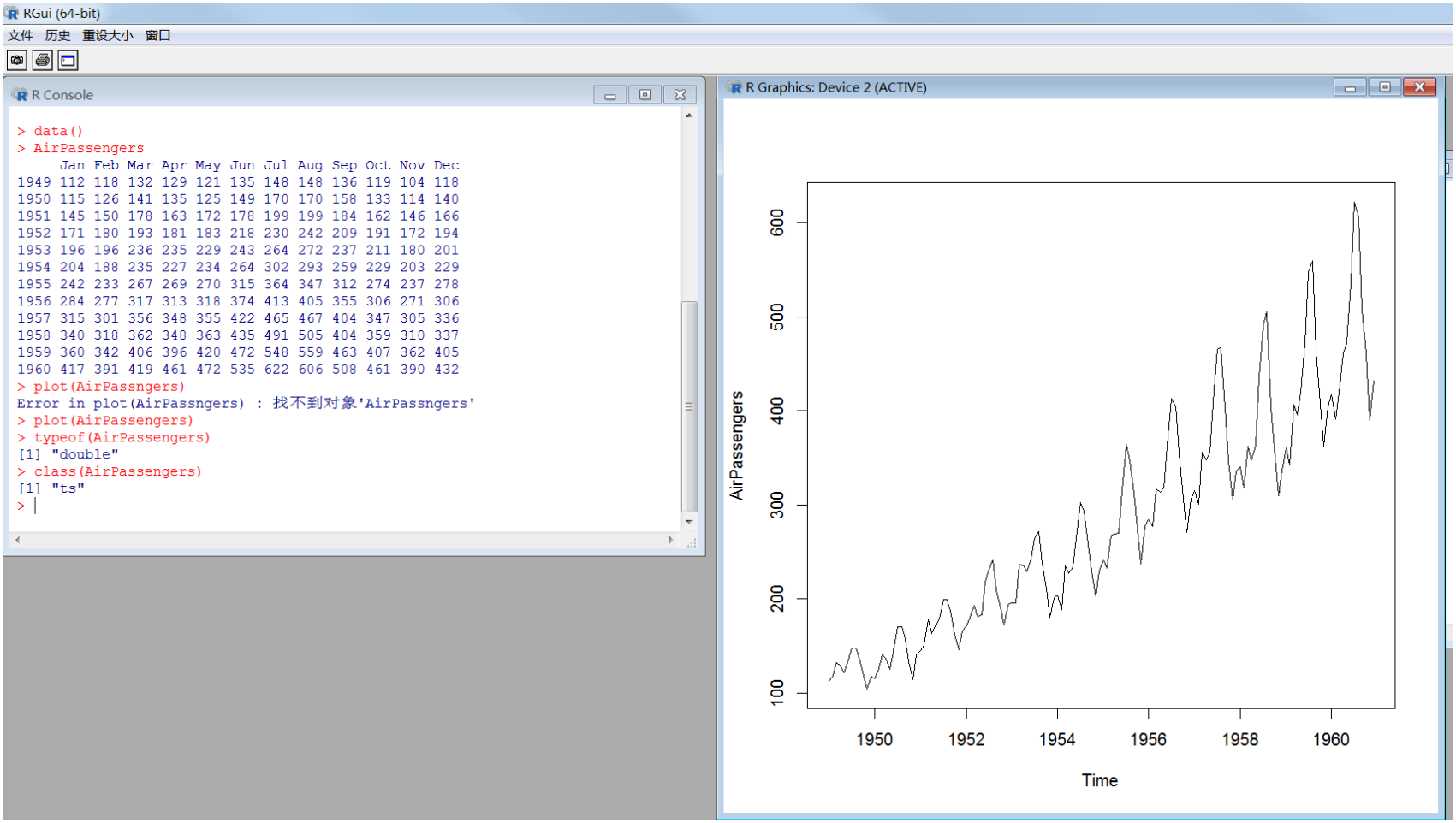


Figure: R的运行界面-命令行式的

观察时间序列AirPassengers

输入:plot(AirPassengers)



一些常用的R命令

- 设置工作目录
  > setwd("C:/R-data")
- 罗列在会话（session）中的对象
  > ls()
- 删除对象
  >rm(myobject)
- 运行某一个源代码
  >source("mysourcecode.R")
- 退出R
  >q()

得到帮助文档

```
>?t.test
> help(t.test)
> example(t.test)
> t.test
```

帮助文档

t.test {stats}

Student'

Description

Performs one and two sample t-tests on vectors of data.

Usage

```
t.test(x, ...)

## Default S3 method:
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)

## S3 method for class 'formula'
t.test(formula, data, subset, na.action, ...)
```

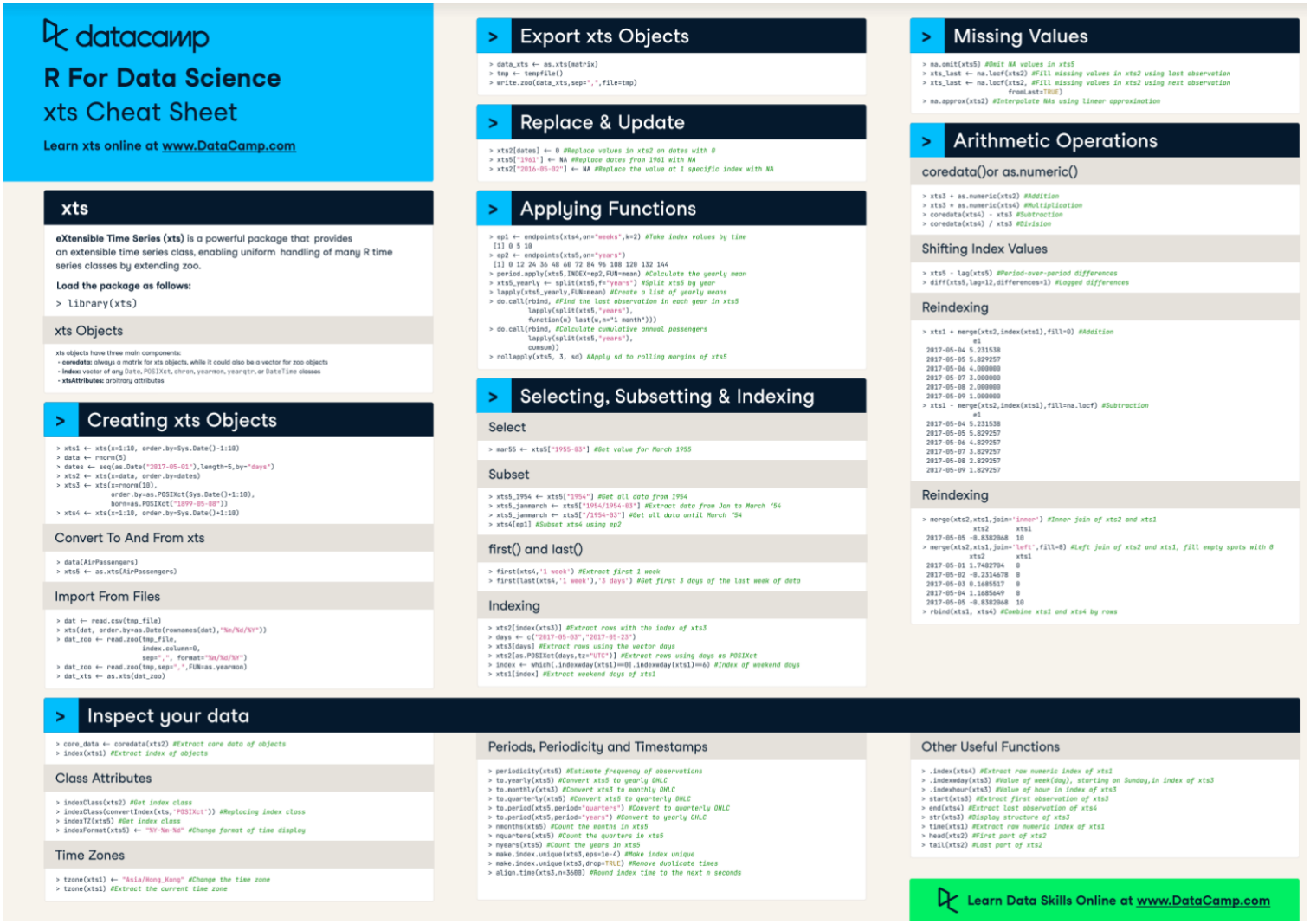
Arguments

`x`
: a (non-empty) numeric vector of data values.

`y`
: an optional (non-empty) numeric vector of data values.

`alternative`

备忘单



datacamp

R For Data Science
xts Cheat Sheet

Learn xts online at www.DataCamp.com

xts

eXtensible Time Series (xts) is a powerful package that provides an extensible time series class, enabling uniform handling of many R time series classes by extending zoo.

Load the package as follows:

```
> library(xts)
```

xts Objects

xts objects have three main components:
- **coredata:** always a matrix for xts objects, while it could also be a vector for zoo objects
- **index:** vector of any `Date`, `POSIXct`, `chron`, `yearmon`, `yearqtr`, or `DateTime` classes
- **xtsAttributes:** arbitrary attributes

> Creating xts Objects

```
> xts1 <- xts(x=1:10, order.by=Sys.Date()-1:10)
> data <- rnorm(5)
> dates <- seq(as.Date("2017-05-01"),length=5,by="days")
> xts2 <- xts(x=data, order.by=dates)
> xts3 <- xts(x=rnorm(10),
              order.by=as.POSIXct(Sys.Date()+1:10),
              born=as.POSIXct("1899-05-08"))
> xts4 <- xts(x=1:10, order.by=Sys.Date()+1:10)
```

Convert To And From xts

```
> data(AirPassengers)
> xts5 <- as.xts(AirPassengers)
```

Import From Files

```
> dat <- read.csv(tmp_file)
> xts(dat, order.by=as.Date(rownames(dat),"%m/%d/%Y"))
> dat_zoo <- read.zoo(tmp_file,
                      index.column=0,
                      sep=",", format="%m/%d/%Y")
> dat_zoo <- read.zoo(tmp,sep=",",FUN=as.yearmon)
> dat_xts <- as.xts(dat_zoo)
```

> Export xts Objects

```
> data_xts <- as.xts(matrix)
> tmp <- tempfile()
> write.zoo(data_xts,sep=",",file=tmp)
```

> Replace & Update

```
> xts2[dates] <- 0 #Replace values in xts2 on dates with 0
> xts5["1961"] <- NA #Replace dates from 1961 with NA
> xts2["2016-05-02"] <- NA #Replace the value at 1 specific index with NA
```

> Applying Functions

```
> ep1 <- endpoints(xts4,on="weeks",k=2) #Take index values by time
  [1] 0 5 10
> ep2 <- endpoints(xts5,on="years")
 [1] 0 12 24 36 48 60 72 84 96 108 120 132 144
> period.apply(xts5,INDEX=ep2,FUN=mean) #Calculate the yearly mean
> xts5_yearly <- split(xts5,f="years") #Split xts5 by year
> lapply(xts5_yearly,FUN=mean) #Create a list of yearly means
> do.call(rbind, #Find the last observation in each year in xts5
          lapply(split(xts5,"years"),
          function(x) last(x,m="1 month")))
> do.call(rbind, #Calculate cumulative annual passengers
          lapply(split(xts5,"years"),
          cumsum))
> rollapply(xts5, 3, sd) #Apply sd to rolling margins of xts5
```

> Selecting, Subsetting & Indexing

Select

```
> mar55 <- xts5["1955-03"] #Get value for March 1955
```

Subset

```
> xts5_1954 <- xts5["1954"] #Get all data from 1954
> xts5_janmarch <- xts5["1954/1954-03"] #Extract data from Jan to March '54
> xts5_janmarch <- xts5["/1954-03"] #Get all data until March '54
> xts4[ep1] #Subset xts4 using ep2
```

first() and last()

```
> first(xts4,'1 week') #Extract first 1 week
> first(last(xts4,'1 week'),'3 days') #Get first 3 days of the last week of data
```

Indexing

```
> xts2[index(xts3)] #Extract rows with the index of xts3
> days <- c("2017-05-03","2017-05-23")
> xts3[days] #Extract rows using the vector days
> xts2[as.POSIXct(days,tz="UTC")] #Extract rows using days as POSIXct
> index <- which(.indexwday(xts1)==0|.indexwday(xts1)==6) #Index of weekend days
> xts1[index] #Extract weekend days of xts1
```

> Missing Values

```
> na.omit(xts5) #Omit NA values in xts5
> xts_last <- na.locf(xts2) #Fill missing values in xts2 using last observation
> xts_last <- na.locf(xts2, #Fill missing values in xts2 using next observation
                      fromLast=TRUE)
> na.approx(xts2) #Interpolate NAs using linear approximation
```

> Arithmetic Operations

coredata()or as.numeric()

```
> xts3 + as.numeric(xts2) #Addition
> xts3 * as.numeric(xts4) #Multiplication
> coredata(xts4) - xts3 #Subtraction
> coredata(xts4) / xts3 #Division
```

Shifting Index Values

```
> xts5 - lag(xts5) #Period-over-period differences
> diff(xts5,lag=12,differences=1) #Logged differences
```

Reindexing

```
> xts1 + merge(xts2,index(xts1),fill=0) #Addition
             e1
 2017-05-04 5.231538
 2017-05-05 5.829257
 2017-05-06 4.000000
 2017-05-07 3.000000
 2017-05-08 2.000000
 2017-05-09 1.000000
> xts1 - merge(xts2,index(xts1),fill=na.locf) #Subtraction
             e1
 2017-05-04 5.231538
 2017-05-05 5.829257
 2017-05-06 4.829257
 2017-05-07 3.829257
 2017-05-08 2.829257
 2017-05-09 1.829257
```

Reindexing

```
> merge(xts2,xts1,join='inner') #Inner join of xts2 and xts1
           xts2       xts1
 2017-05-05 -0.8382068  10
> merge(xts2,xts1,join='left',fill=0) #Left join of xts2 and xts1, fill empty spots with 0
           xts2       xts1
 2017-05-01 1.7482704   0
 2017-05-02 -0.2314478  0
 2017-05-03 0.1685517   0
 2017-05-04 1.1685649   0
 2017-05-05 -0.8382068  10
> rbind(xts1, xts4) #Combine xts1 and xts4 by rows
```

> Inspect your data

```
> core_data <- coredata(xts2) #Extract core data of objects
> index(xts1) #Extract index of objects
```

Class Attributes

```
> indexClass(xts2) #Get index class
> indexClass(convertIndex(xts,'POSIXct')) #Replacing index class
> indexTZ(xts5) #Get index class
> indexFormat(xts5) <- "%Y-%m-%d" #Change format of time display
```

Time Zones

```
> tzone(xts1) <- "Asia/Hong_Kong" #Change the time zone
> tzone(xts1) #Extract the current time zone
```

Periods, Periodicity and Timestamps

```
> periodicity(xts5) #Estimate frequency of observations
> to.yearly(xts5) #Convert xts5 to yearly OHLC
> to.monthly(xts3) #Convert xts3 to monthly OHLC
> to.quarterly(xts5) #Convert xts5 to quarterly OHLC
> to.period(xts5,period="quarters") #Convert to quarterly OHLC
> to.period(xts5,period="years") #Convert to yearly OHLC
> nmonths(xts5) #Count the months in xts5
> nquarters(xts5) #Count the quarters in xts5
> nyears(xts5) #Count the years in xts5
> make.index.unique(xts3,eps=1e-4) #Make index unique
> make.index.unique(xts3,drop=TRUE) #Remove duplicate times
> align.time(xts3,n=3600) #Round index time to the next n seconds
```

Other Useful Functions

```
> .index(xts4) #Extract raw numeric index of xts1
> .indexwday(xts3) #Value of week(day), starting on Sunday,in index of xts3
> .indexhour(xts3) #Value of hour in index of xts3
> start(xts3) #Extract first observation of xts3
> end(xts4) #Extract last observation of xts4
> str(xts3) #Display structure of xts3
> time(xts1) #Extract raw numeric index of xts1
> head(xts2) #First part of xts2
> tail(xts2) #Last part of xts2
```

Learn Data Skills Online at www.DataCamp.com

安装软件包

> install.packages("lmtest")

```
> library(lmtest)
载入需要的程辑包：zoo

载入程辑包：'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

>
```

Figure: lmtest: A collection of tests, data sets, and examples for diagnostic checking in linear regression models. Furthermore, some generic tools for inference in parametric models are provided.

加载软件包

> library(lmtest)

```
> library(lmtest)
载入需要的程辑包：zoo

载入程辑包：'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric

>
```

R首先可以用来当成计算器-1

- > 102.5*21
  [1]2152.5
  [1]表示命令的第一个输出结果。许多命令有多个输出结果。
- > 290/123
  [1]2.357724
- > 35+3*(26.5-21)
  [1] 51.5
- > 5^ 2
  [1]25
- > exp(3)
  [1]20.08554

R首先可以用来当成计算器-2

- 除法运算的余数，
  $> 25\,\%\%\,3$
  [1]1
- 除法运算的整数部分，
  $> 25\,\%/\%\,3$
  [1]8
- $> 3 * 8 + 1$
  [1]25

R基本数据类型

R has five basic or "atomic" classes of data type:

- Numeric(real numbers，double)
- Integer
- Complex
- Logical(True/False)
- Character(string)

```
> a <- 1
> typeof(a)
[1] "double"
> a <- 1L
> typeof(a)
[1] "integer"
```

R的基本数据结构

- vector
- list
- matrix
- array
- factor
- data frame

R的命名与值

- $> a <$-1.05 ^ 20
  $> a$
  [1]2.653298
- R使用<-对变量进行赋值运算，当然也可以直接使用"="，但是功能不是完全等价。
- 等号(=)的作用有两个：赋值，传递函数参数。一般情况下等号(=)有赋值功能；如果变量写在函数的参数位置，它就是参数，属于局部变量。

R的命名与值

- 等号(=)的作用有两个：赋值，传递函数参数。一般情况下等号（=）有赋值功能；如果写在函数的参数位置，它就是参数，属于局部变量。
- 用"="传递参数例子

```
> rm(a)
> a<-1:5
> sum(a)
[1] 15
> sum(a=1:6)
[1] 21
> a
[1] 1 2 3 4 5
```

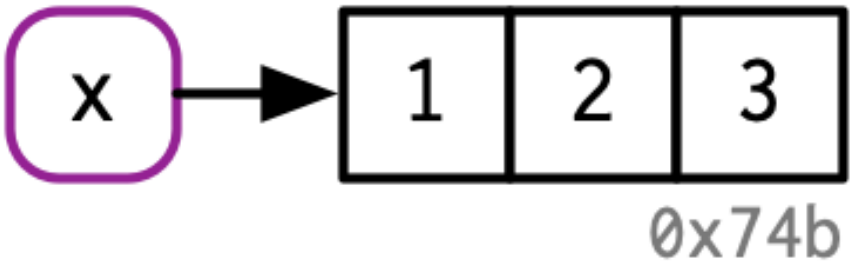
R的命名与值

- 在传递参数时候使用"<-" 的例子，使用" <-"实际上是全局变量的赋值运算了。

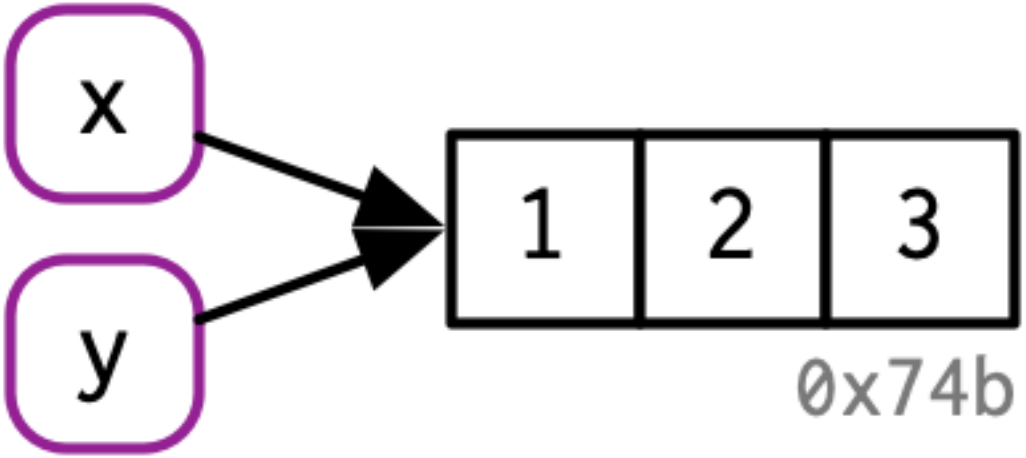
```
> rm(a)
> a<-1:5
> sum(a)
[1] 15
> sum(a<-1:6)
[1] 21
> a
[1] 1 2 3 4 5 6
```

- 结果显而易见。那么，我们认为在函数传递参数的时候需要对参数变量赋值的，要用"="，一般赋值运算用" < −"和"="均可。

x <- c(1, 2, 3)
创建一个对象，一个有值1，2，3的向量。
并把这个对象绑定一个名字, x。名字可以理解成值的引用（reference）。



y <- x



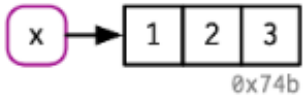
名字与对象

library(lobstr)

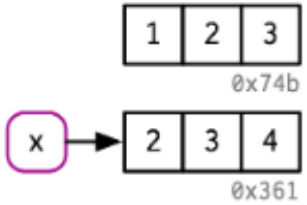
```
> x <- c(1,2,3)
> lobstr::obj_addr(x)
[1] "0x186056e6688"
> y <- x
> lobstr::obj_addr(y)
[1] "0x186056e6688"
>
```

名字与对象

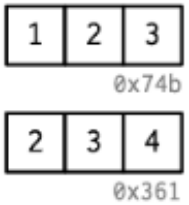
```r
x <- 1:3
```




```r
x <- 2:4
```

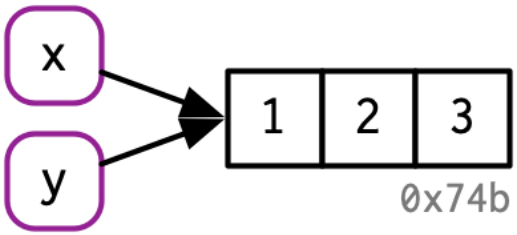


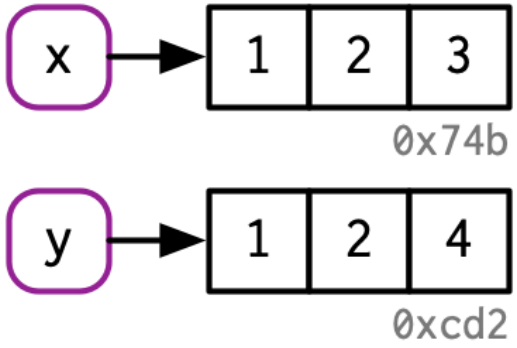

```r
rm(x)
```



Copy-on-modify

```
x <- c(1, 2, 3)
y <- x
y[[3]] <- 4
x
> [1] 1 2 3
```





练习

- 假设存款100元，年利率5%，按年计算复合收益，存款10年一共得到多少利息?

  $> interest_rate <\text{-}\ 0.05$

  $>$ p $<\text{-}$ 100

  $> interest <\text{-}\ 100 * (1 + interest_rate)$ ^ 10 - 100

  $>$ interest

  [1] 62.88946

练习

- 假设某人购买房子现在要贷款100万，贷款年限是20年，贷款月利率是$i = 1\%$，每个月等额本息还款。那么每个月还款多少钱?
- 假设贷款的现值是$P = 100$万，20年有$n = 240$个月，每个月还款额为R，根据时间价值计算方法计算公式是:

$$P = R(1+i)^{-1} + R(1+i)^{-2} + ... + R(1+i)^{-n}$$

对以上公式求和化简得到

$$P = R(\frac{1-(1+i)^{-n}}{i})$$
$$R = P(\frac{i}{1-(1+i)^{-n}})$$

练习

- 一小段代码实现计算
-
```
> i <- 0.01
> P <- 100
> n <- 240
> R <- P*(i/(1-(1+i)^(-n)))
> R
[1] 1.101086
```

序列向量

- 可以用"a:b" 表达序列向量，其中a与b不一定是整数，但是一般情况用的是整数。例如1:30表示1，2，3，4，...，30这个递增的序列向量。
- R语言对序列向量可以做各种算数运算。

R函数

- R的输入的命令，实际上是函数。函数有的有参数，有的没有参数。例如我们可以观察R环境的退出函数q()的源代码。
- > q
  function (save = "default", status = 0, runLast = TRUE)
  .Internal(quit(save, status, runLast))
  <bytecode: 0x0000000034290510>
  <environment: namespace:base>
- q 这个函数有三个参数：save, status, 和runLast。但是它们都有缺省的值。例如status的缺省值是0。

R函数

- q 这个函数有三个参数：save, status, 和runLast。但是它们都有缺省的值。例如status的缺省值是0。
- 如果我们保留前面两个参数的缺省值，修改最后一个参数的值，可以这样：

$$\mathrm{q}(\mathrm{runLast} = \mathrm{FALSE})$$

或者用逗号"，"隔开，例如

$$\mathrm{q}(\ ,\ ,\ \mathrm{FALSE})$$

R对大小写敏感

- x <- 1:10
  MEAN(x)
  ## Error in eval(expr, envir, enclos): could not find function "MEAN"
- > x <- 1:10
  > mean(x)
  [1] 5.5

罗列在workspace中的对象名称

- objects()函数，也可以用ls()函数。如果关闭R工作环境没有保存它，那么用户创建的对象在下次打开工作空间的时候将会消失。

RGui (64-bit)

文件 编辑 查看 其他 程序包 窗口 帮助

R Console

```
> mean(x)
[1] 5.5
> objects()
 [1] "a"                    "areas"
 [3] "cancer"               "data"
 [5] "fit"                  "full"
 [7] "i"                    "interest"
 [9] "interest_rate"        "iris"
[11] "ldamod"               "lrm"
[13] "main"                 "mydata"
[15] "n"                    "net.sum"
[17] "p"                    "P"
[19] "plda"                 "pred"
[21] "prediction"           "prediction_response"
[23] "predictions"          "probabilities"
[25] "qdamod"               "R"
[27] "radii"                "sfit"
[29] "SUM"                  "sum.data"
[31] "svm_model"            "svm_model_after_tune"
```

R的向量

- R表达数据向量的方法用c()函数。例如，c(0, 7, 8)表示一个数据向量，向量中的元素有3个分别是0，7，8。
- 也可以把向量赋值到一个变量（对象），x <- c(0, 7, 8)
- 如果要看x的对象内容，直接输入x，就得到

```
> x<-c(0,7,8)
> x
[1] 0 7 8
```

- 注意：向量必须用c()这个函数定义，如果没有c将是错误的，例如：

```
> x<-(0,7,8)
错误: 意外的',' in "x<-(0,"
```

R的向量

- 对于整数递增或者递减序列，可以用简单的":"表达，例如3:10
  > 3:10
  [1] 3 4 5 6 7 8 9 10
  > y <- 3:10
  > y [1] 3 4 5 6 7 8 9 10
  >

R的向量

- 也可以用c()把两个向量连起来。c表示concatenate

```
> z = c(x,y)
> z
[1] 0 7 8 3 4 5 6 7 8 9 10
>
> z = c(x,y,1,2,3,4)
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4
>
```

R的向量

- 也可以用c()把两个向量连起来。

```
> z = c(x,y,1,2,3,4,5:6)
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6
>
> z = c(x,y,1,2,3,4,5:9)
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
>
```

从向量中抽取元素

- $>$ z
  [1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
  $> z[2]$
  [1]7
  $> z[3]$
  [1]8
  $> z[2]$
  [1]7

从向量中抽取元素

- $>$ z
  [1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
  $> z[c(1,4,5)]$
  [1] 0 3 4
  $> z[2:4]$
  [1] 7 8 3

从向量中抽取元素

- $>$ z
  [1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
  $> z[-c(1,3,4)]$
  [1] 7 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
  $>$ z[-(1:4)]
  [1] 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9

从向量中抽取元素

- $>$ z
  [1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
- 注意向量的索引(index)是整数，不能出现这样的表达$z[0.5]$，这是错误的。
- 一些错误的表达：
  $> z[c(-2,3)]$
  Error in $z[c(-2,3)]$: 只有负下标里才能有零
  $> z[-2,3]$
  Error in $z[-2,3]$: 量度数目不对
  $> z[3,-3]$
  Error in $z[3,-3]$: 量度数目不对
  $> z[-3,-4]$
  Error in $z[-3,-4]$: 量度数目不对

从向量中抽取元素

- 然而，下面的表达是正确的:

```
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
> z[-2]
[1] 0 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
```

向量的数学运算

- 向量可以做加减乘除简单的数序运算。

```
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
> z * 3
[1] 0 21 24 9 12 15 18 21 24 27 30 3 6 9 12 15 18 21 24 27
```

向量的数学运算

- 向量可以做加减乘除简单的数序运算。

```
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
> y <- z - 5
> y
[1] -5 2 3 -2 -1 0 1 2 3 4 5 -4 -3 -2 -1 0 1 2 3 4
```

向量的数学运算

- 向量可以做加减乘除简单的数序运算。

```
> z
[1] 0 7 8 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9
> y <- z^2
> y
[1] 0 49 64 9 16 25 36 49 64 81 100 1 4 9 16 25
[17] 36 49 64 81
```

向量的数学运算

- 向量的幂运算。

```
> a<-c(1,2,3)
> b<-c(3,2,1)
> y <- a^b
> y
[1] 1 4 3
```

向量的数学运算

- 向量长度不对等的运算，短的向量循环运算。

```
>c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9,10, 10) %% 2:3
[1] 1 1 0 2 1 0 0 1 1 2 0 0 1 1 0 2 1 0 0 1
```

向量的数学运算

- 向量长度不对等的运算，短的向量循环运算，如果长向量的长度不是短向量的整数倍。
  > c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10) %% 2:4
  > Warning in c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10)%% 2:4: longer object length is not a multiple of shorter object length
  [1] 1 1 2 0 0 3 0 1 1 1 0 2 1 1 0 0 0 1 0 1

简单模式向量

- seq()函数可以产生有简单模式的向量。参数by如果没有指定值，那么by=1。
- > seq(1, 20, by=3)
  [1] 1 4 7 10 13 16 19

简单模式向量

- rep()函数可以产生具有重复模式的向量。
- > rep(3,12)
  [1] 3 3 3 3 3 3 3 3 3 3 3 3
  > rep(seq(2, 20, by = 2), 2)
  [1] 2 4 6 8 10 12 14 16 18 20 2 4 6 8 10 12 14 16 18 20

简单模式向量

- > rep(c(1, 4), c(3, 2))
  [1] 1 1 1 4 4
  > rep(c(1, 4), each = 3)
  [1] 1 1 1 4 4 4
  > rep(1:10, rep(2, 10))
  [1] 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10

rep的另外一些例子

```
rep(1:4, 2)
rep(1:4, each = 2) # not the same.
rep(1:4, c(2,2,2,2)) # same as second.
rep(1:4, c(2,1,2,1))
rep(1:4, each = 2, len = 4) # first 4 only.
rep(1:4, each = 2, len = 10) # 8 integers plus two recycled 1's.
rep(1:4, each = 2, times = 3) # length 24, 3 complete replications
```

随机抽样产生

- 用sample()函数。
  >sample(1:6, size = 8, replace = TRUE) # an imaginary die is tossed 8 times
  > [1] 1 5 1 5 3 1 2 1

字符向量

- >colors <- c("red", "yellow", "blue")
  >more.colors <- c(colors, "green", "magenta", "cyan")
  >z <- c("red", "green", 1)
  >more.colors
  [1] "red" "yellow" "blue" "green" "magenta" "cyan"
  z # 1 has been converted to the character "1"
  [1] "red" "green" "1

字符向量

- 要获取子字符串，请使用substr()。有几个参数，substr(x，start，stop)，其中x是字符串的向量，并开始和停止说出要保留的字符。例如，要打印每种颜色使用的前两个字母

```
> substr(colors, 1, 2)
[1] "re" "ye" "bl"
```

字符向量

- 另一个基本操作是通过串联构建字符串。使用这个的paste()函数。缺省连接字符是空格，例如，
  >paste(colors, "flowers")
  [1] "red flowers" "yellow flowers" "blue flowers"
  > paste("several", colors, "s")
  [1] "several red s" "several yellow s" "several blue s"
  >
- 去掉缺省的空格连接符
  > paste("several ", colors, "s", sep = "")
  [1] "several reds" "several yellows" "several blues"
  可以看到，后面的s与colors中的元素连在了一起。

字符向量

- paste0()函数完成上面的功能。
  >paste0("several ", colors, "s")
  [1] "several reds" "several yellows" "several blues"

字符向量

- paste()的collapse参数允许所有组件生成的向量将折叠为单个字符串：

```
> paste("I like", colors, collapse = ", ")
[1] "I like red, I like yellow, I like blue"
```

Factors

- Factor提供了另一种存储“类别”数据的方法。例如，一个有四个元素并且有两个level, "control"和"treatment"可以使用：

```
> grp <- c("control", "treatment", "control", "treatment")
> grp
[1] "control" "treatment" "control" "treatment"
> grp <- factor(grp)
> grp
[1] control treatment control treatment
Levels: control treatment
```

Factors

- 当在向量中的元素有很多重复的时候，Factor是存储字符数据的有效方式。这是因为一个factor的level在内部编码为整数。例如查看一个factor 对象的内部整数编码：

```
> as.integer(grp)
[1] 1 2 1 2
```

Factors

- Level 的标签只存储一次，而不是重复存储。代码是level向量的索引：

  $> levels(grp)$

  [1] "control" "treatment"

  > levels(grp)[as.integer(grp)]

  [1] "control" "treatment" "control" "treatment"

Factors

- levels()函数可用于将factor标签更改。例如，假设我们希望将“control”标签更改为“placebo”。由于“control”是第一个level，我们改变第一个元素级别(grp)向量：levels(grp)[1] <- "placebo"
- Factor的一个重要用途是列出所有可能的值，即使有些值没有。例如，

```
> sex <- factor(c("F", "F"), levels = c("F", "M"))
> sex
[1] F F
Levels: F M
```

Factor的元素访问

- 方括号[]用于索引因子和向量元素例如，因子grp有四个元素，所以我们可以通过输入打印出第三个元素

```
>grp[3]
[1] placebo
Levels: placebo treatment
```

List

列表是一种特殊类型的向量，它可以包含不同类型的元素。

```
> x <- list(1, "a", TRUE, 1 + 4i)
> x
[[1]]
[1] 1

[[2]]
[1] "a"

[[3]]
[1] TRUE

[[4]]
[1] 1+4i
```

矩阵与数组

- matrix()

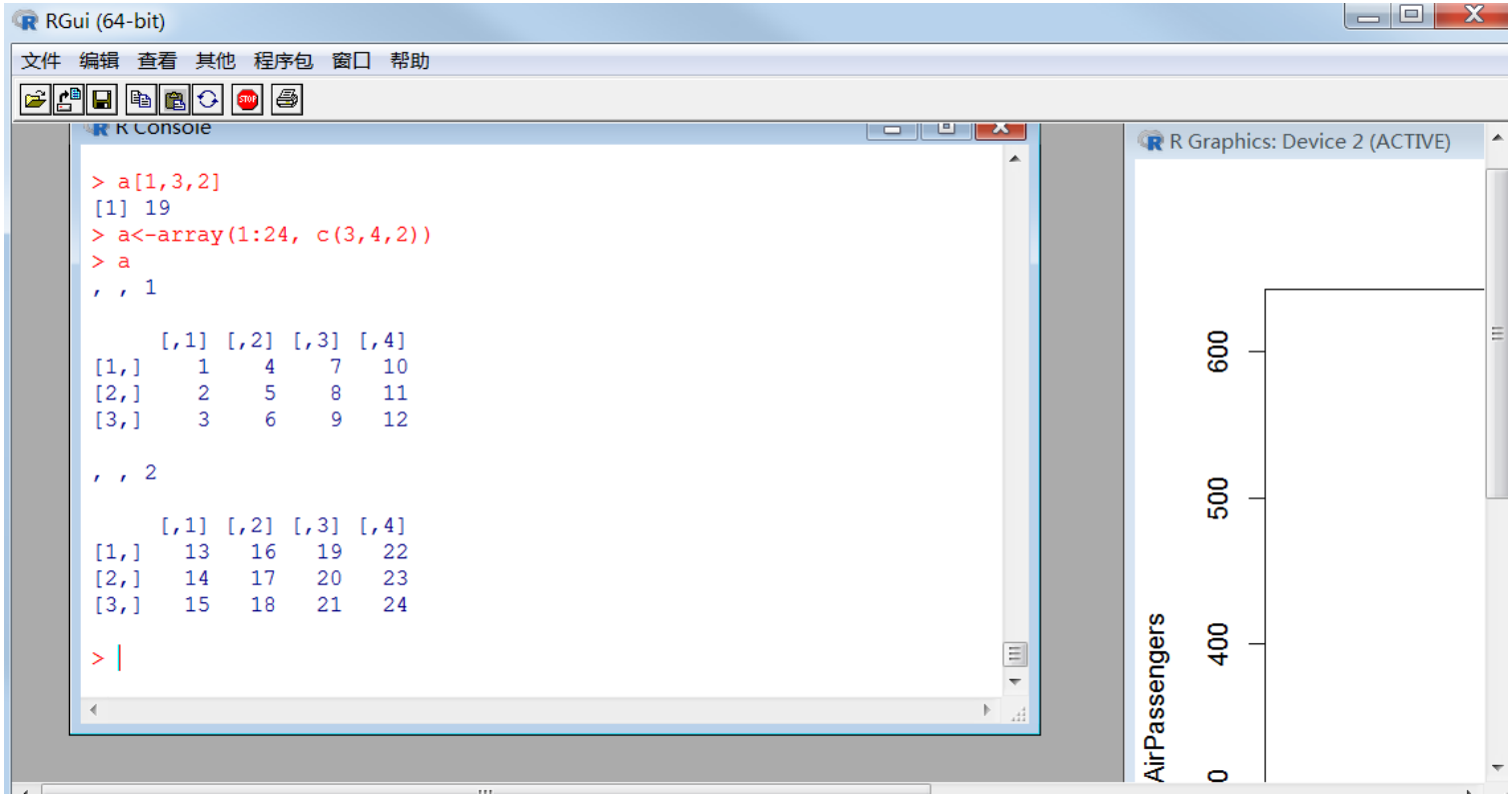
```
> m <- matrix(1:6, nrow = 2, ncol = 3)
> m
[,1][,2][,3]
[1,] 1 3 5
[2,] 2 4 6
```

矩阵的访问

- > m[2,3]
  [1] 6
- 访问一行
  $> m[1,]$
  [1] 1 3 5
- 访问一列
  > m[,2]
  [1] 3 4

数组

- 数组可拥有多维度索引，更为一般的保存数据的数据结构。
- array()



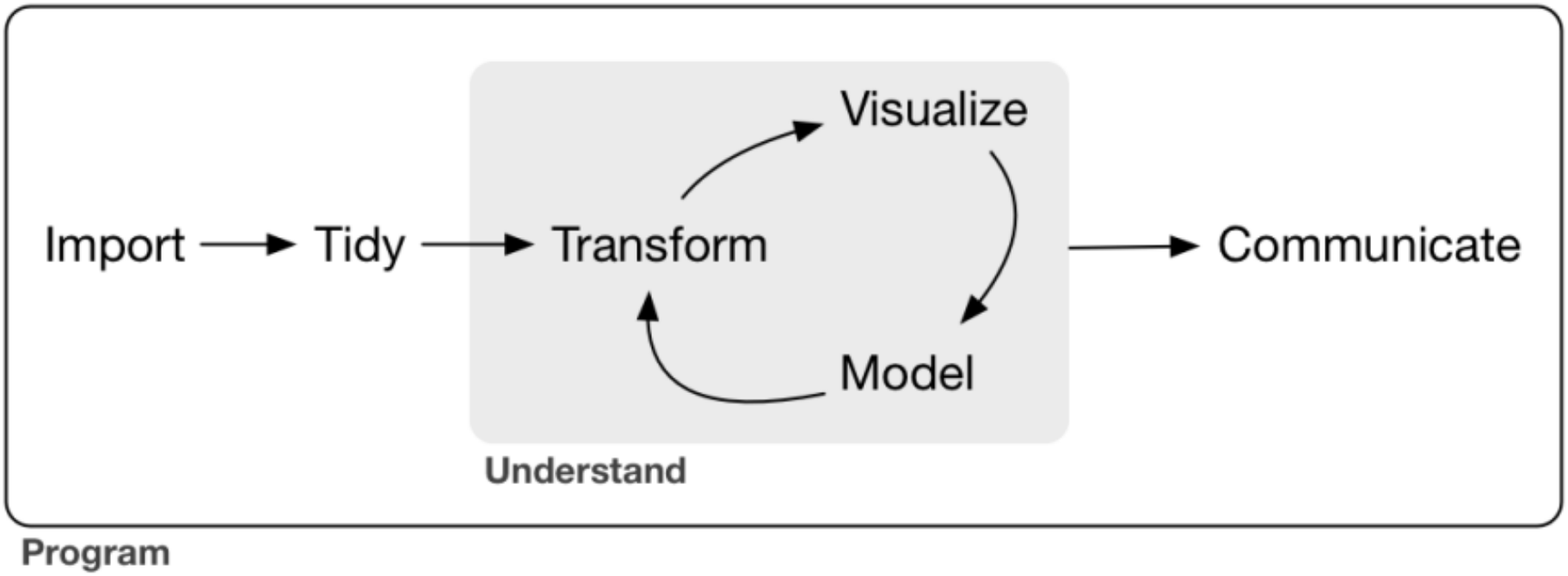
数组元素的提取

- $> a[1,3,2]$
  [1] 19

outline

- 课程介绍
- R基本语法
- **使用ggplot2可视化数据**

data science project



需要安装的软件包

- install.packages("tidyverse")
- library(tidyverse)
- package::function()

mpg datasett

```
> library(tidyverse)
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
# ℹ 224 more rows
# ℹ Use `print(n = ...)` to see more rows
>
```

分析数据的目标是回答问题：What does the relationship between engine size and fuel efficiency look like? Is it positive? Negative? Linear? Nonlinear?

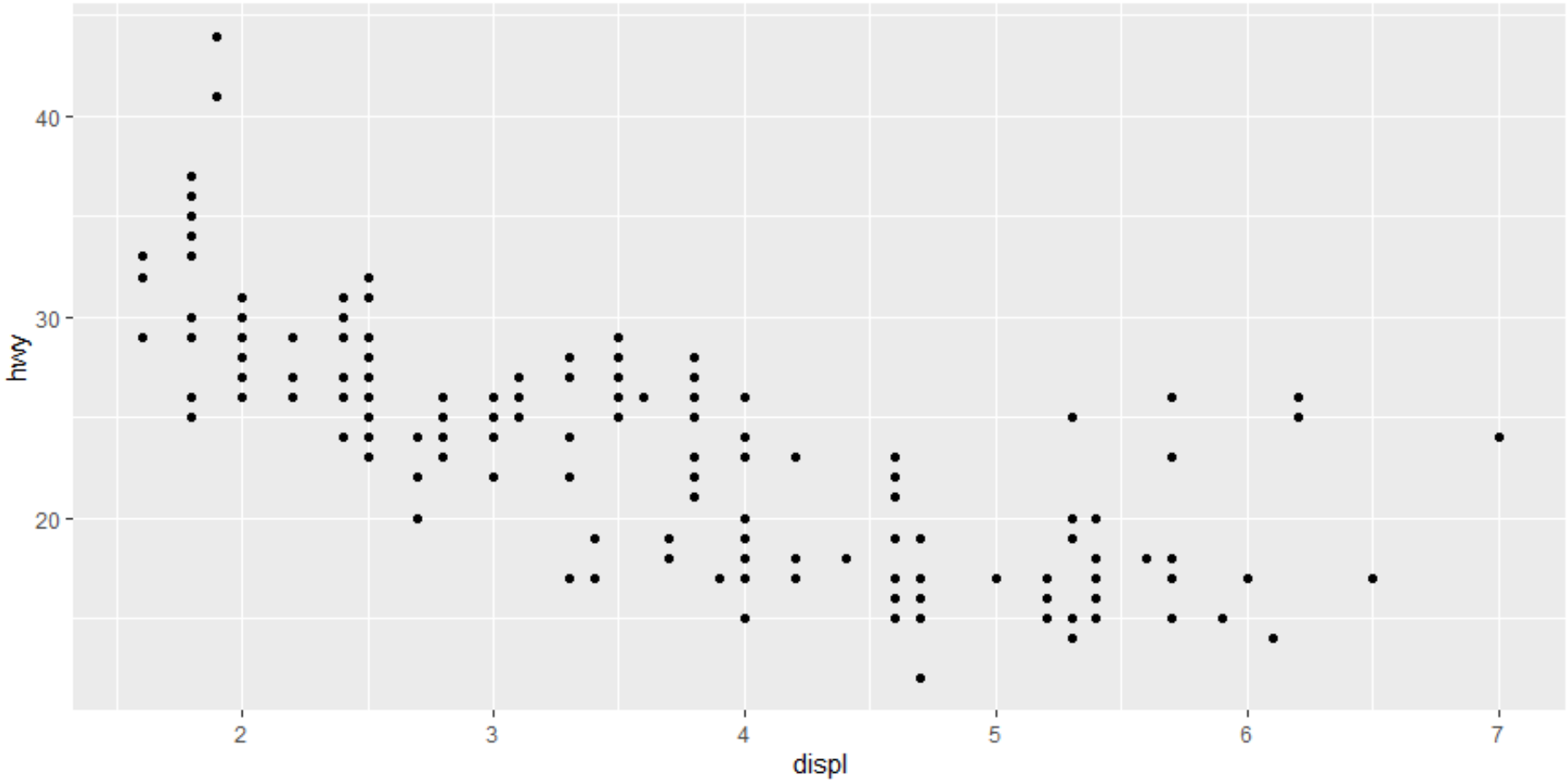
mpg datasett

- displ, a car's engine size, in liters.
- hwy, a car's fuel efficiency on the highway, in miles per gallon (mpg).

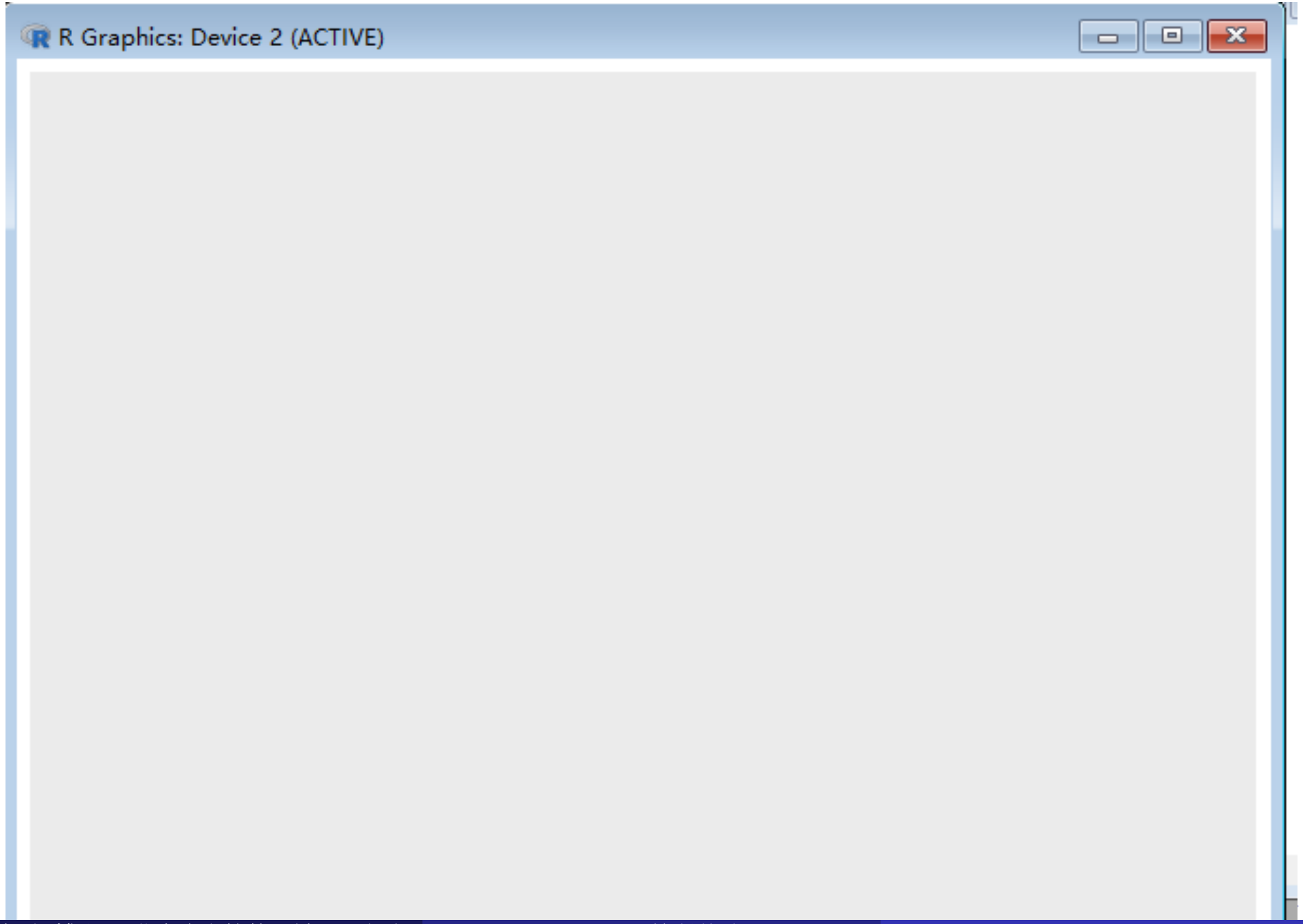
```
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
 2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
 3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
 4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
 5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
 6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
 7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
 8 audi         a4 quattr    1.8  1999     4 manu… 4        18    26 p     comp…
 9 audi         a4 quattr    1.8  1999     4 auto… 4        16    25 p     comp…
10 audi         a4 quattr    2    2008     4 manu… 4        20    28 p     comp…
# ℹ 224 more rows
# ℹ Use `print(n = ...)` to see more rows
```

绘图

ggplot(data = mpg) +geom_point(mapping = aes(x = displ, y = hwy))



如果只是执行语句ggplot(data = mpg)，会产生什么结果?



绘图模板

To make a graph, replace the bracketed sections in the following code with a dataset, a geom function, or a collection of mappings:

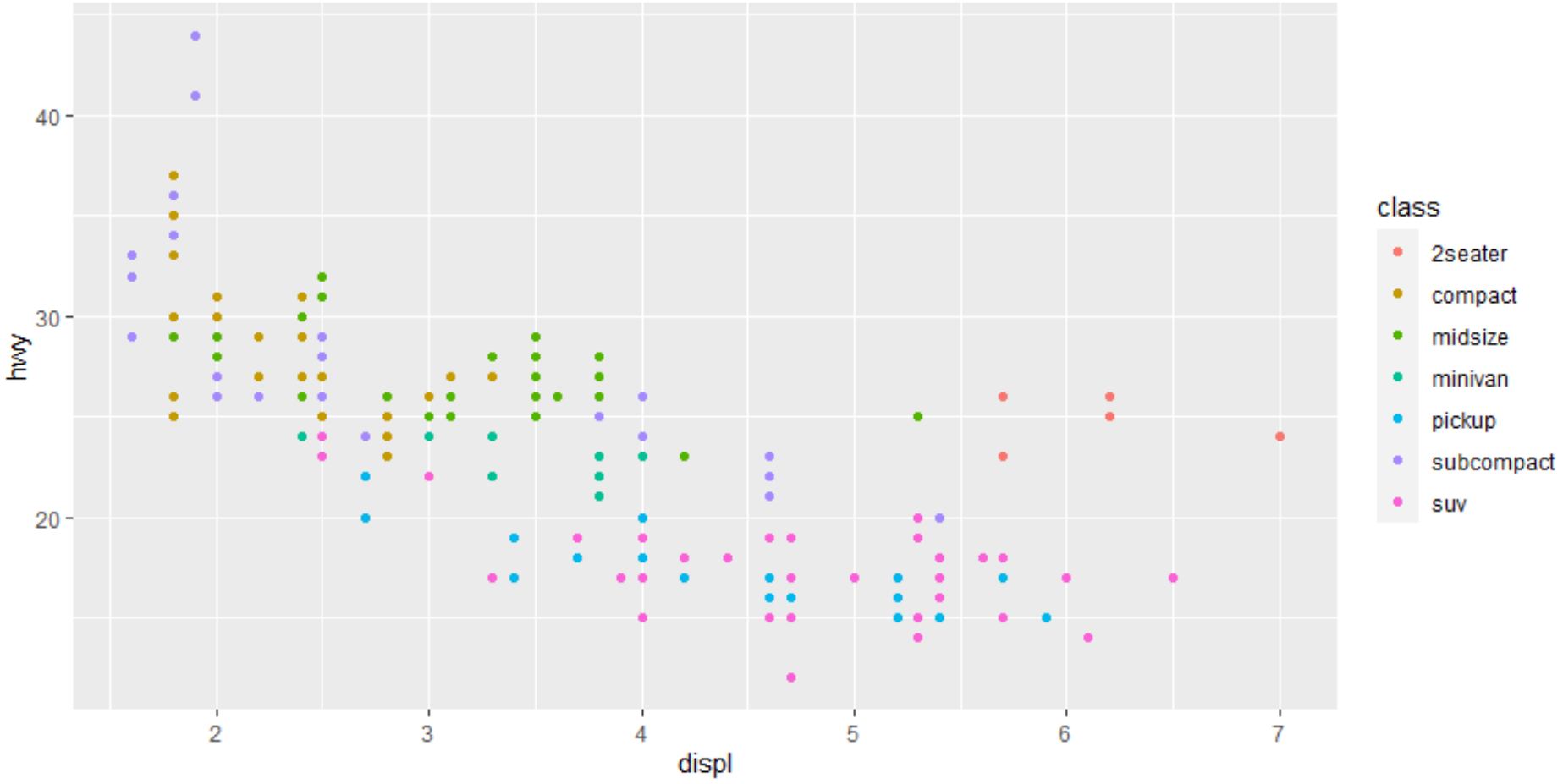
$$\text{ggplot(data} = \text{<DATA>)} + \text{<GEOM_FUNCTION>(mapping} = \text{aes(<MAPPINGS>))}$$

增加变量

- 前面绘的图其实只表达了两个维度的信息。一个是发动机容量displ，另外一个是燃油燃油效率hwy。现在我们再增加一个变量就是class。观察汽车的类型。这个时候我们增加aesthetic的映射内容。
- An aesthetic is a visual property of the objects in your plot. Aesthetics include things like the size, the shape, or the color of your points.
- For each aesthetic you use, the aes() to associate the name of the aesthetic with a variable to display

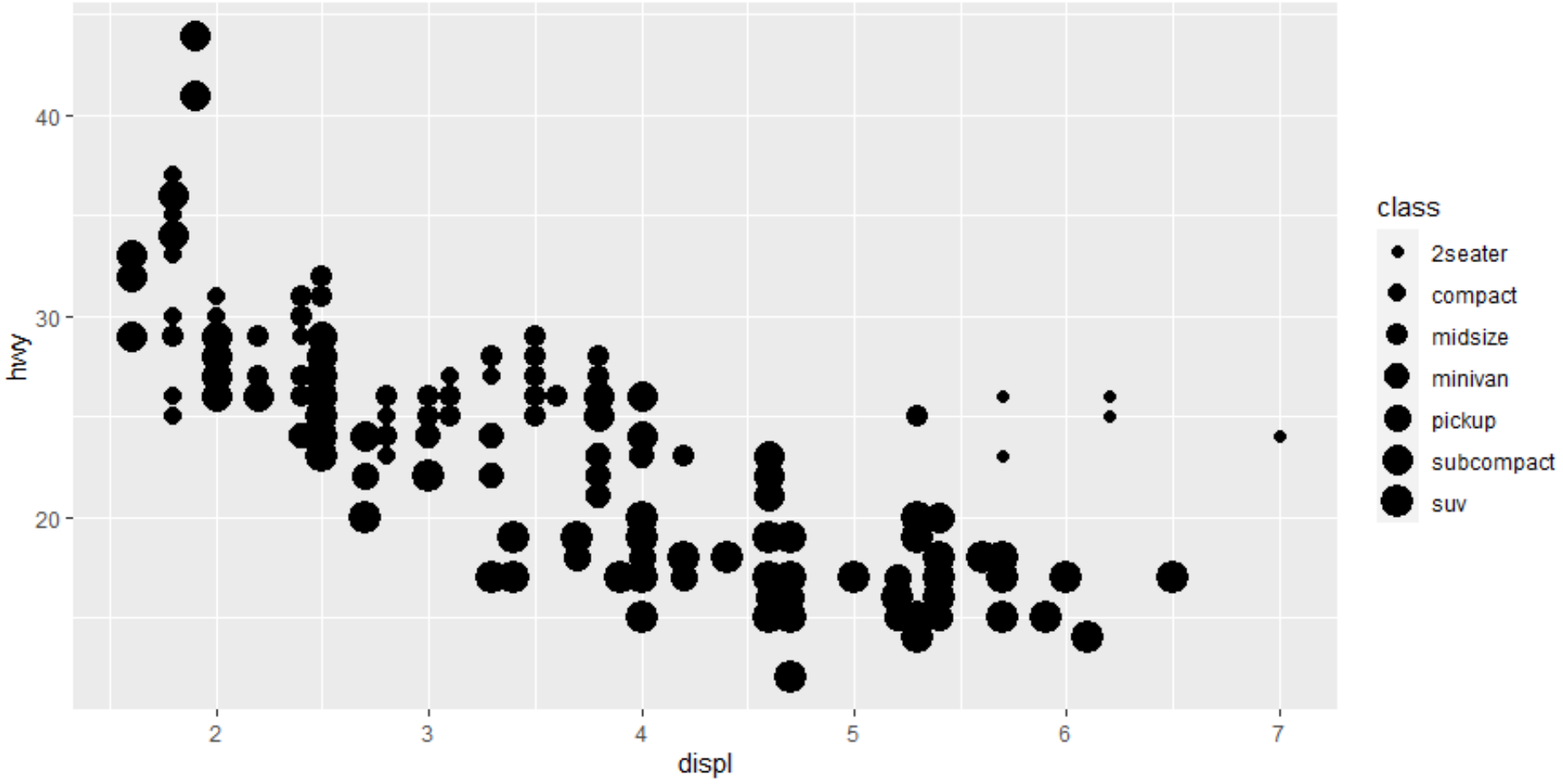
用不同的颜色表示不同类型的汽车

在aes中，关联aesthetic和变量。ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = class))



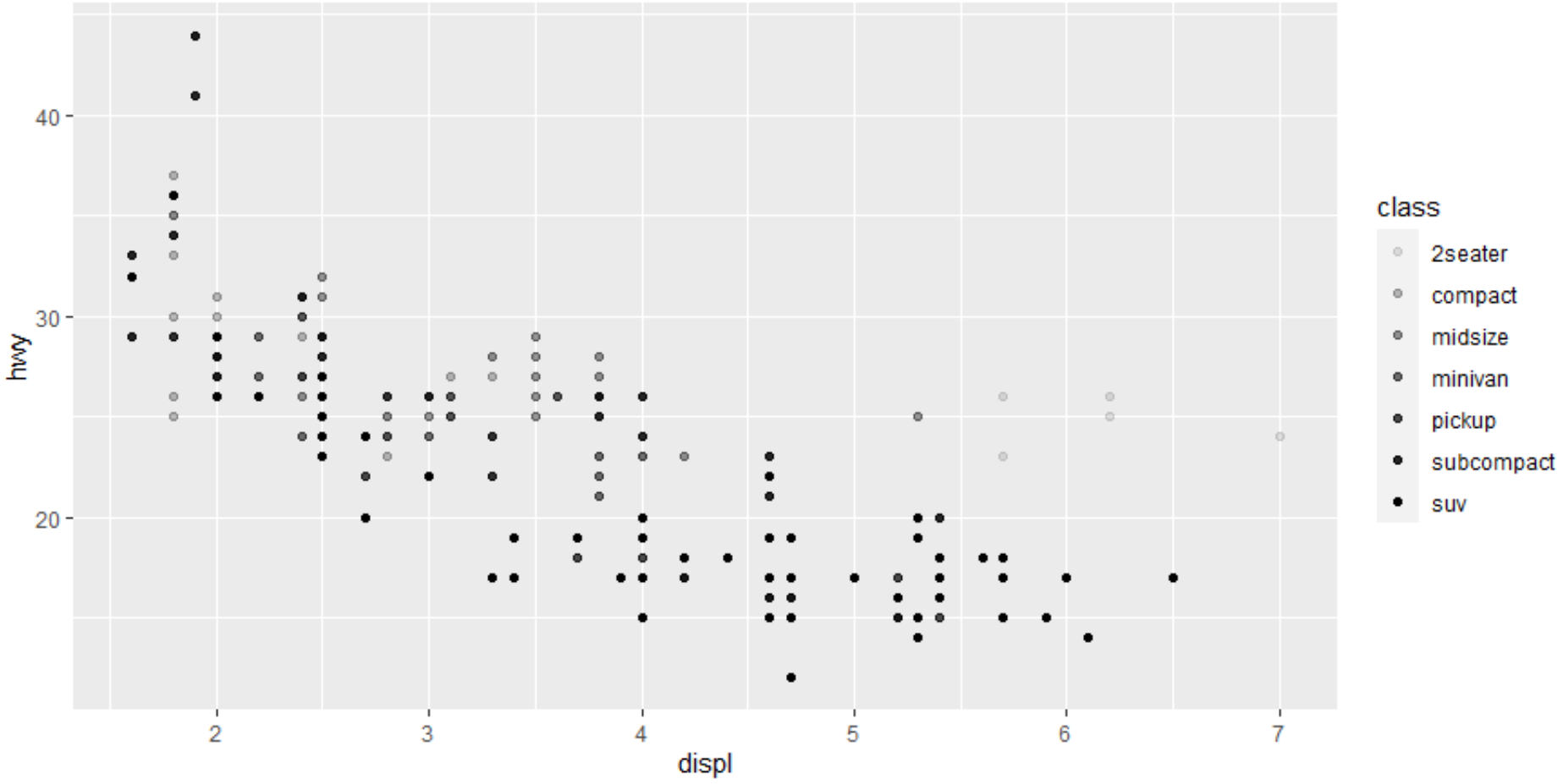
用不同的大小表示不同类型的汽车

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, size = class))



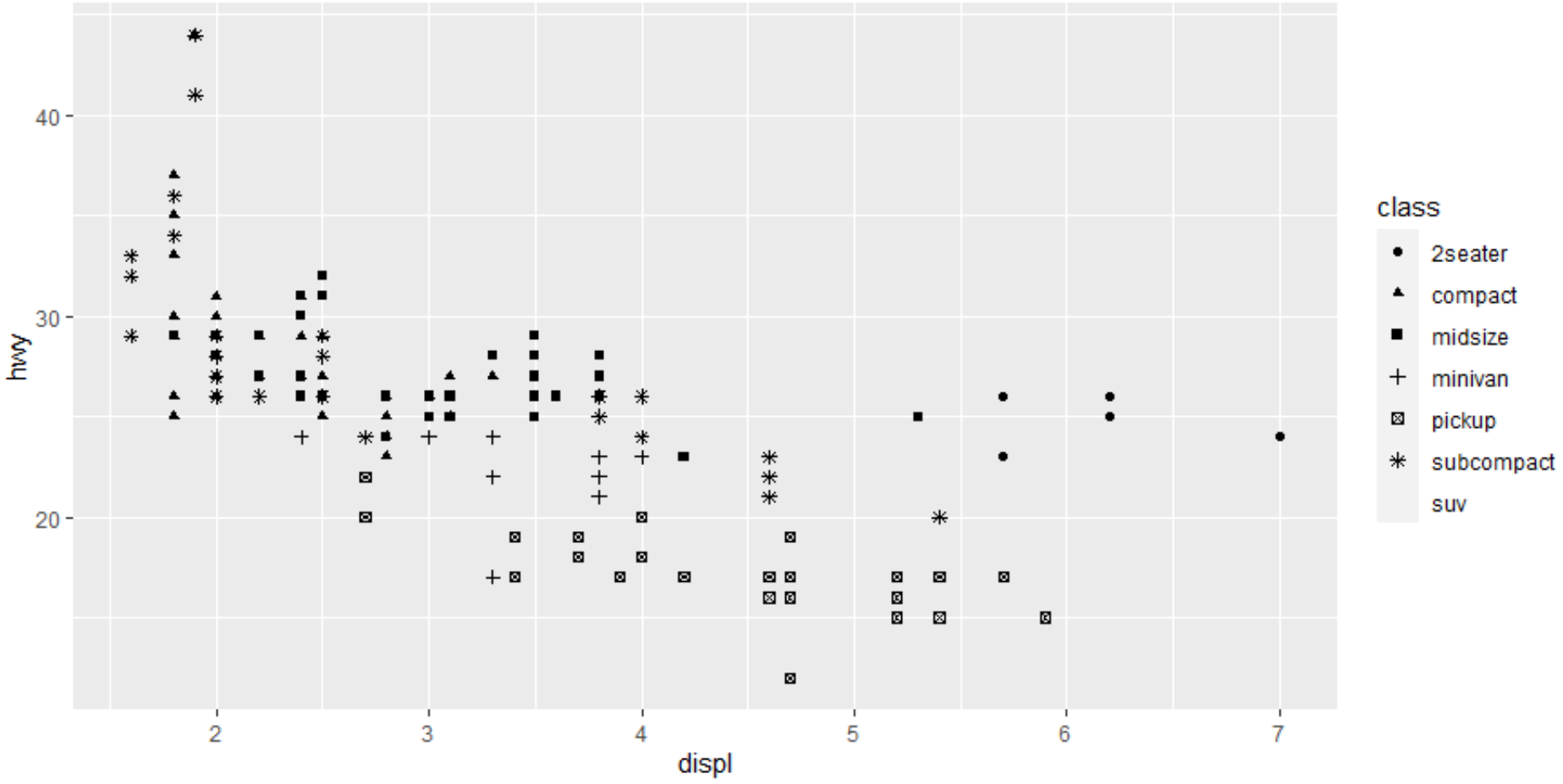
透明度

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, alpha = class))



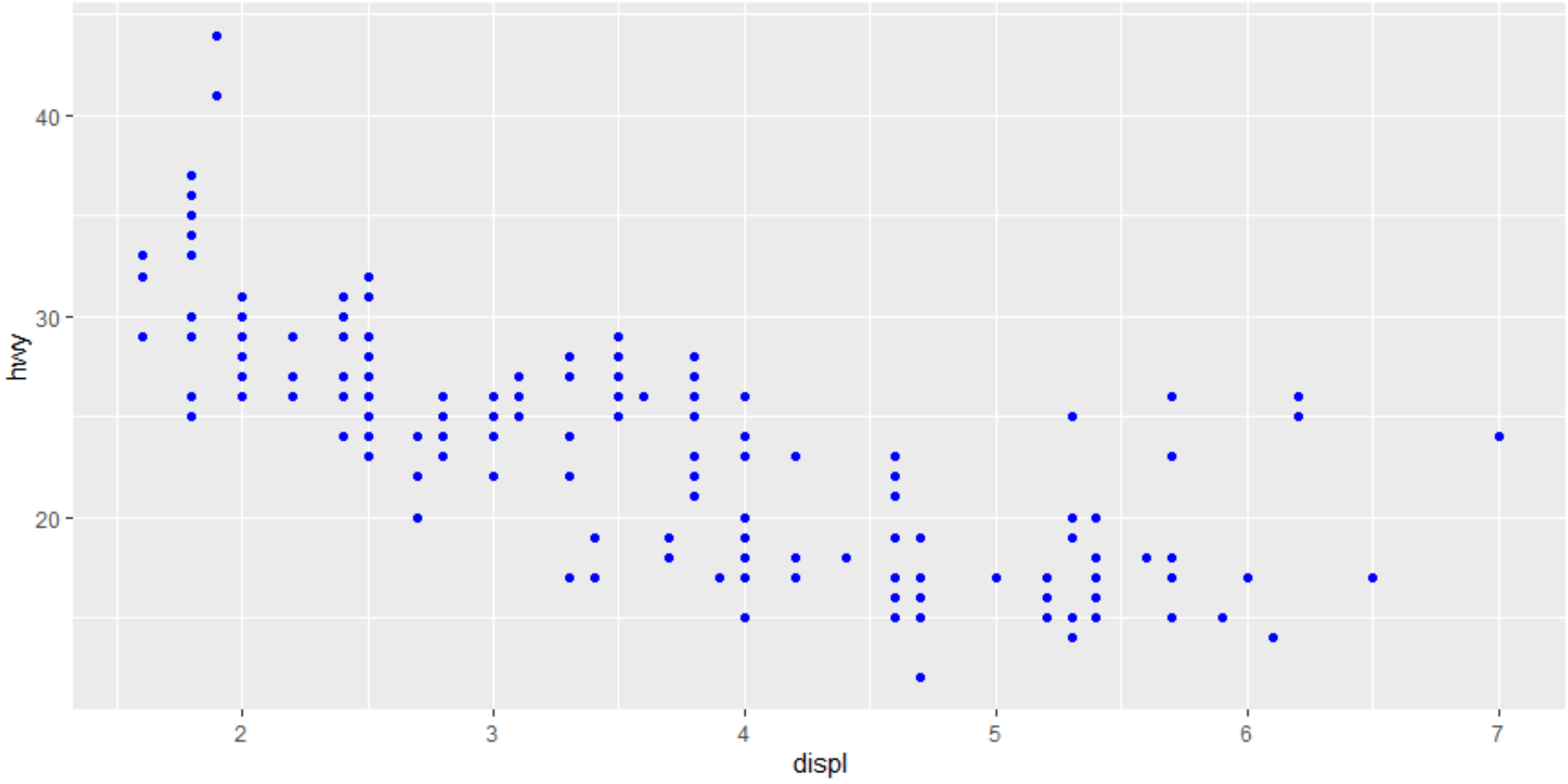
形状

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, shape = class))



设置plot属性

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy), color = "blue")，注意color的位置。

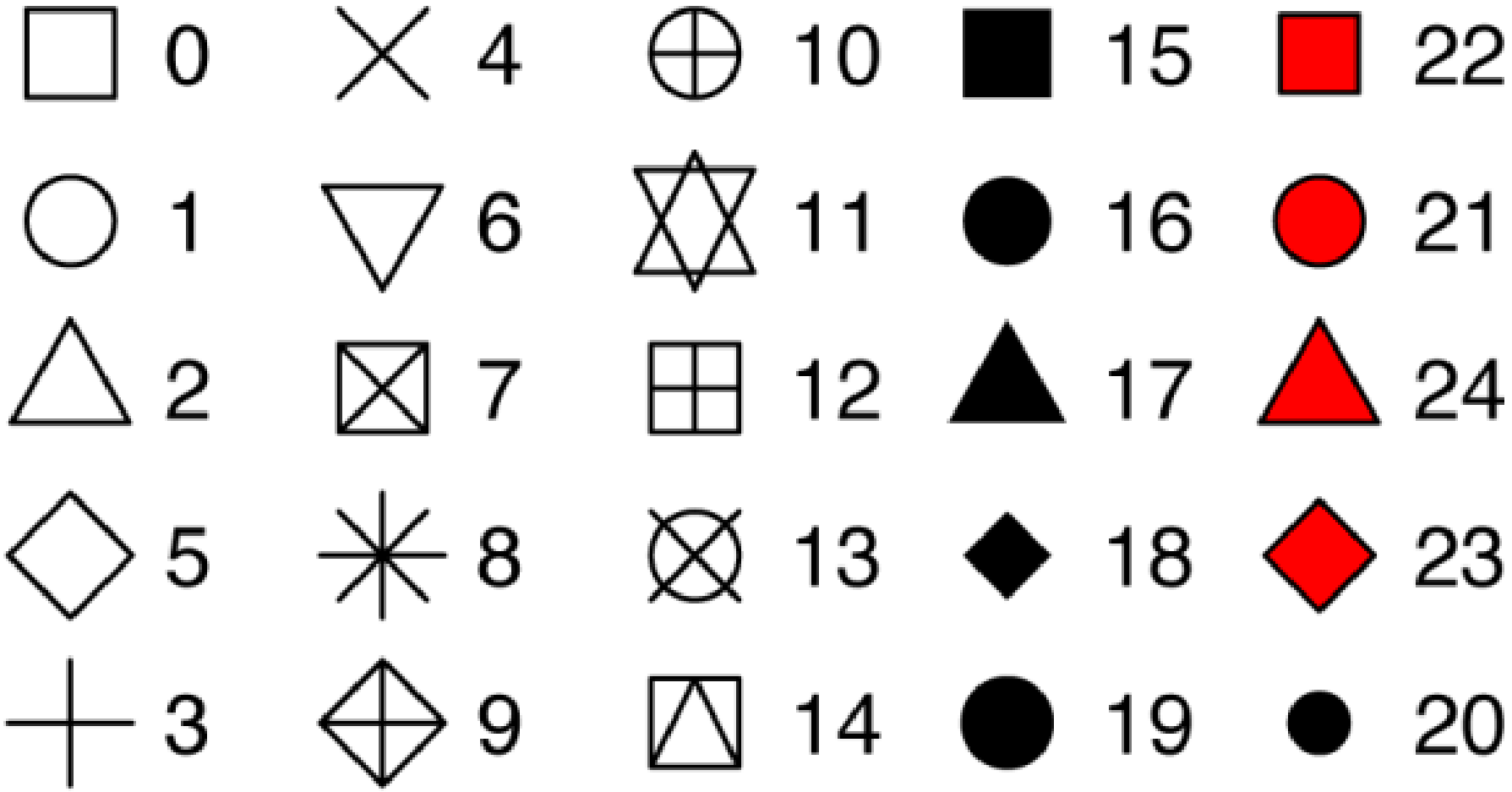


设置plot属性

改变绘图中的属性包括:

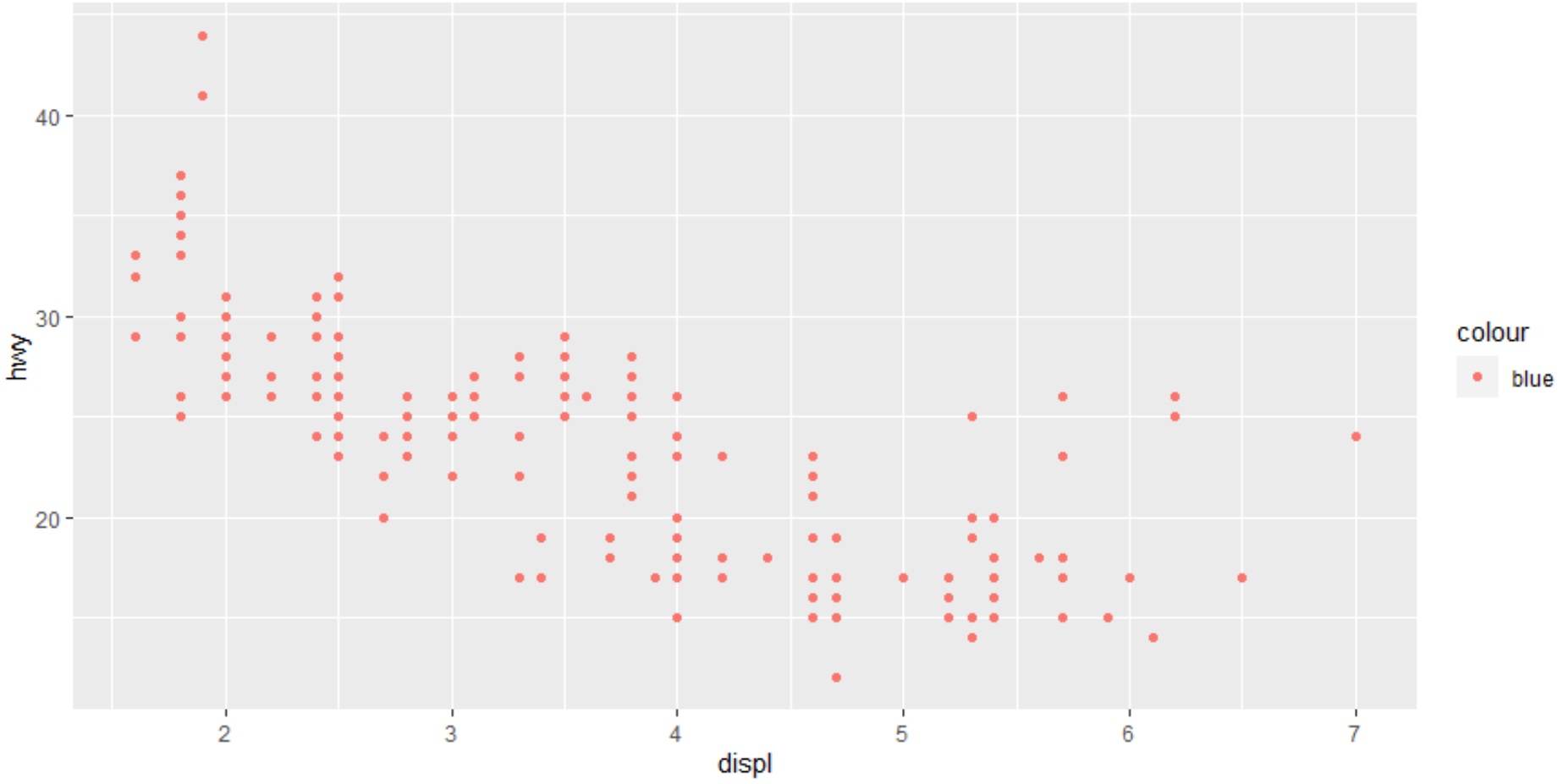
- The name of a color as a character string.
- The size of a point in mm.
- The shape of a point as a number

The shape of a point as a number



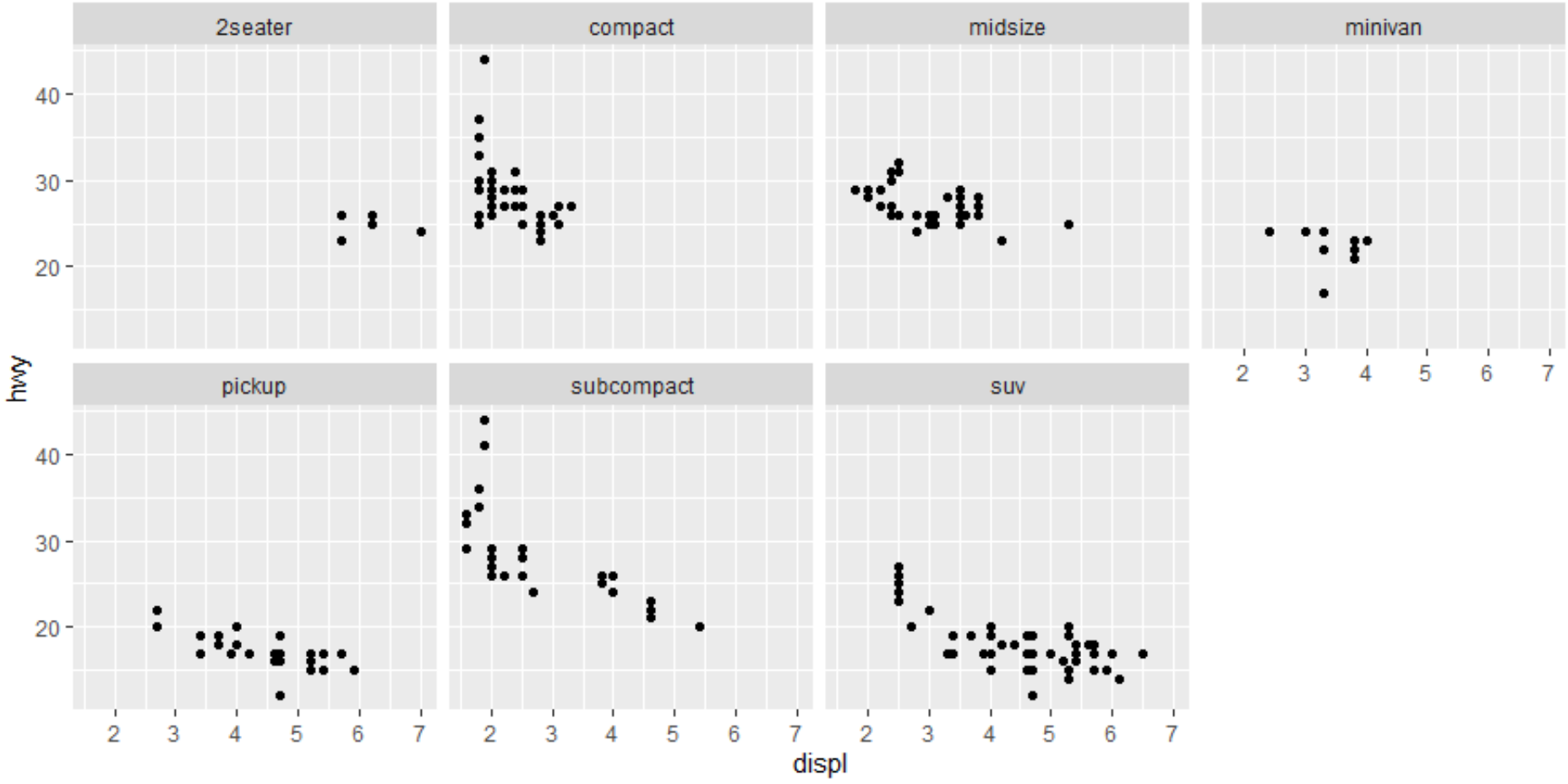
下面语句输出

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy, color = "blue"))



Facets

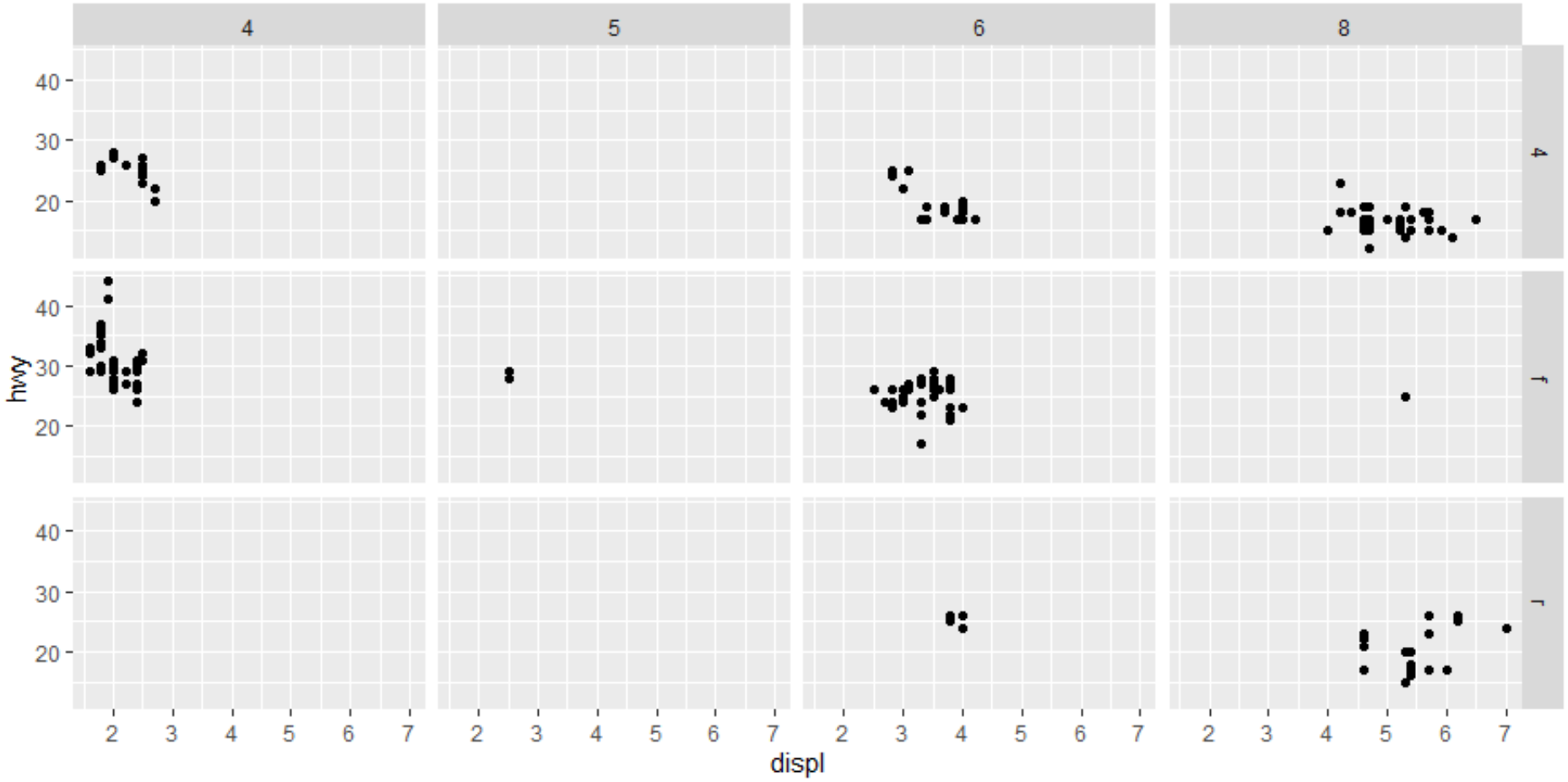
ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + facet_wrap(~class, nrow = 2)



Facets

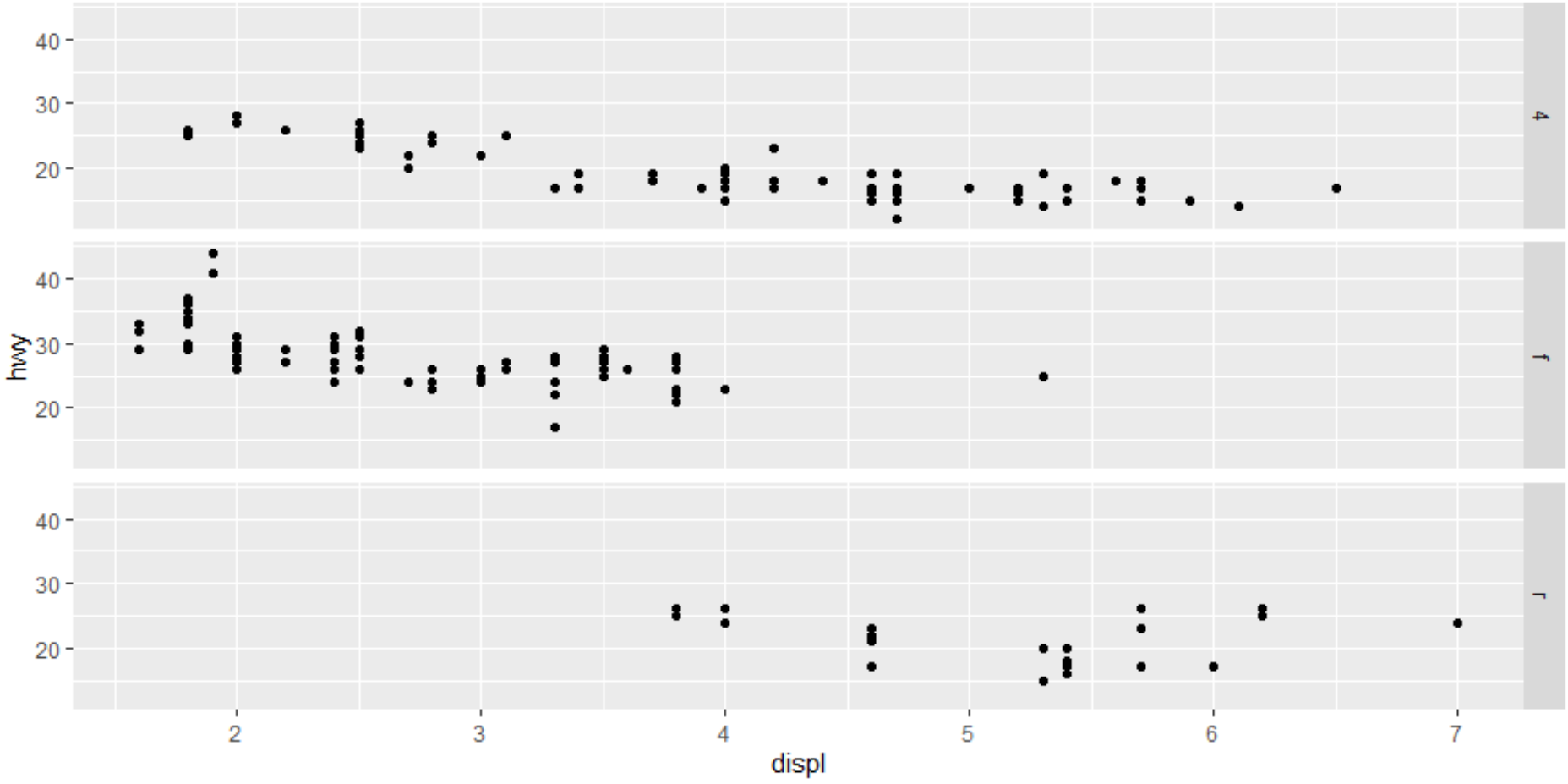
两个变量时候用facet_grid()

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + facet_grid(drv ~cyl)



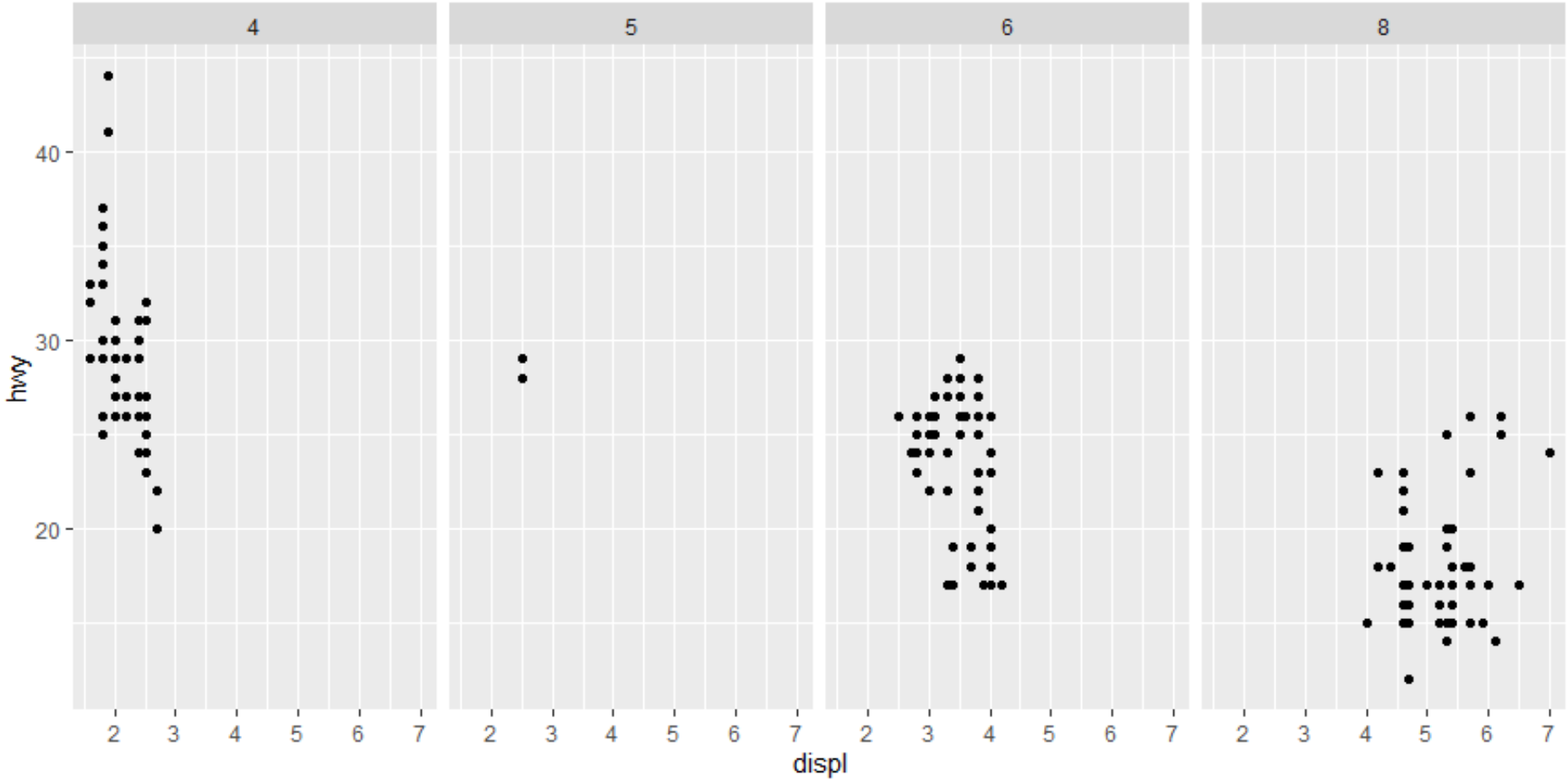
Facets

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + facet_grid(drv~.)



Facets

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + facet_grid(. ~cyl)



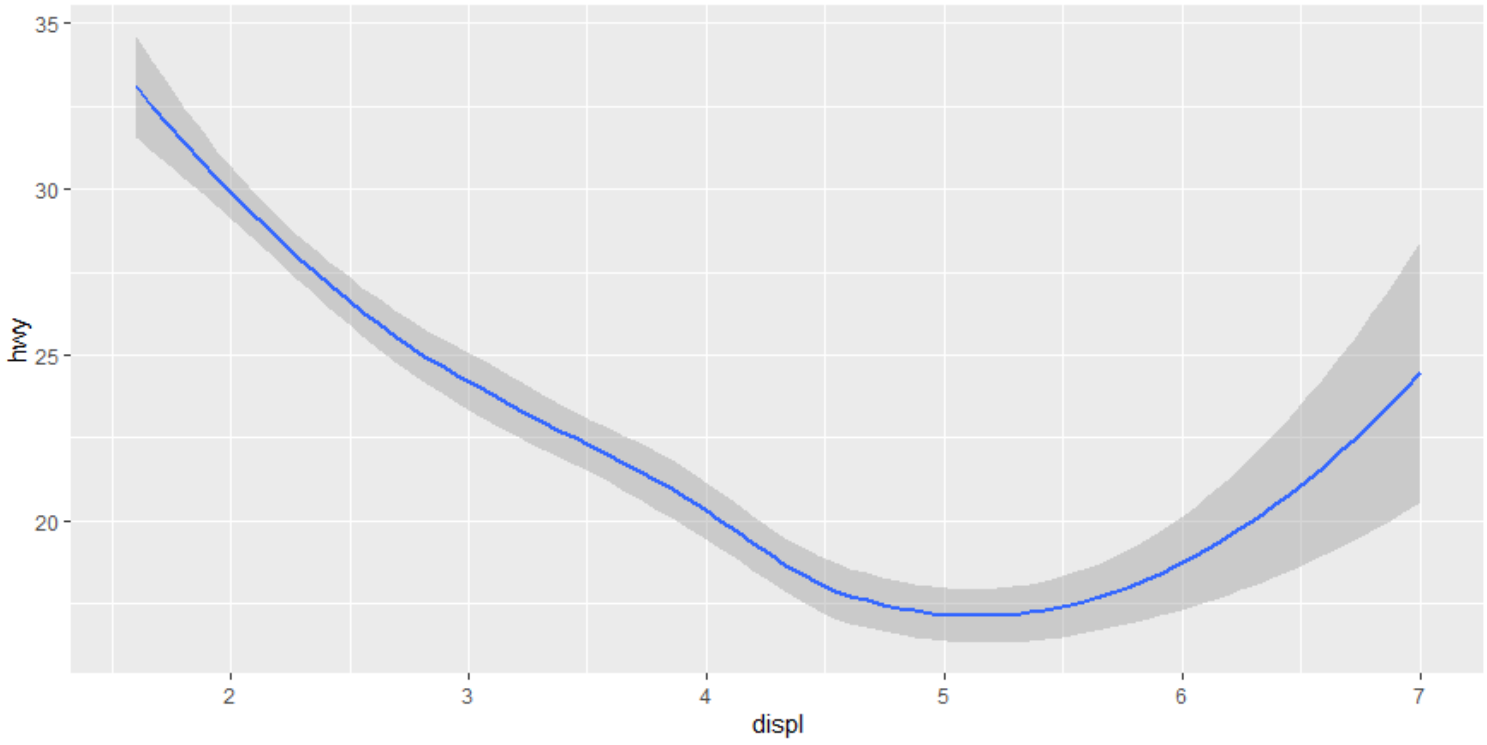
Geometric Objects

A geom is the geometrical object that a plot uses to represent data.

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy))

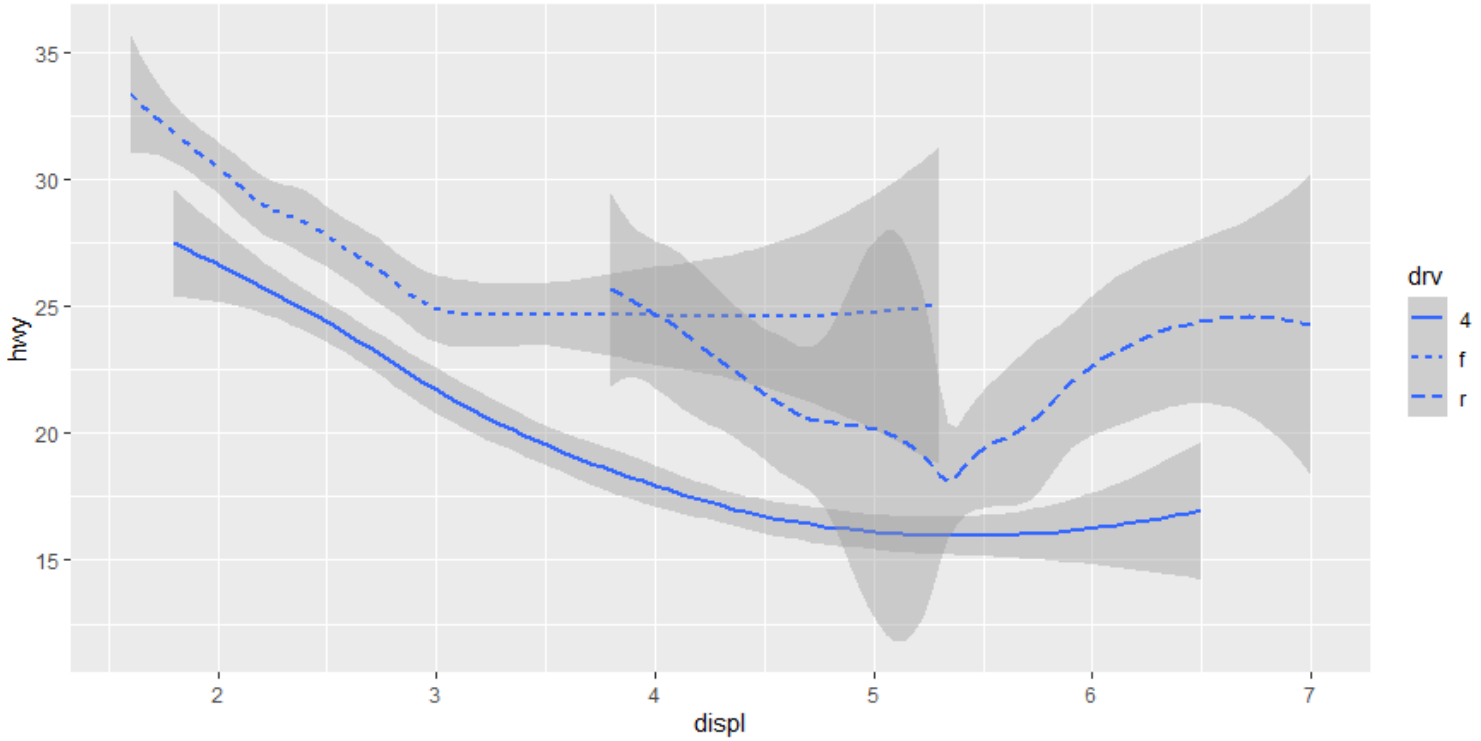
Geometric Objects

$$\text{ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy))}$$



Geometric Objects

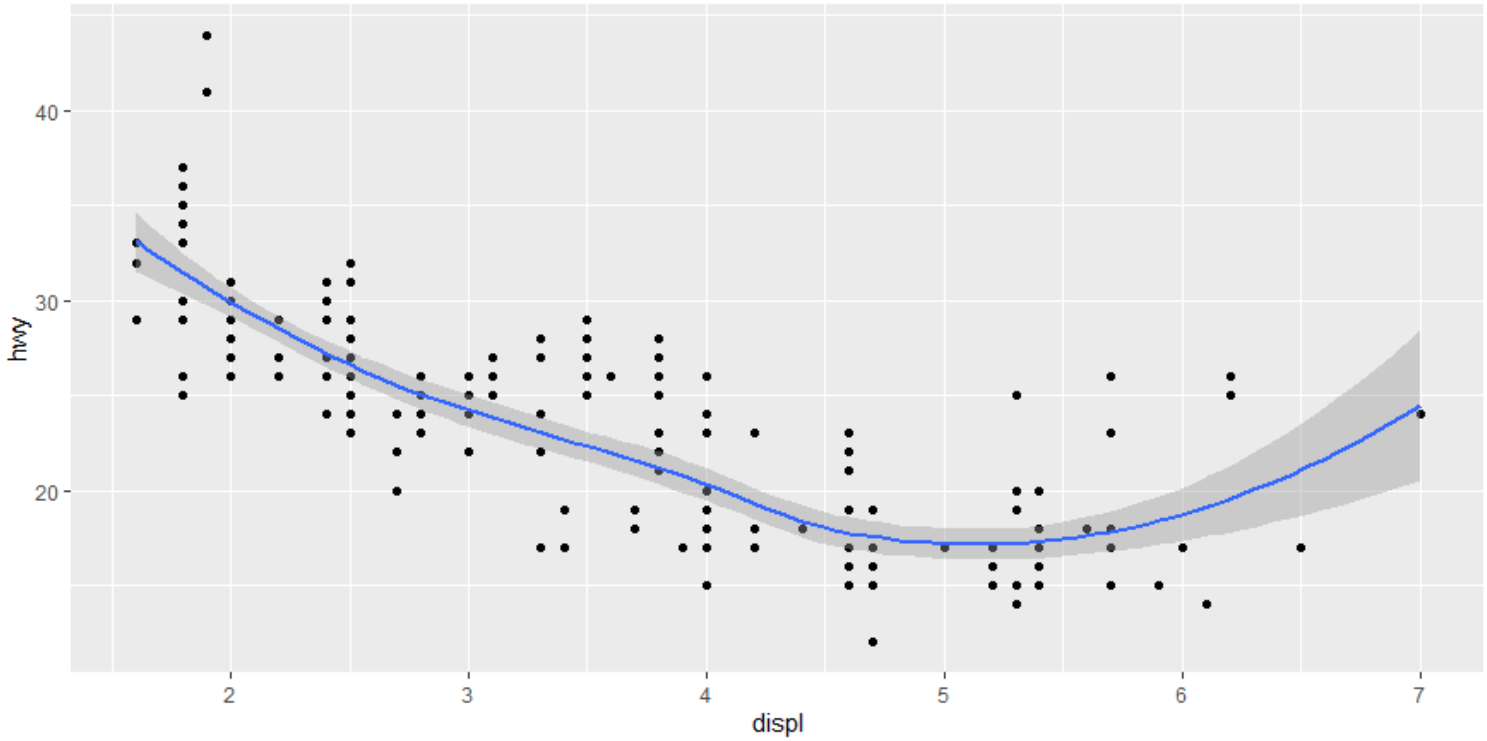
ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy, linetype = drv))



```
ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy))
ggplot(data = mpg) + geom_smooth(mapping = aes(x = displ, y = hwy,
group = drv))
ggplot(data = mpg) + geom_smooth( mapping = aes(x = displ, y = hwy,
color = drv), show.legend = FALSE )
```

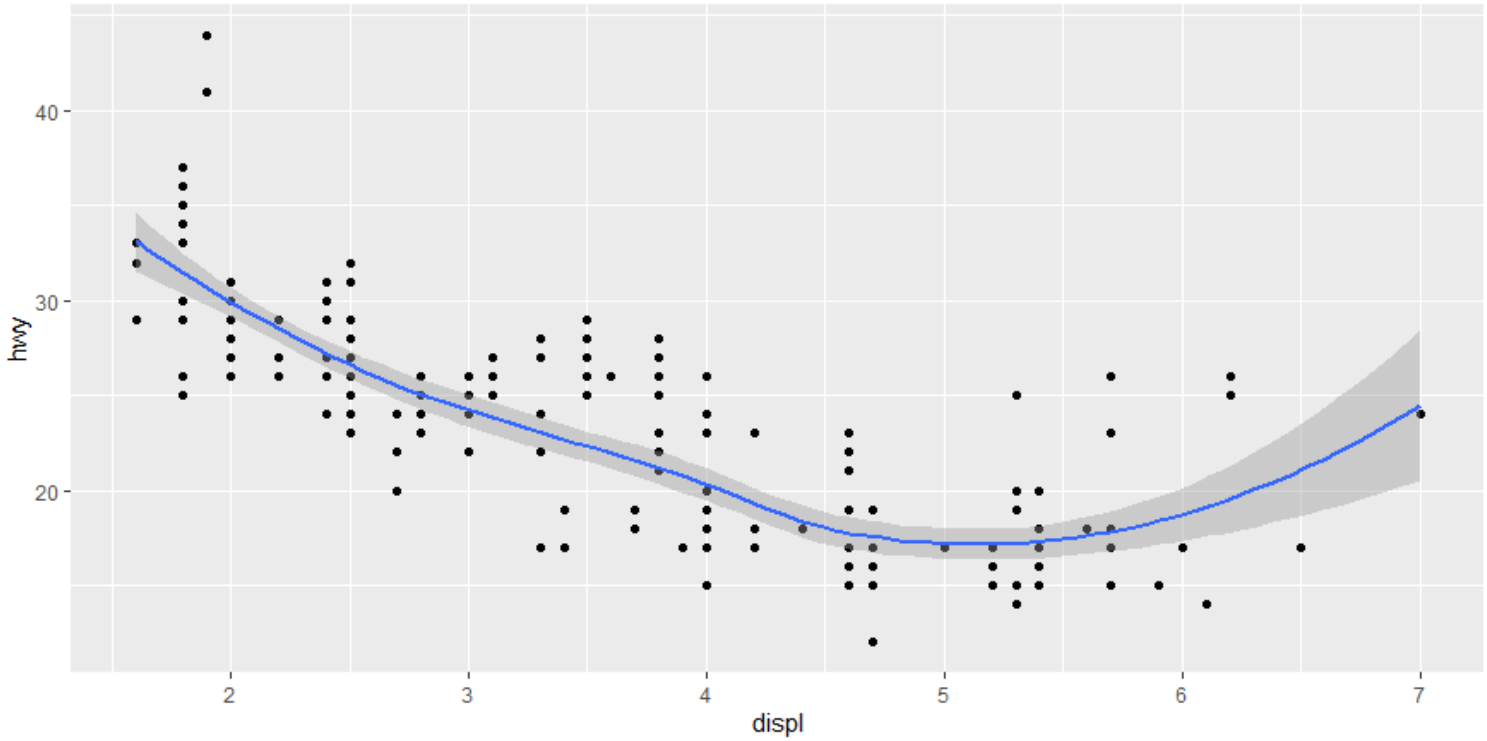
multiple geoms

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy)) + geom_smooth(mapping = aes(x = displ, y = hwy))



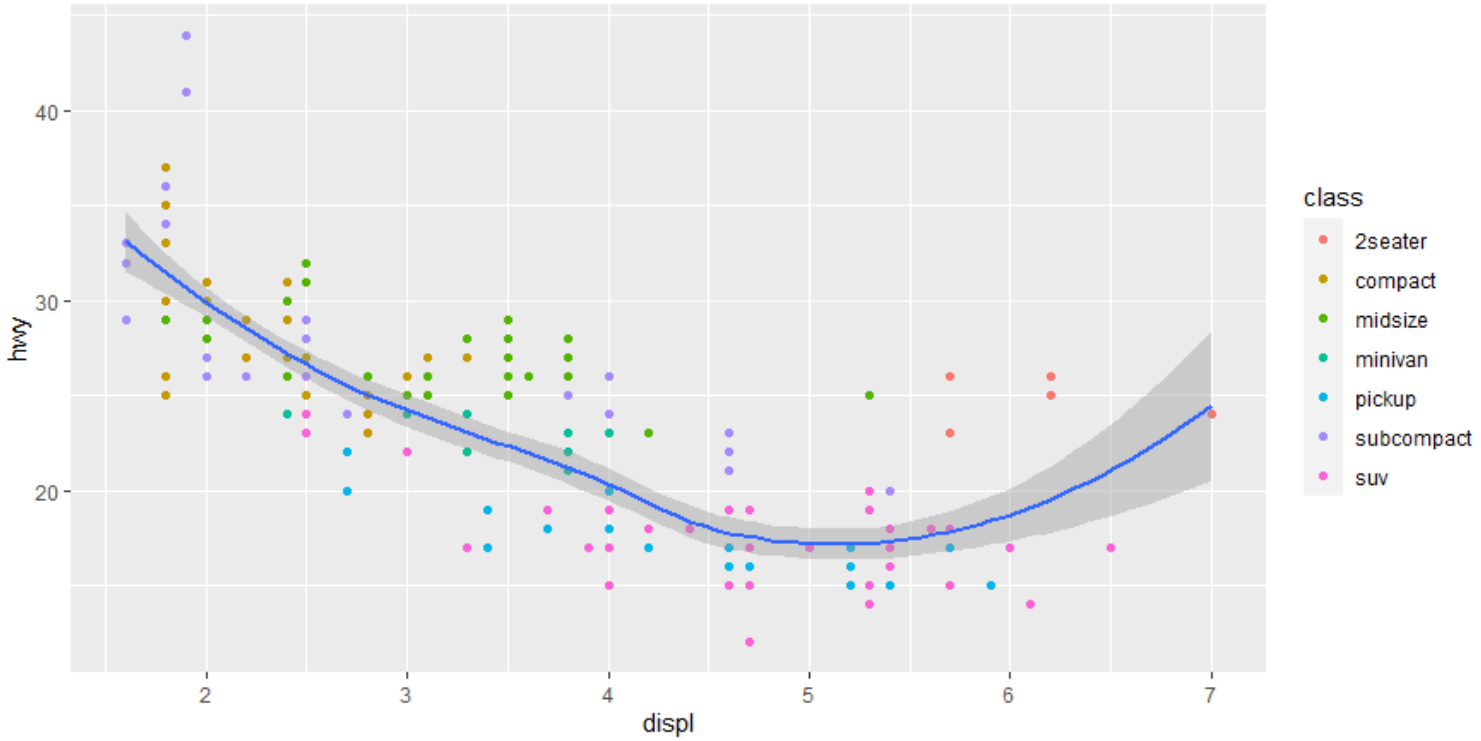
multiple geoms

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) + geom_point() + geom_smooth()



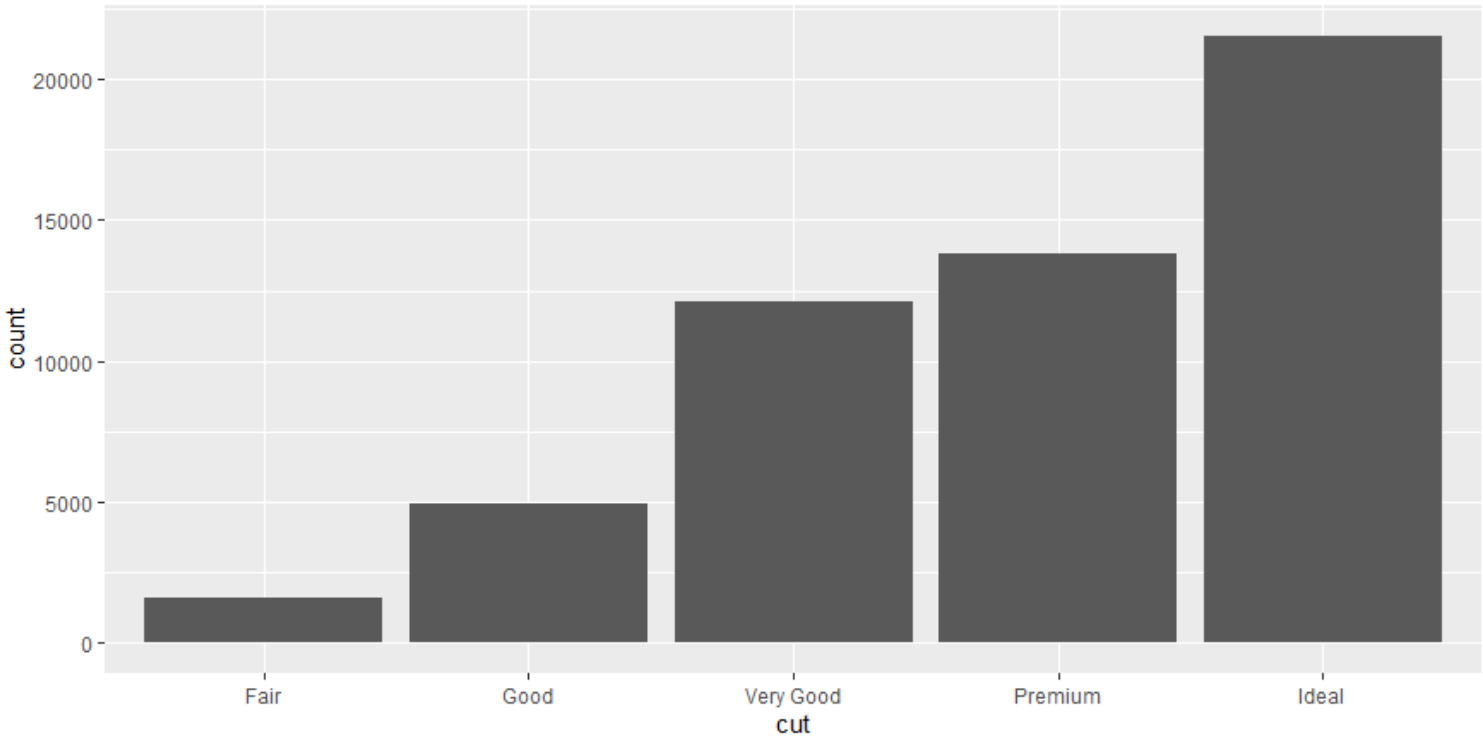
multiple geoms

ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
geom_point(mapping = aes(color = class)) + geom_smooth()



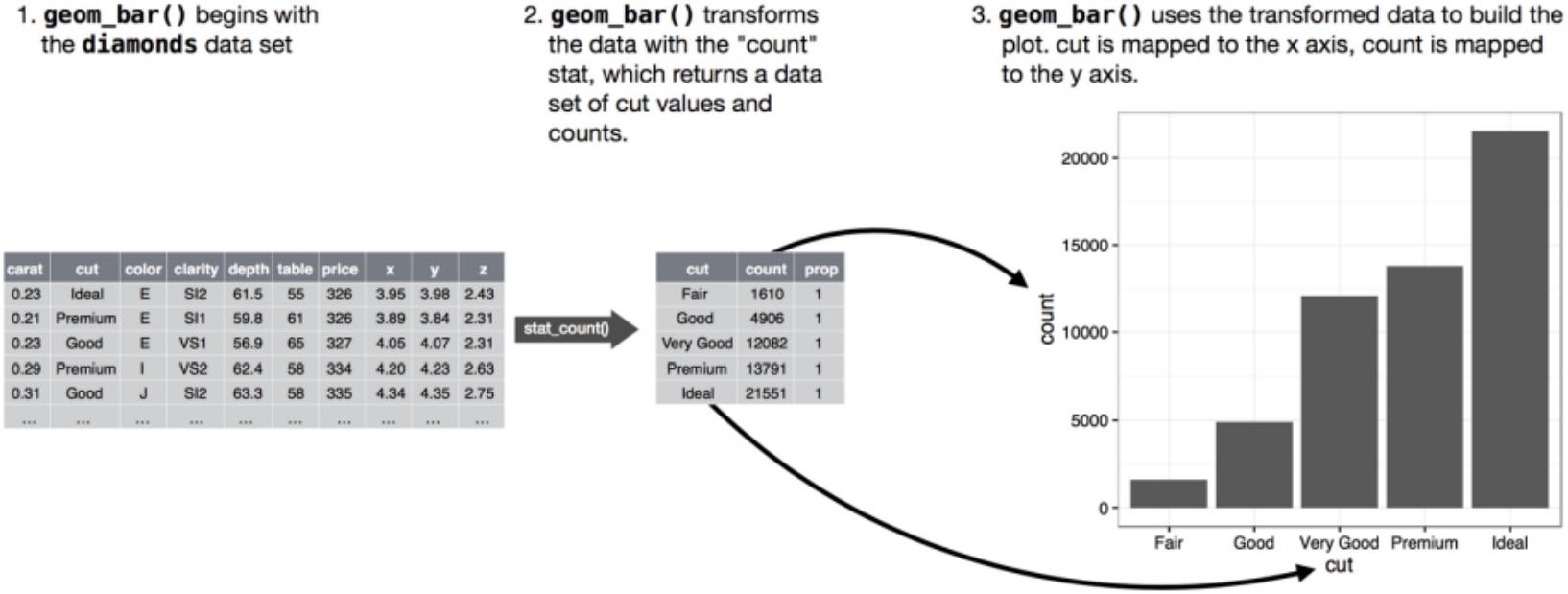
Statistical Transformations

ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut))



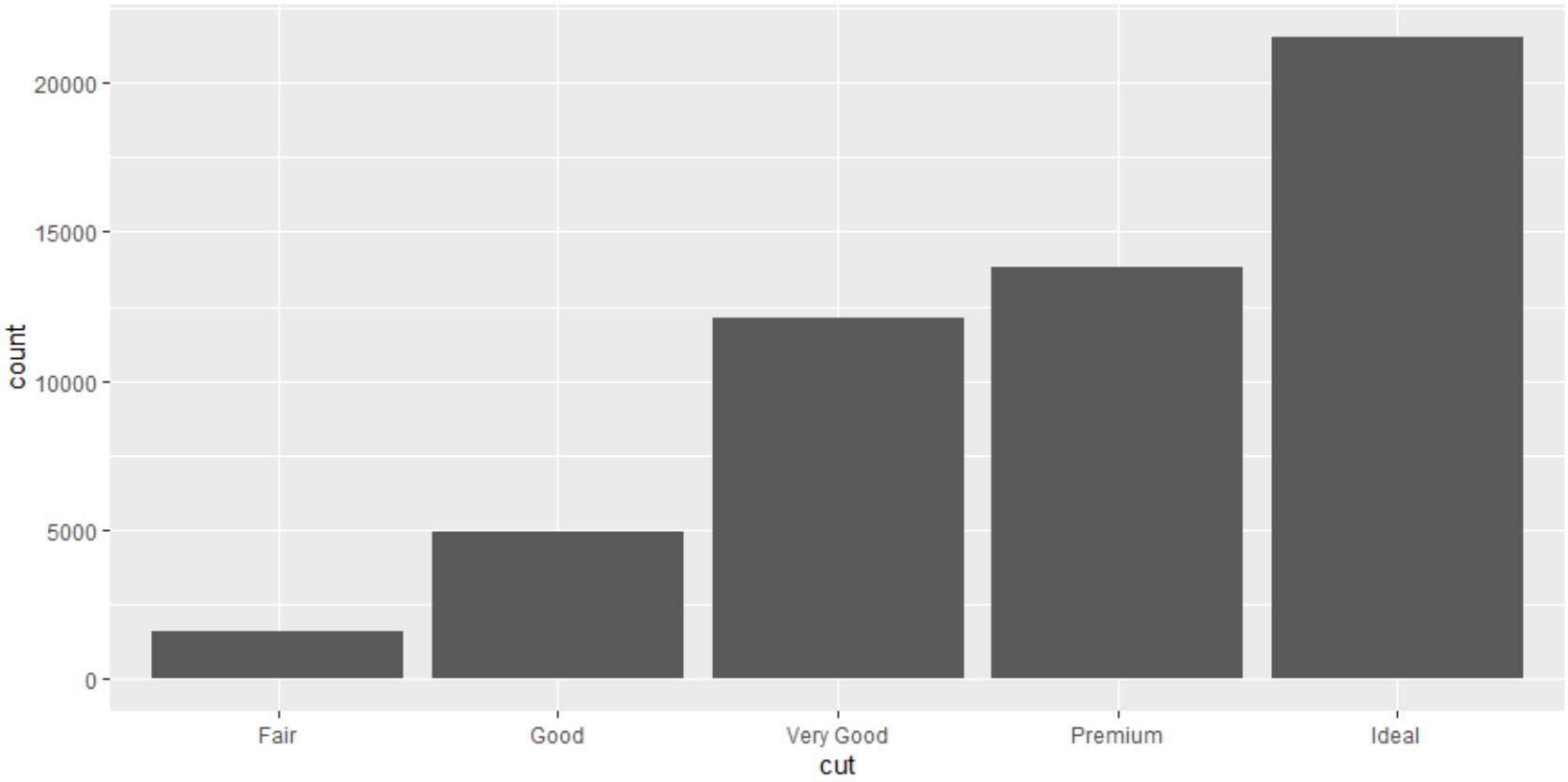
Statistical Transformations

$$\text{ggplot}(\text{data} = \text{diamonds}) + \text{geom_bar}(\text{mapping} = \text{aes}(\text{x} = \text{cut}))$$



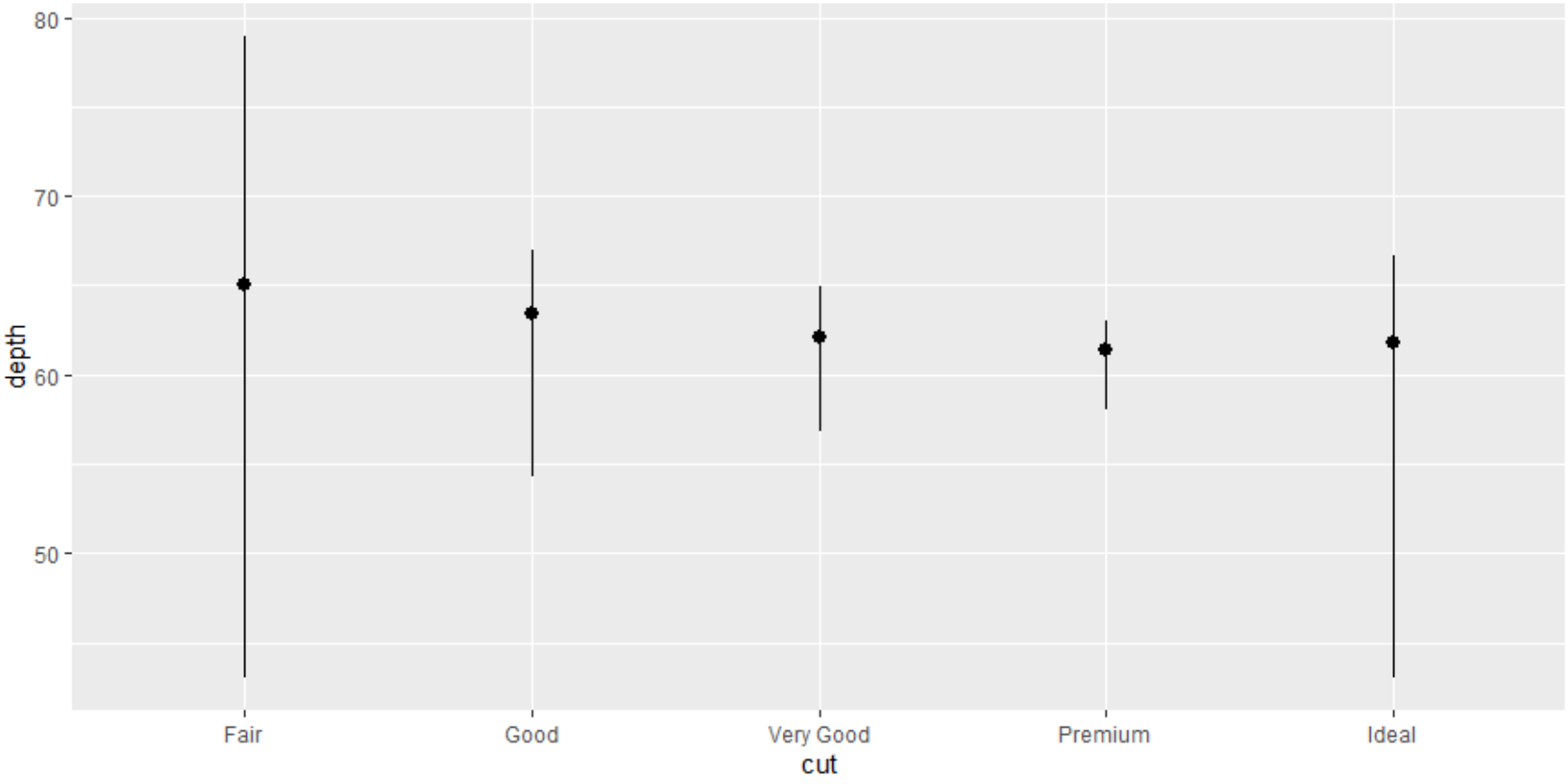
Statistical Transformations

ggplot(data = diamonds) + stat_count(mapping = aes(x = cut))



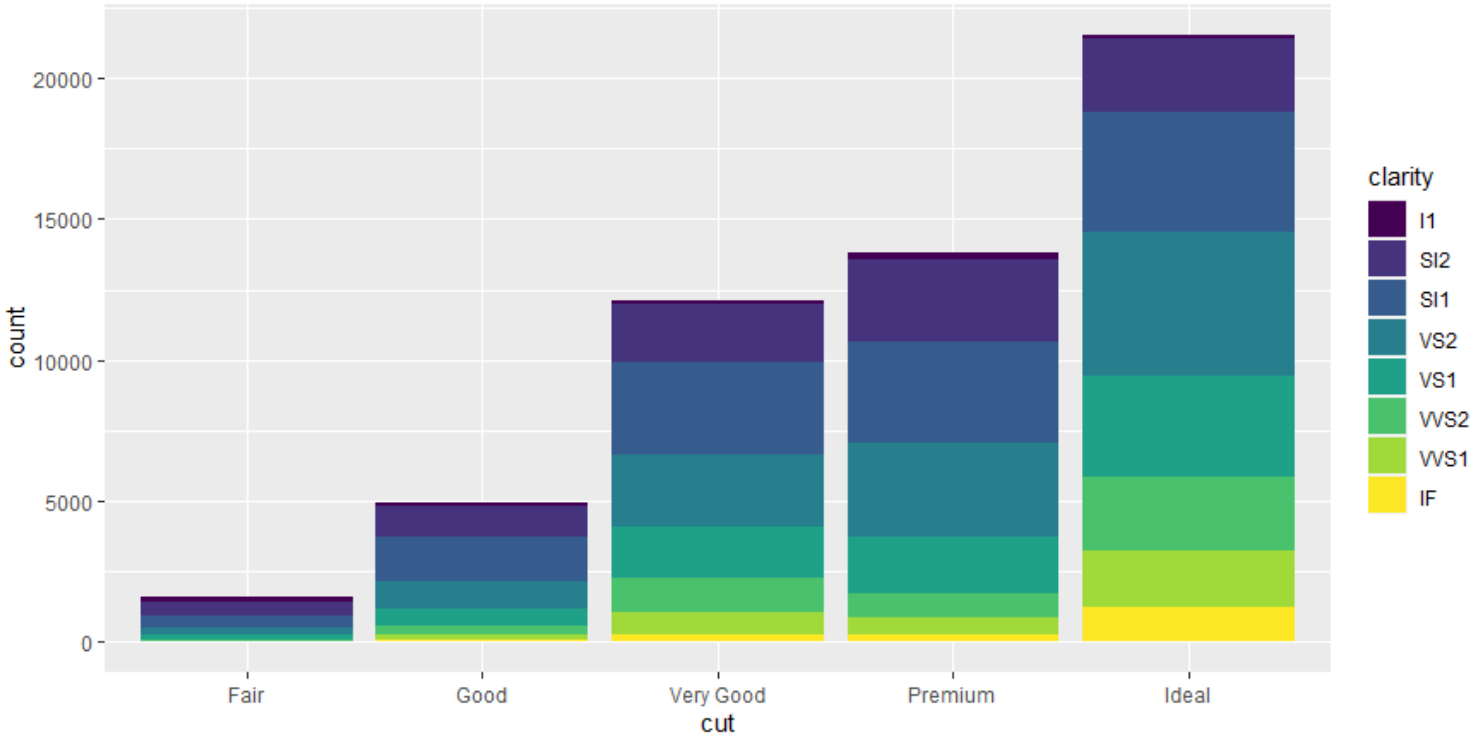
Statistical Transformations

ggplot(data = diamonds) + stat_summary(mapping = aes(x = cut, y = depth), fun.ymin = min, fun.ymax = max, fun.y = median)



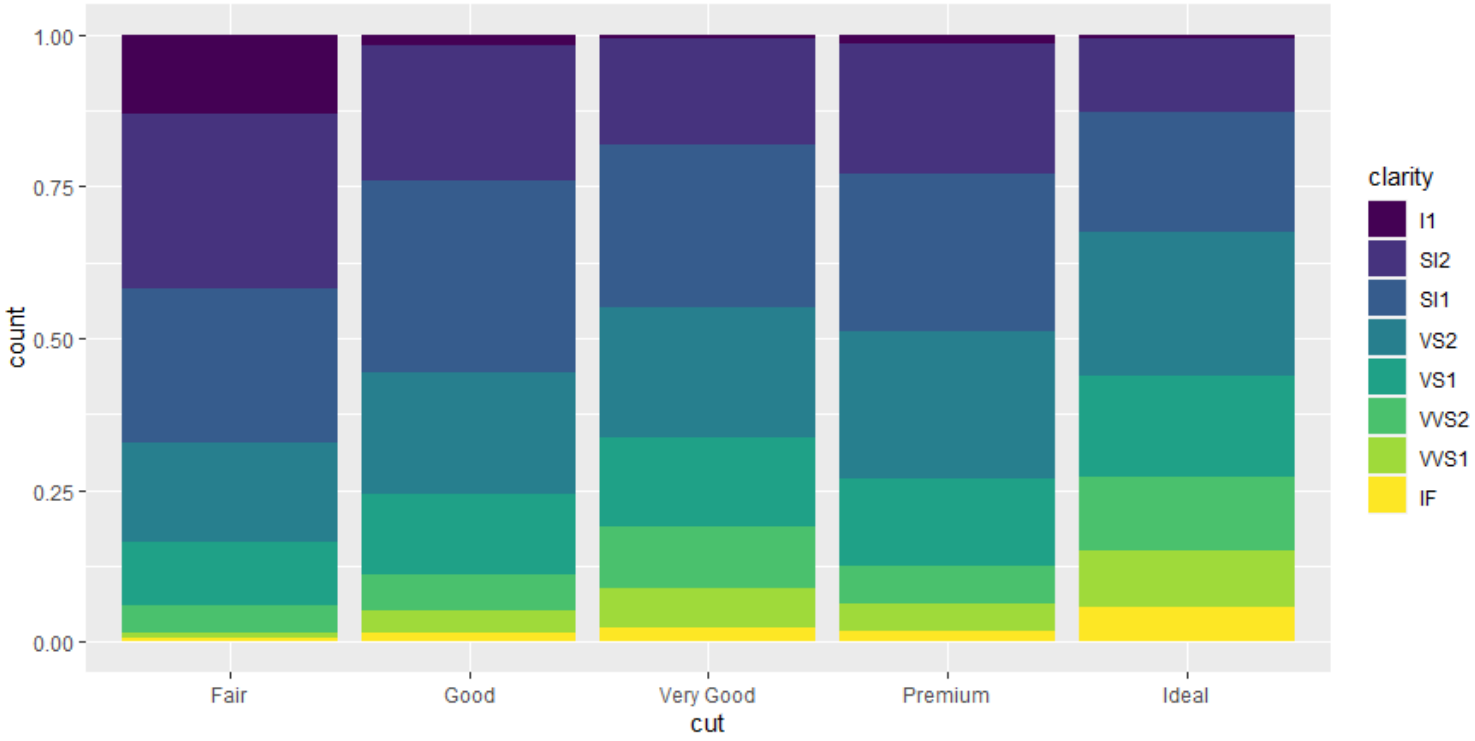
Statistical Transformations

ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut, fill = clarity))



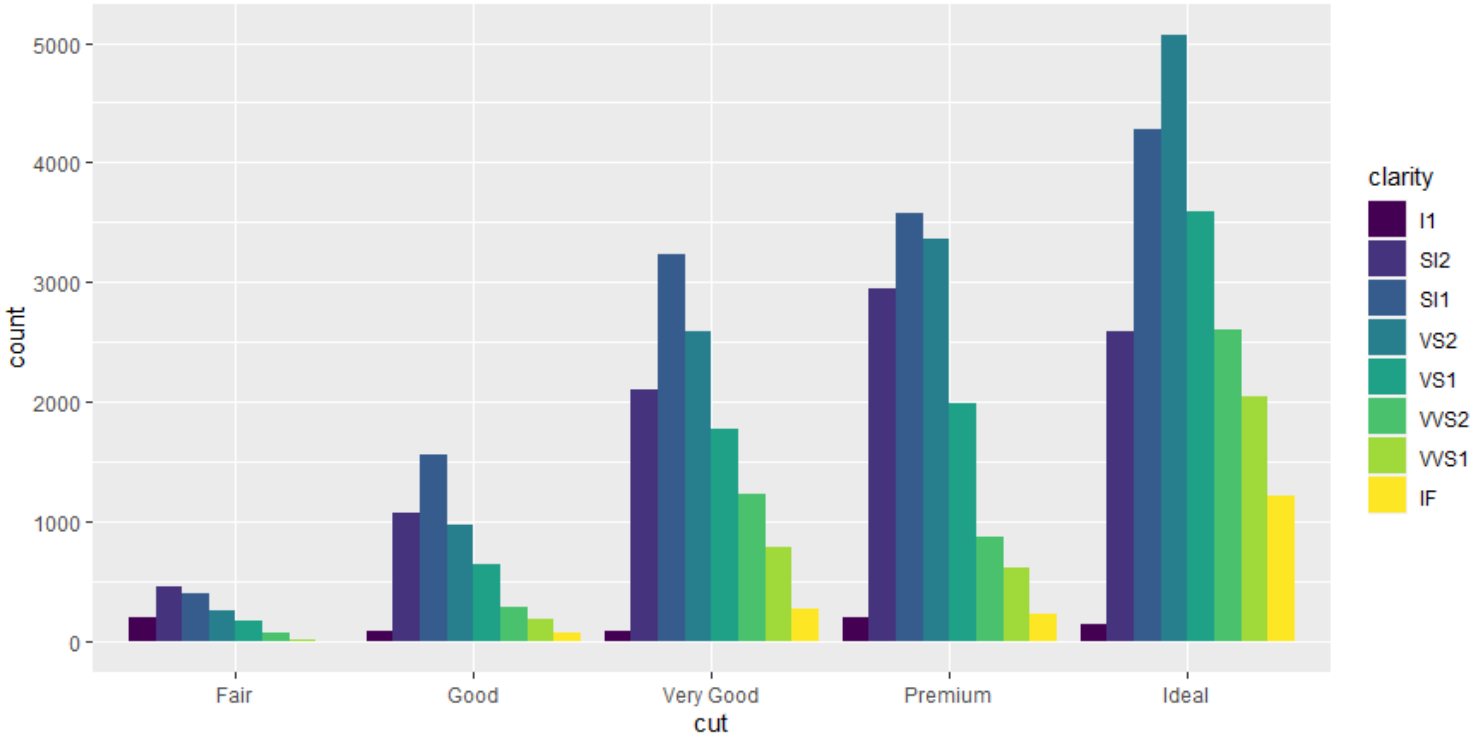
Statistical Transformations

ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut, fill = clarity), position = "fill")



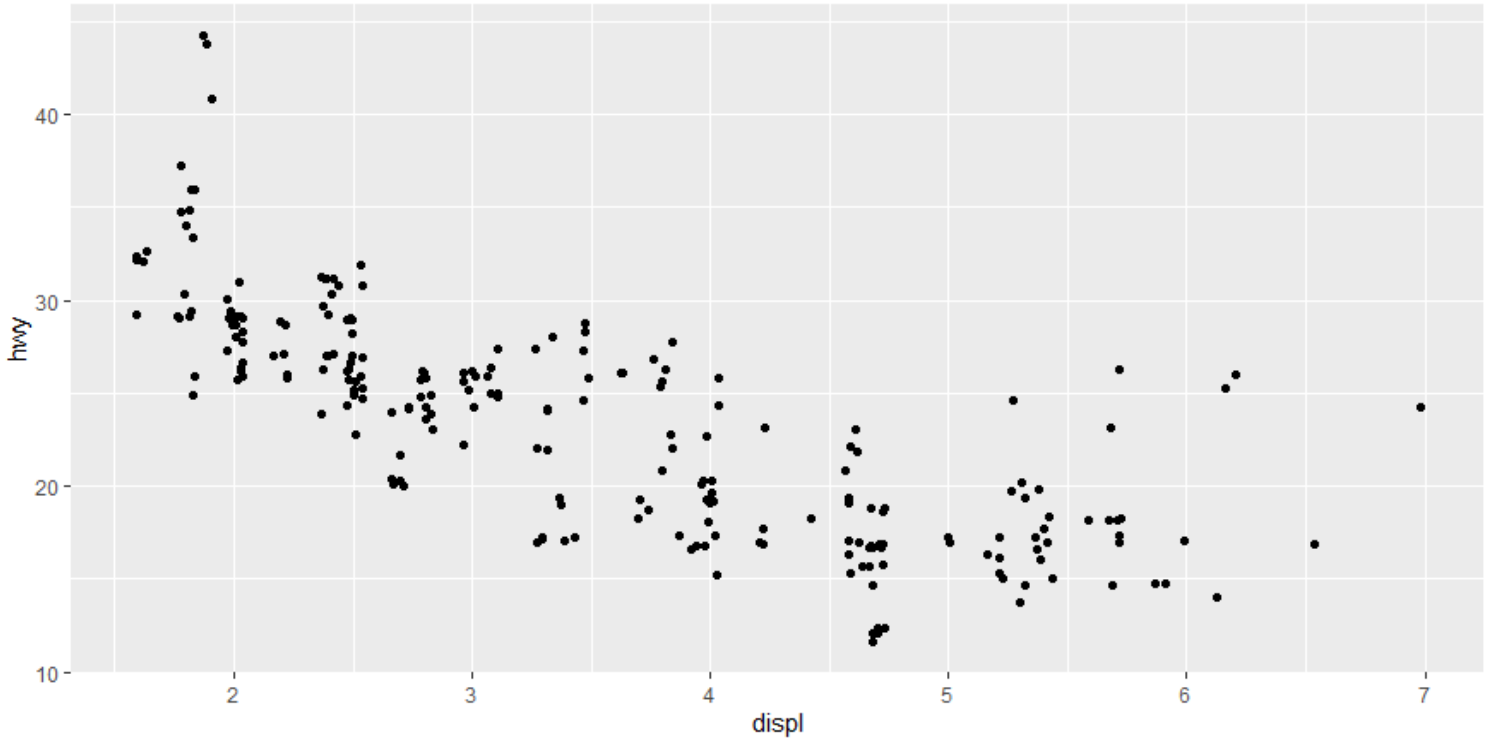
Statistical Transformations

ggplot(data = diamonds) + geom_bar(mapping = aes(x = cut, fill = clarity), position = "dodge")



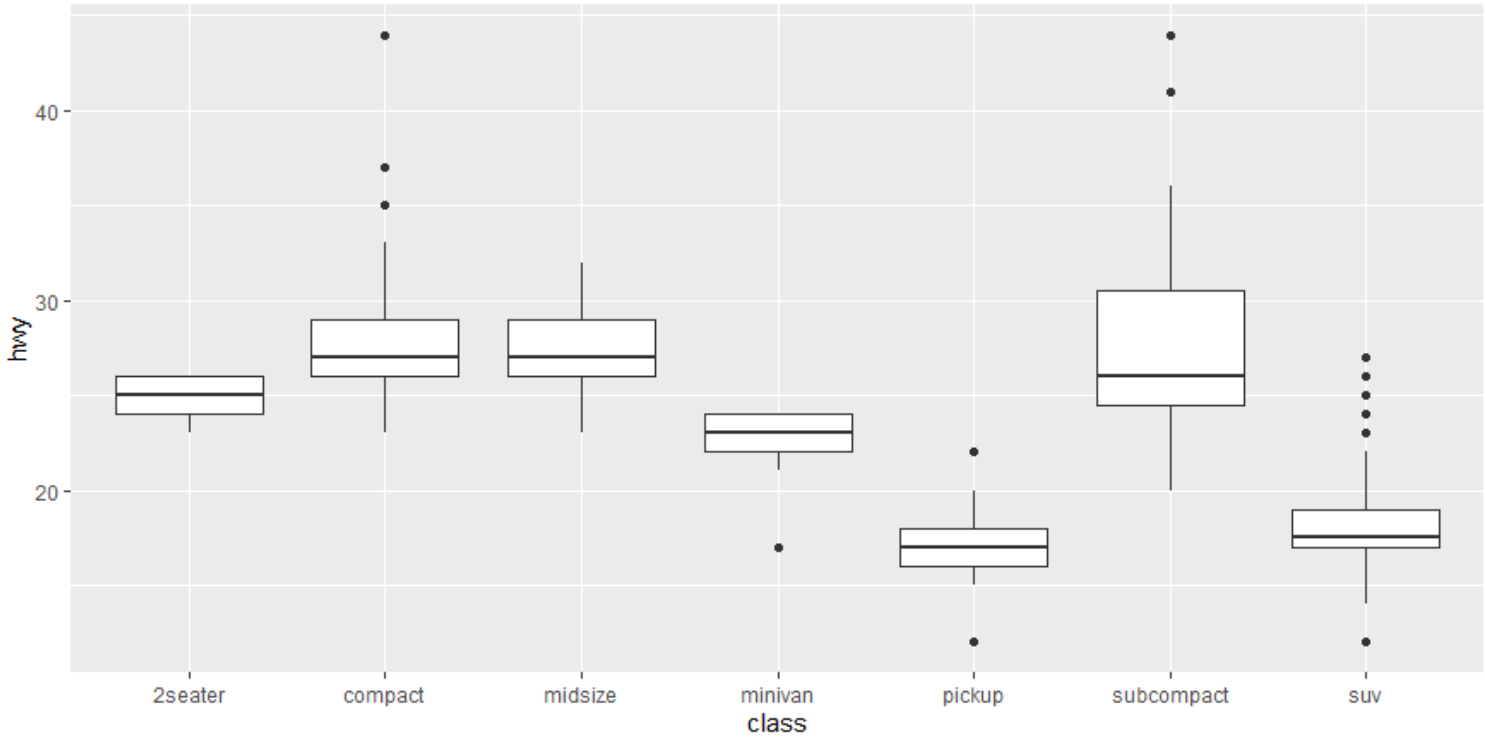
Statistical Transformations

ggplot(data = mpg) + geom_point(mapping = aes(x = displ, y = hwy), position = "jitter")



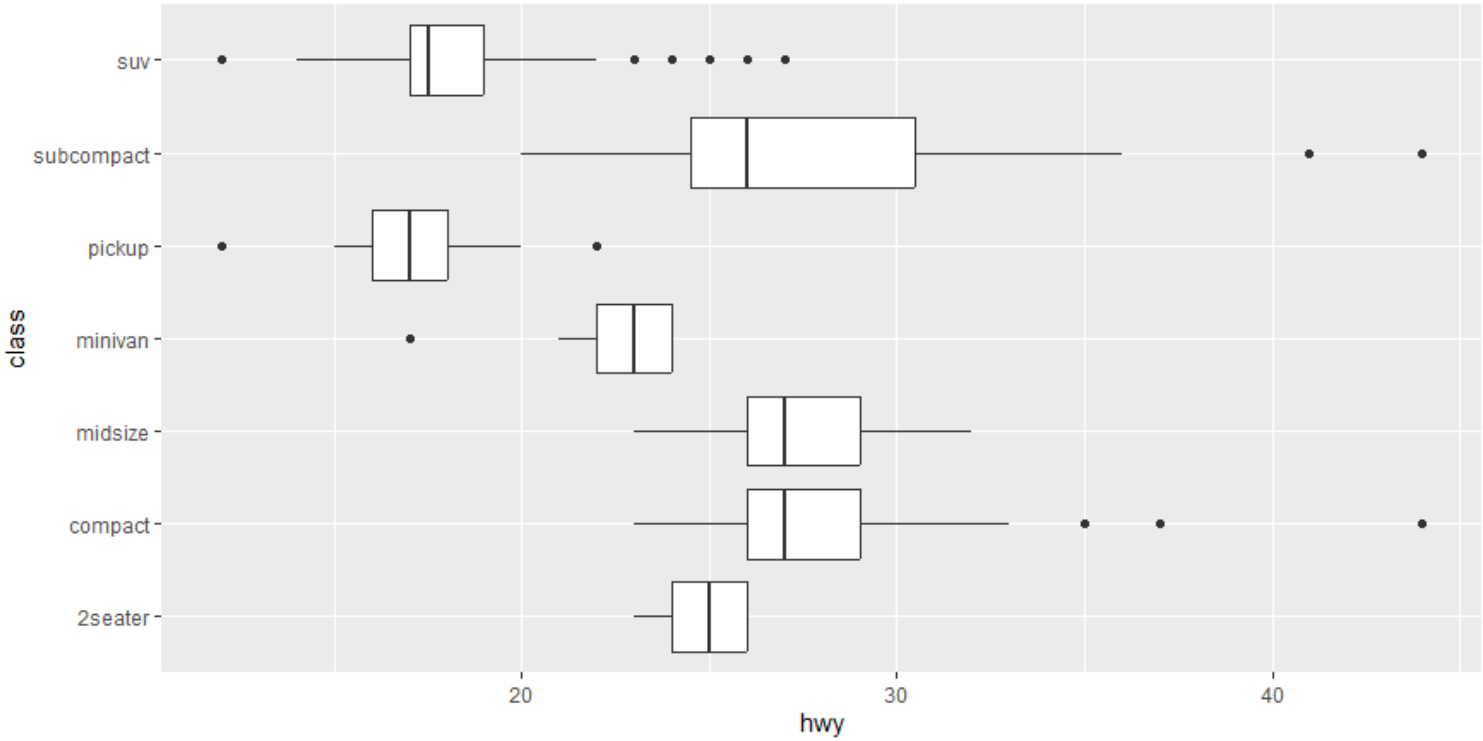
Coordinate Systems

ggplot(data = mpg, mapping = aes(x = class, y = hwy)) + geom_boxplot()



Coordinate Systems

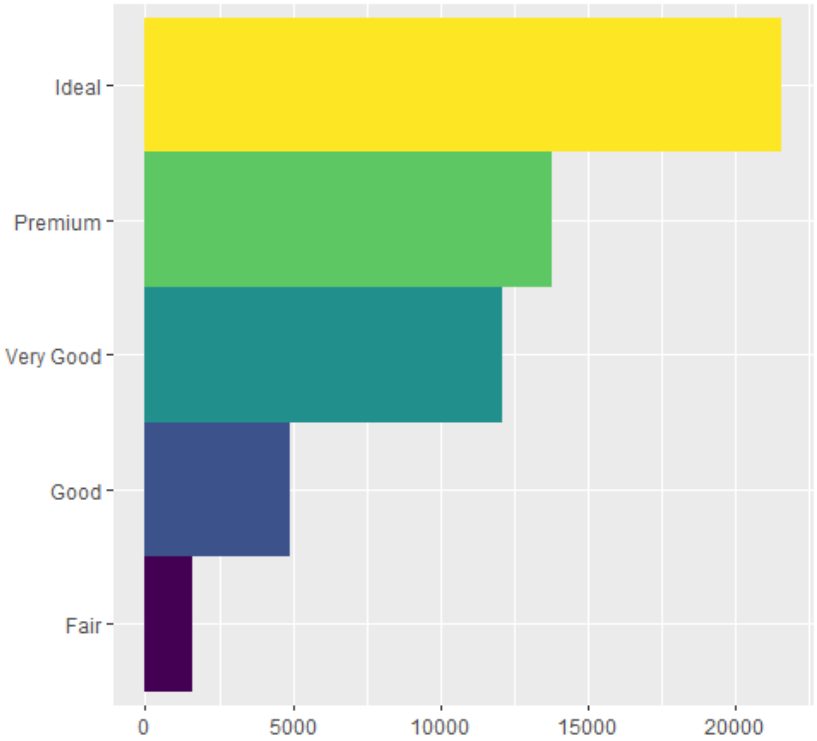
ggplot(data = mpg, mapping = aes(x = class, y = hwy)) + geom_boxplot() + coor_flip()



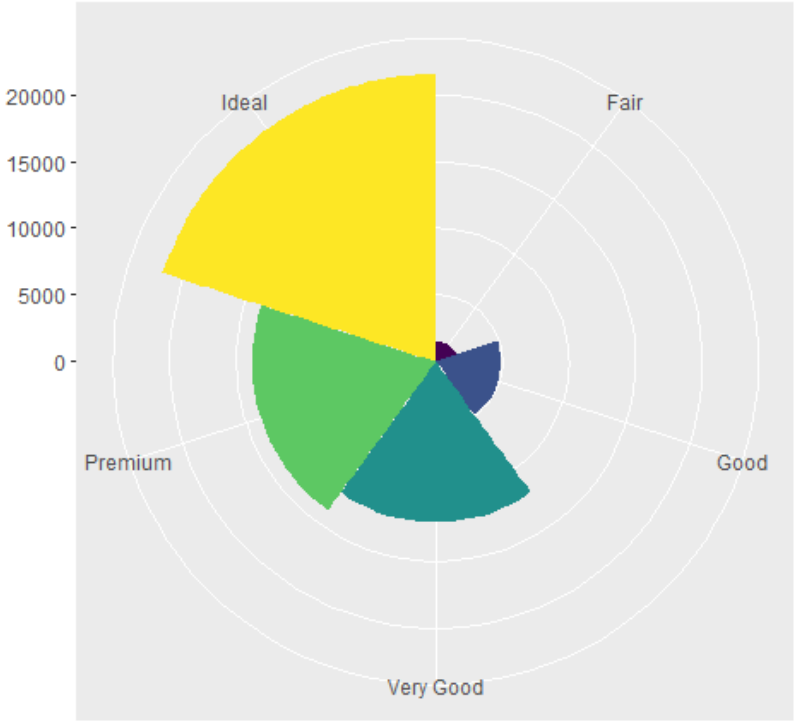
Coordinate Systems

```
bar <- ggplot(data = diamonds) + geom_bar( mapping = aes(x = cut, fill
= cut), show.legend = FALSE, width = 1 ) +
theme(aspect.ratio = 1) + labs(x = NULL, y = NULL)
bar + coord_flip()
bar + coord_polar()
```

Coordinate Systems



Coordinate Systems



谢谢。