Lecture1
课程绪论

刘晨辉
邮箱：chenhuiliu@hnu.edu.cn
办公室：土木楼A422
2023.02.28

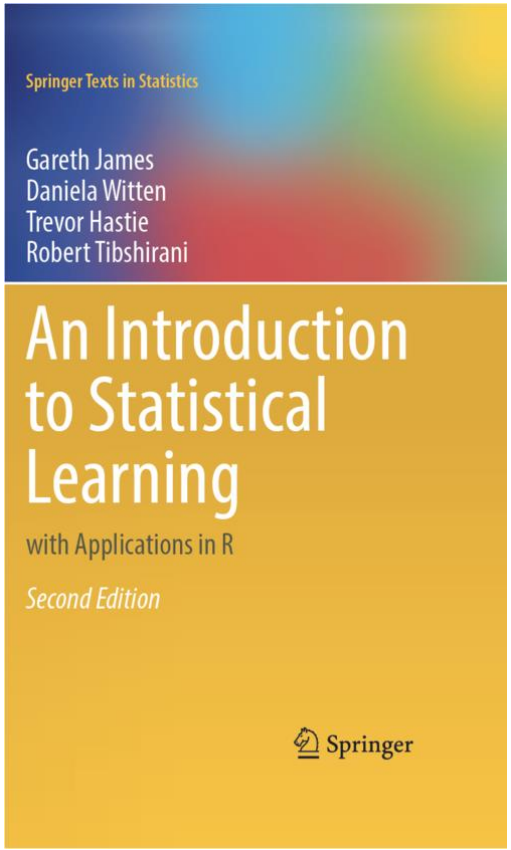
目录

1. 课程简介

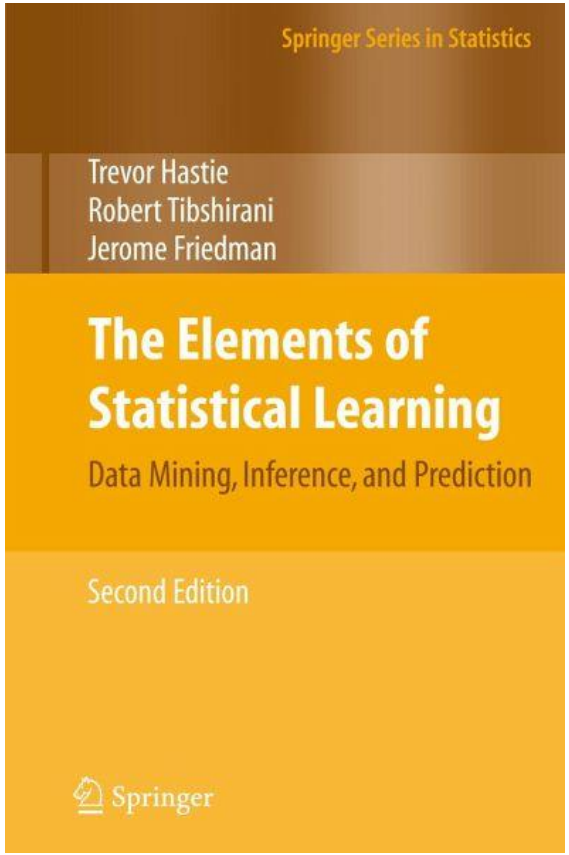
课程名称	应用数据分析与建模介绍(GE14208)
课程时间	3-13周，周二，9-11节
课程地点	复204
编程语言	R语言
课程考核	10%课堂表现+60%平时作业+30%课程项目
答疑时间	周二，下午4-6点
答疑地点	土木楼A422
联系方式	chenhuiliu@hnu.edu.cn
课程基础	《概率论与数理统计》，《线性代数》
课程目标	帮助大家了解基本的数据分析与建模知识，能利用R完成简单的数据分析项目。

1. 课程简介

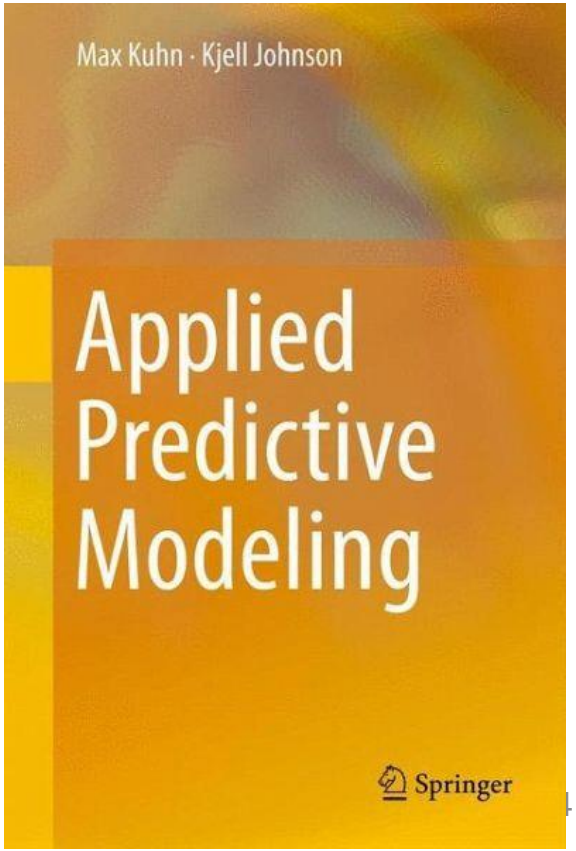
参考书目：三本书电子版均可以免费下载！



https://www.statlearning.com/



https://hastie.su.domains/ElemStatLearn/



http://appliedpredictivemodeling.com/

2. 数据分析背景

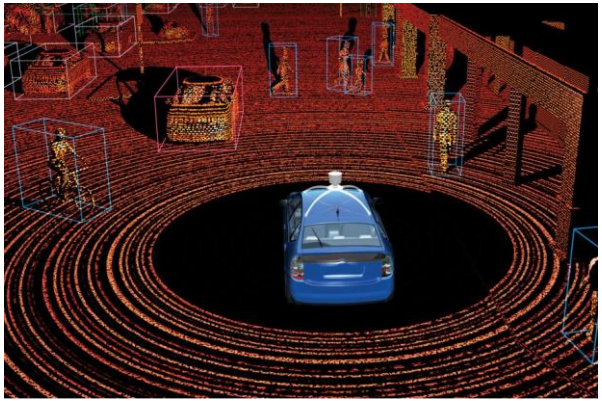
支撑科技发展的基础工作之一是数据分析！

2. 数据分析背景

数据案例1：交通数据。从天马公寓到教室，你需要走多长时间?

2. 数据分析背景

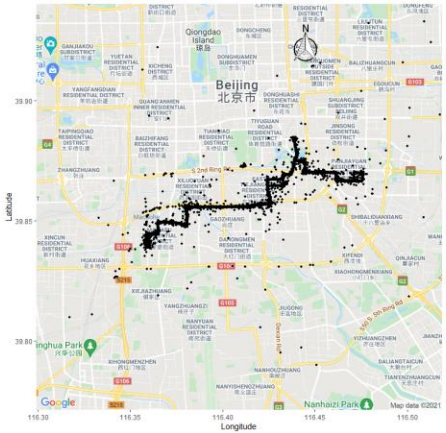
Bluetooth®



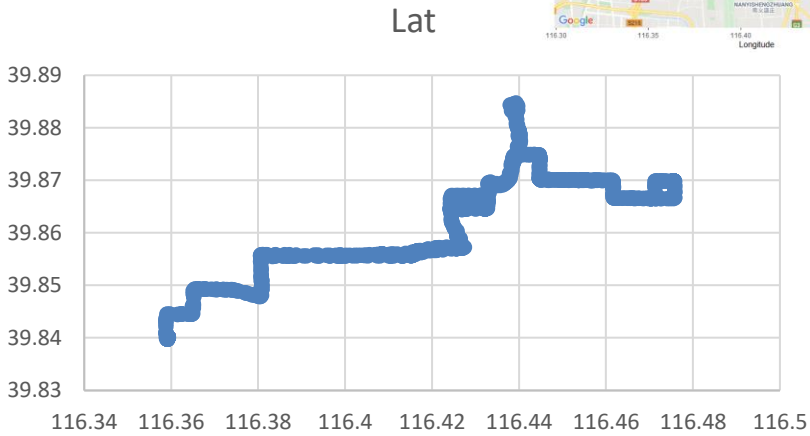
2. 数据分析背景

北京某电动公交车轨迹数据(ebus.csv)：

时间，坐标，速度，加速度等。



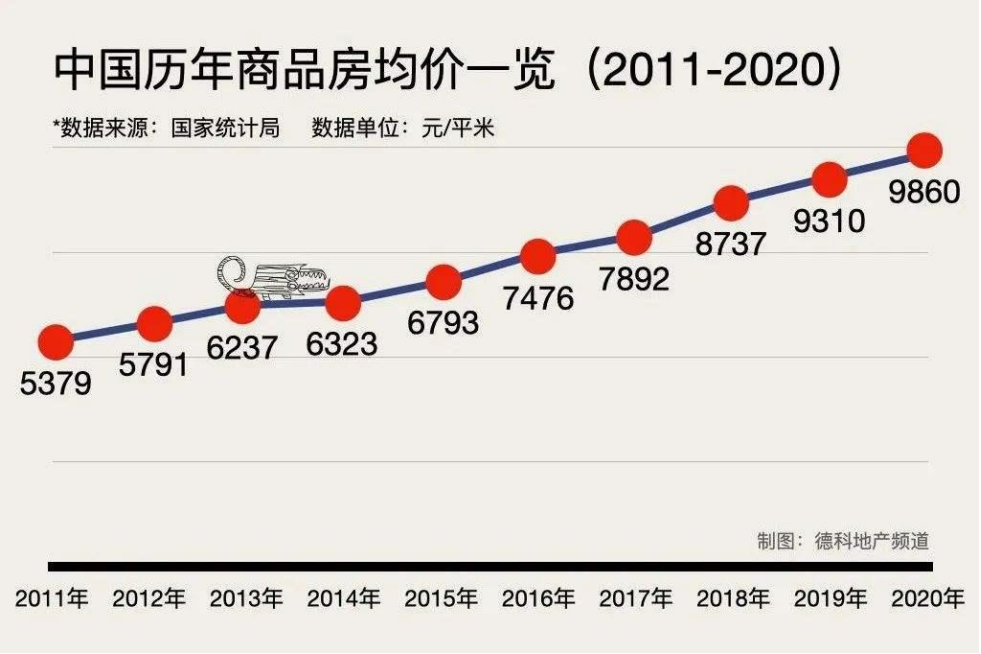
	A	B	C	D	E	F
1	vin	Time	Lon	Lat	Speed	Operation
2	LVCB4L4D3HM002839	5/20/2020 5:47	116.3592	39.83979	0	1
3	LVCB4L4D3HM002839	5/20/2020 5:47	116.3592	39.83979	0	1
4	LVCB4L4D3HM002839	5/20/2020 5:47	116.3592	39.83979	0	1
5	LVCB4L4D3HM002839	5/20/2020 5:48	116.3592	39.83979	0	1
6	LVCB4L4D3HM002839	5/20/2020 5:48	116.3592	39.83979	0	1
7	LVCB4L4D3HM002839	5/20/2020 5:48	116.3592	39.83979	0	1
8	LVCB4L4D3HM002839	5/20/2020 5:48	116.3592	39.83979	0	1
9	LVCB4L4D3HM002839	5/20/2020 5:49	116.3592	39.83979	15.7	1
10	LVCB4L4D3HM002839	5/20/2020 5:49	116.3592	39.83979	4.1	1
11	LVCB4L4D3HM002839	5/20/2020 5:49	116.3592	39.83979	26.8	1



- 速度：最大值，最小值，平均值，…
- 轨迹：在excel中画出来

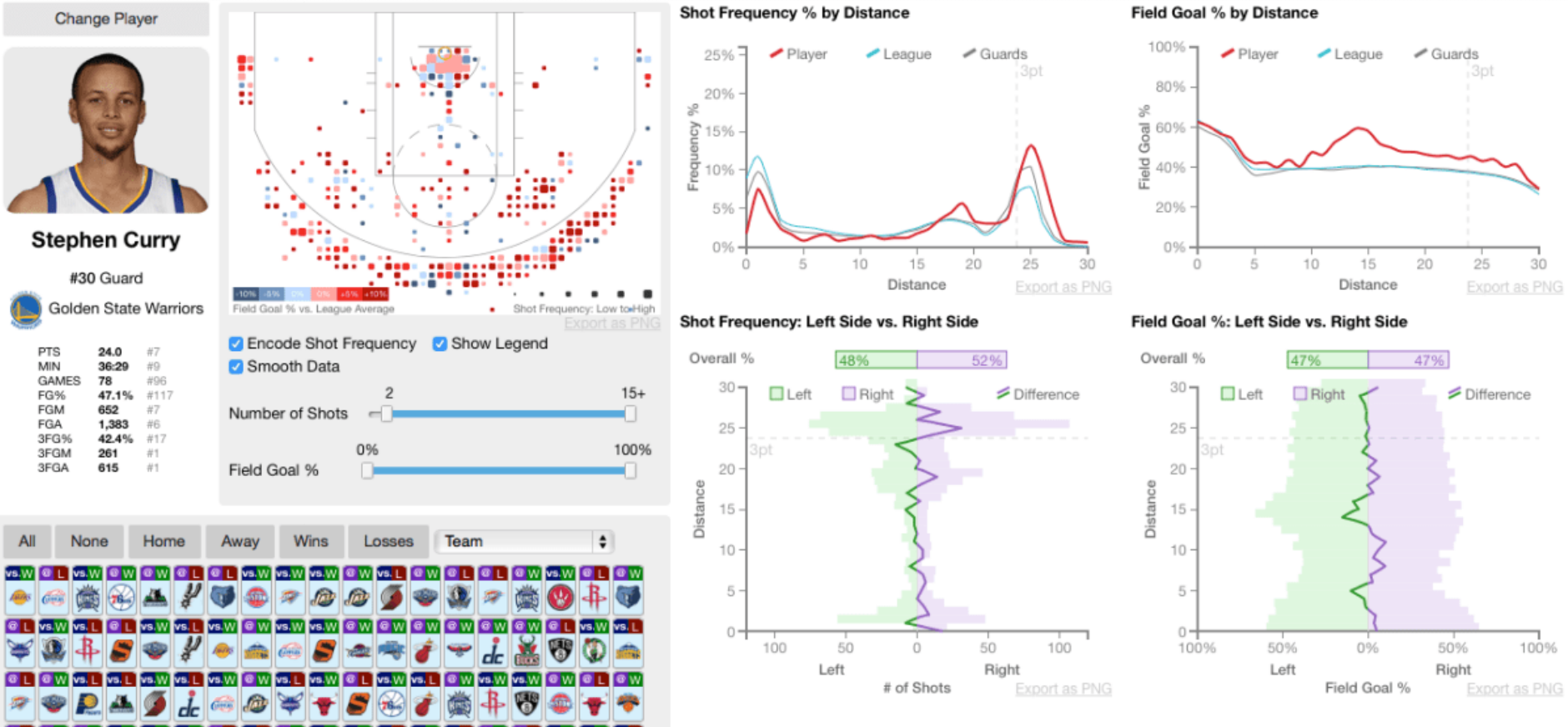
2. 数据分析背景

数据案例2：房价数据。房价会继续上涨吗？



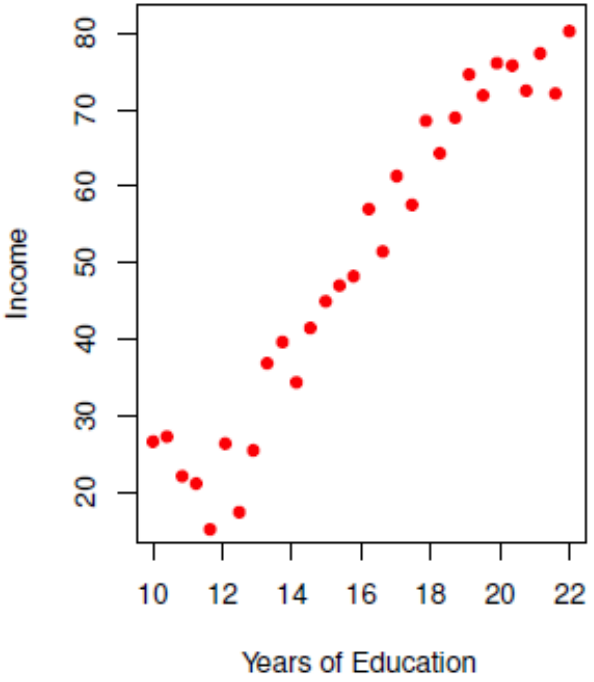
2. 数据分析背景

数据案例3：体育数据。库里的投球命中率有多少？



3. 统计学习简介

3.1 什么是统计学习?



假设观察到某个量化响应值Y，以及一系列预测变量，X_1，X_2，…，X_p。假设Y与$X=(X_1,X_2,\ldots,X_p)$存在某种关系，那么可以表示成

$$Y=f(X)+\epsilon$$

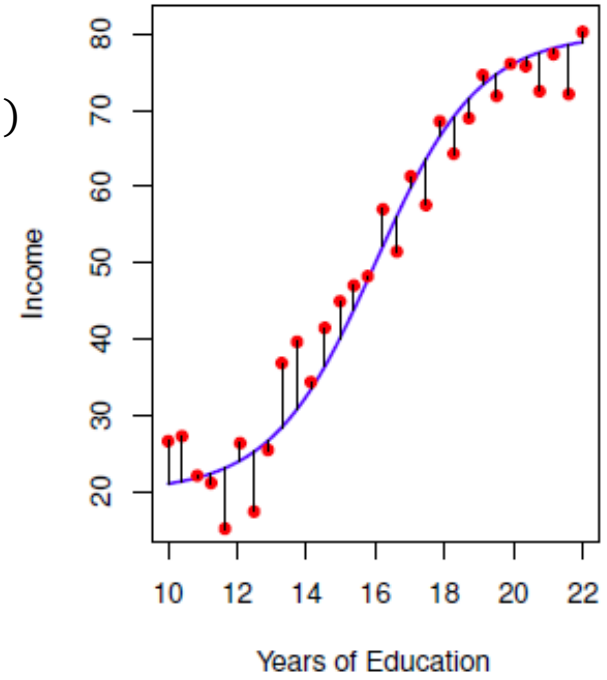
其中，

ϵ是一个随机误差项，与X相独立，且平均值为0；

f是一个未知的公式，代表了X提供的关于Y的系统信息，也是我们求解的目标。

统计学习本质上就是使用一系列的方法来估算f。

- 回归(Regression)
- 分类(Classification)



3. 统计学习简介

3.2 为什么估算f?

预测 (Prediction)

- $\hat{Y} = \hat{f}(X)$
- $\hat{f}$ 经常被视为黑箱(Blackbox)。只要预测精度够高，我们有时并不关心它的精确形式。

推导 (Inference)

- $\hat{Y} = \hat{f}(X)$
- $\hat{f}$也被视为黑箱(Blackbox)，但我们需要知道它的精确形式。

$$E\left[\left(Y-\hat{Y}\right)^2\right]=E\left[\left(f(X)+\epsilon-\hat{f}(X)\right)^2\right]$$
$$=\left[f(X)-\hat{f}(X)\right]^2+Var(\epsilon)$$

$\hat{Y}$的预测精度取决于两个数据：

- 可减少误差(reducible error)：$\hat{f}$
- 不可减少误差(irreducible error)：ϵ

在最小化可减少误差的前提下估算f！

问题：
哪些预测变量重要?
变量与响应值呈现什么关系?
用什么样的公式可以表征两者之间的关系?

3. 统计学习简介

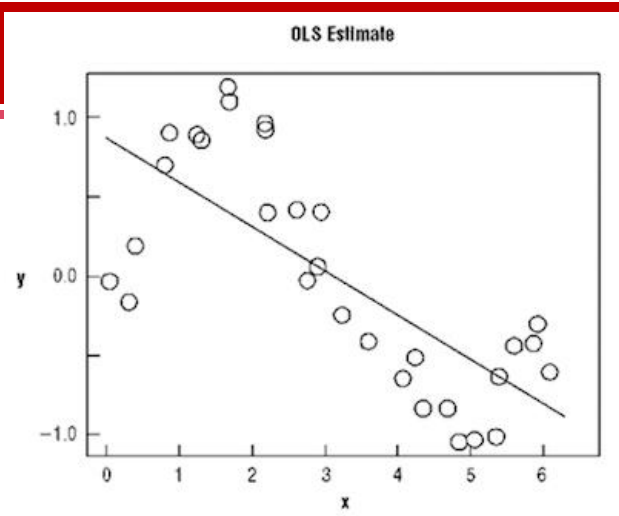
3.3 怎么估算f?

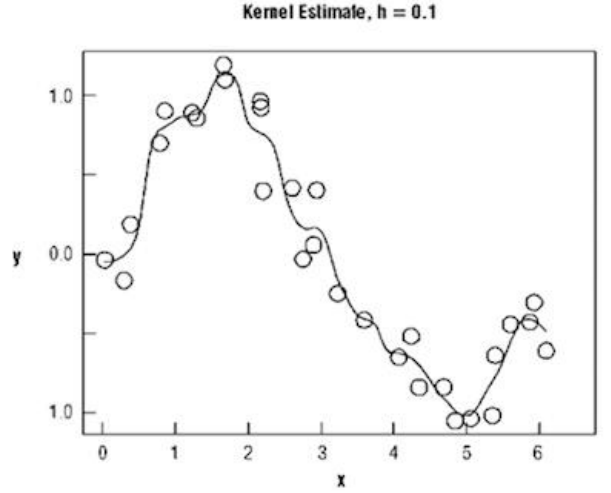
参数模型
(Parametric)

- $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$
- 需要假设模型形式

非参模型
(Non-Parametric)

- $Y = g(X)$
- 不需要假设模型形式，但是需要较大的数据集。





3. 统计学习简介

3.4 如何评价估算出的 f ?

- 准确性(Accuracy)
- 可解释性(Interpretability)
- Trade-off between prediction accuracy and model interpretability!

4. R&RStudio介绍

- R是一种开源的统计计算与绘图语言，是目前最流行的数据分析语言之一。
- R源于贝尔实验室S语言，开发者为Robert Gentleman和Ross Ihaka。
- PYPL排名: Sep 2022排名7位(https://pypl.github.io/PYPL.html/)。

Worldwide, Sept 2022 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.29 %	-1.8 %
2		Java	17.31 %	-0.7 %
3		JavaScript	9.44 %	-0.1 %
4		C#	7.04 %	-0.1 %
5		C/C++	6.27 %	-0.4 %
6		PHP	5.34 %	-1.0 %
7		R	4.18 %	+0.3 %
8	↑↑↑	TypeScript	3.05 %	+1.5 %
9	↑↑↑	Go	2.16 %	+0.6 %
10		Swift	2.11 %	+0.5 %

Languages	Best for	Features
Python	General programming, data analysis and deep learning.	Semantic way to generate complex plots.Allow to construct dynamic and interactive visualization.
R	Statistical analysis and data analysis.	Helps consider data manipulation challenges.Provides a “verbs” function to help translate the thoughts into codes.
Java	Scientific projects	Consistent usage without any disruption.Works on multiple projects at the same time.
Julia	Numerical computing with a syntax	Deep learning frameworksupports GPU operation and automatic differentiation.
Scala	Support object-oriented programming and functional languages.	Navigate the graph of related entitiesCarry out client-side validation
MATLAB	Array manipulation and specialized math functions	Allows the other modules to be loaded simultaneously
TensorFlow	Numerical computation and large scale machine learning.	Works with mathematical expressions efficiently. Deep neural networks and machine learning principles are well supported.

4. R&RStudio介绍

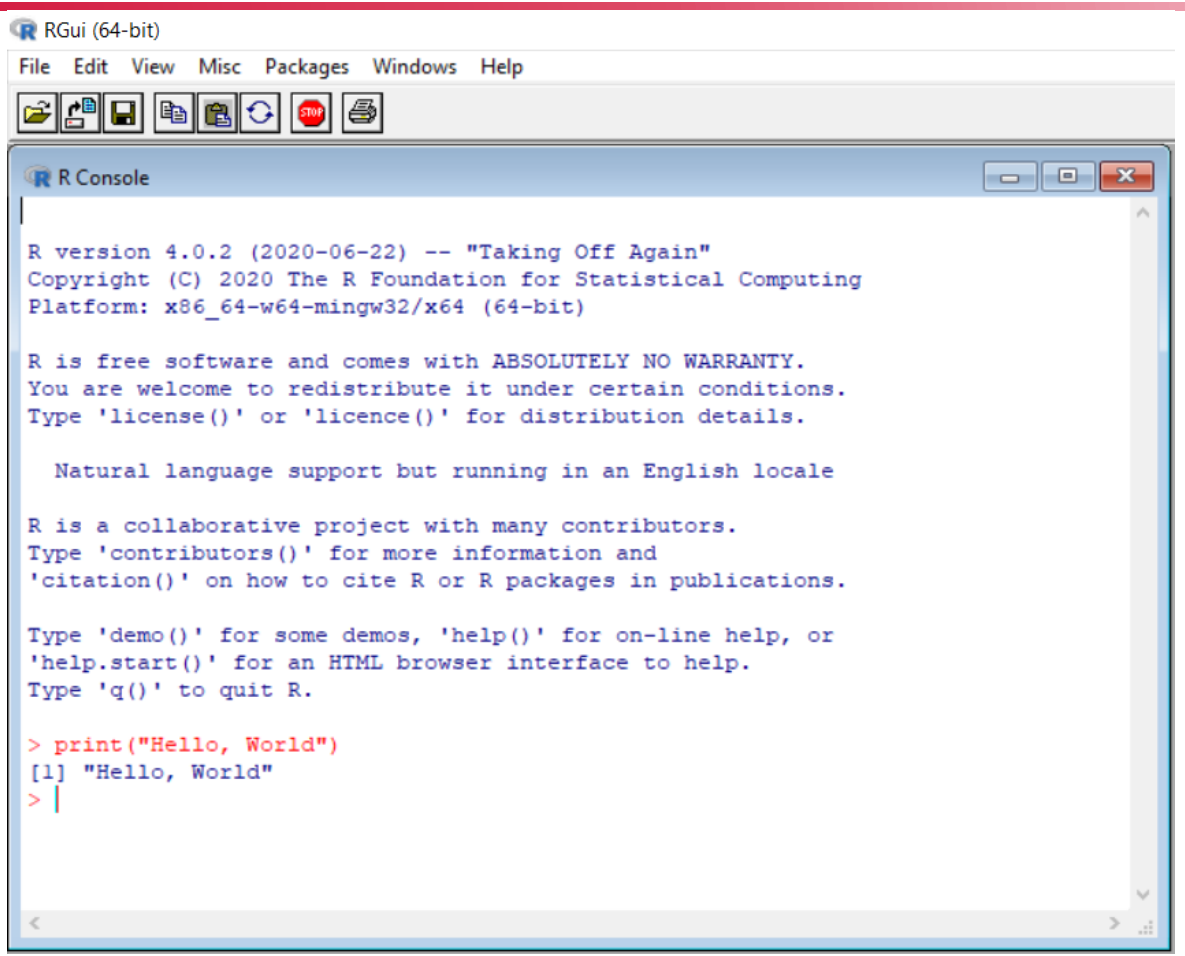
R官方地址：https://www.r-project.org/

R下载地址：https://cran.r-project.org/mirrors.html

China	
https://mirrors.tuna.tsinghua.edu.cn/CRAN/	TUNA Team, Tsinghua University
https://mirrors.bfsu.edu.cn/CRAN/	Beijing Foreign Studies University
https://mirrors.ustc.edu.cn/CRAN/	University of Science and Technology of China
https://mirror-hk.koddos.net/CRAN/	KoDDoS in Hong Kong
https://mirrors.e-ducation.cn/CRAN/	Elite Education
https://mirror.lzu.edu.cn/CRAN/	Lanzhou University Open Source Society
https://mirrors.nju.edu.cn/CRAN/	eScience Center, Nanjing University
https://mirrors.tongji.edu.cn/CRAN/	Tongji University
https://mirrors.sjtug.sjtu.edu.cn/cran/	Shanghai Jiao Tong University
https://mirrors.sustech.edu.cn/CRAN/	Southern University of Science and Technology (SUSTech)

4. R&RStudio介绍

R运行界面



4. R&RStudio介绍

RStudio是R的集成开发环境(IDE)：界面丰富，使用方便。

https://rstudio.com/products/rstudio/download/#download

RStudio Desktop 1.4.1106 - Release Notes

1. Install R. RStudio requires R 3.0.1+.

2. Download RStudio Desktop. Recommended for your system:

DOWNLOAD RSTUDIO FOR WINDOWS
1.4.1106 | 155.97MB

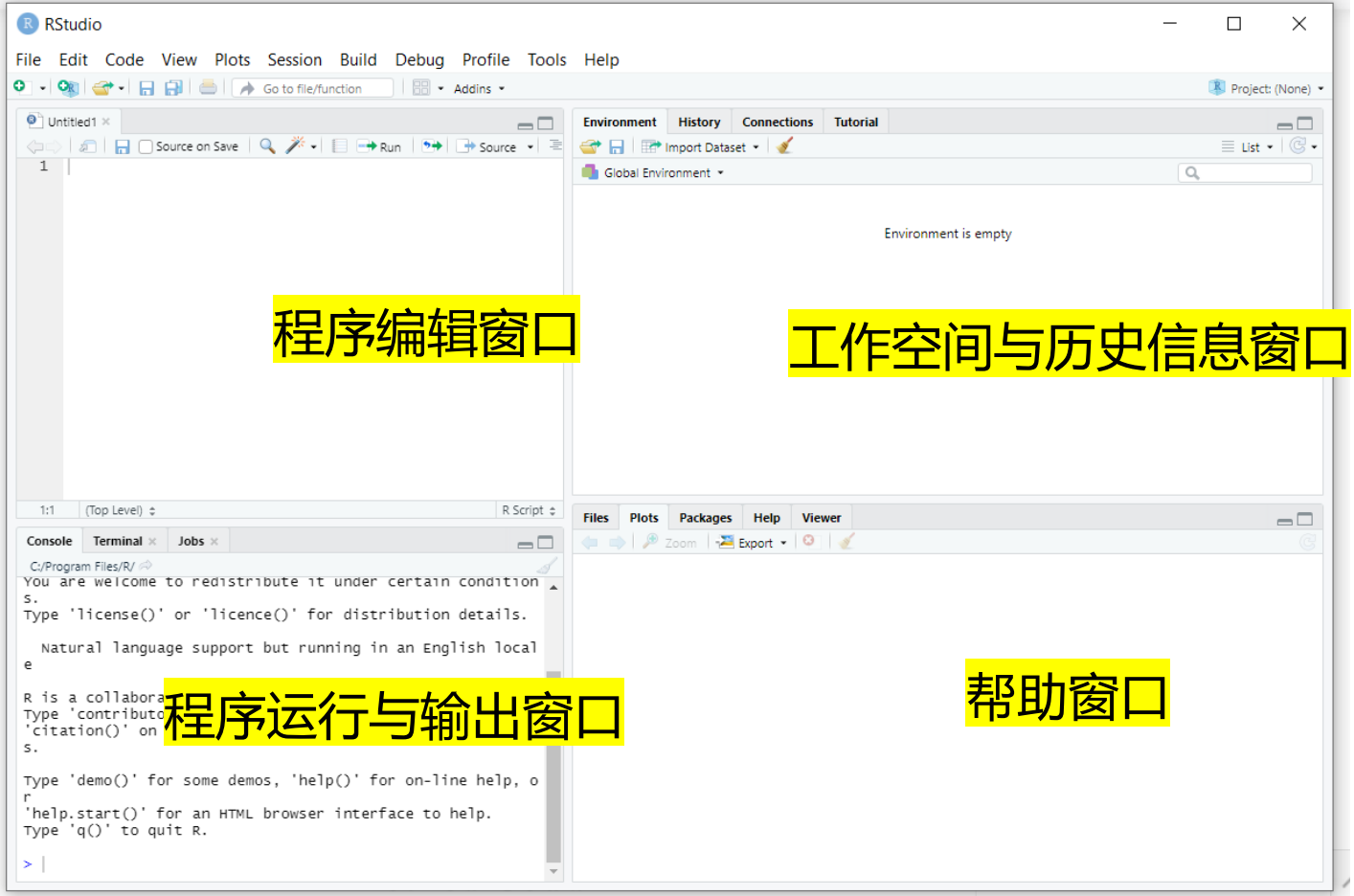
Requires Windows 10/8/7 (64-bit)

All Installers

4. R&RStudio介绍

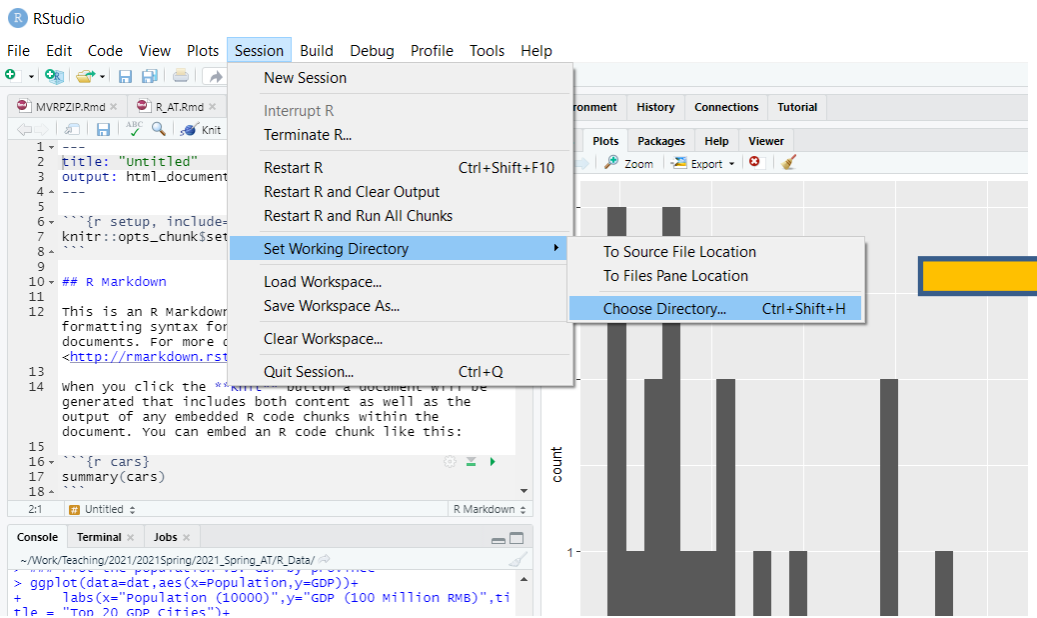
RStudio

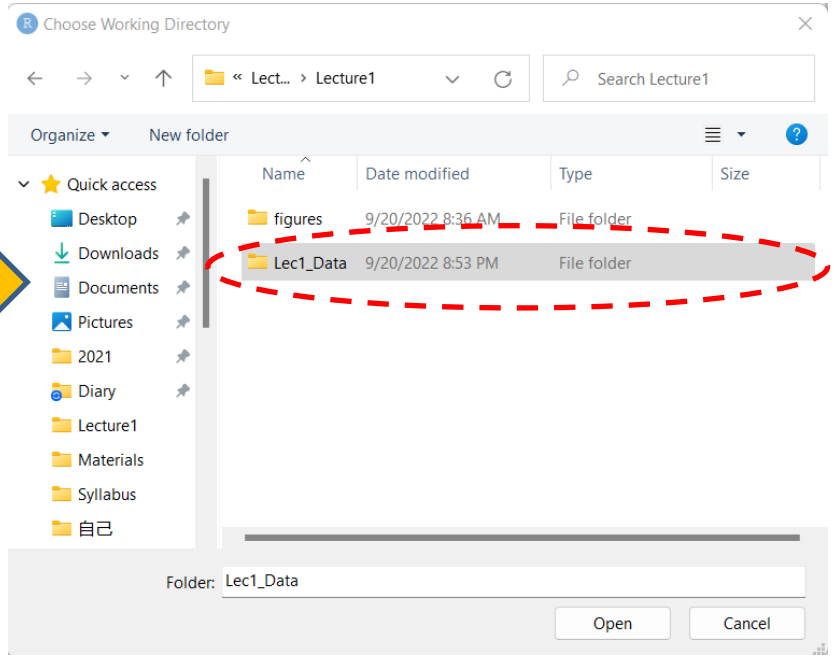
运行界面



4. R&RStudio介绍

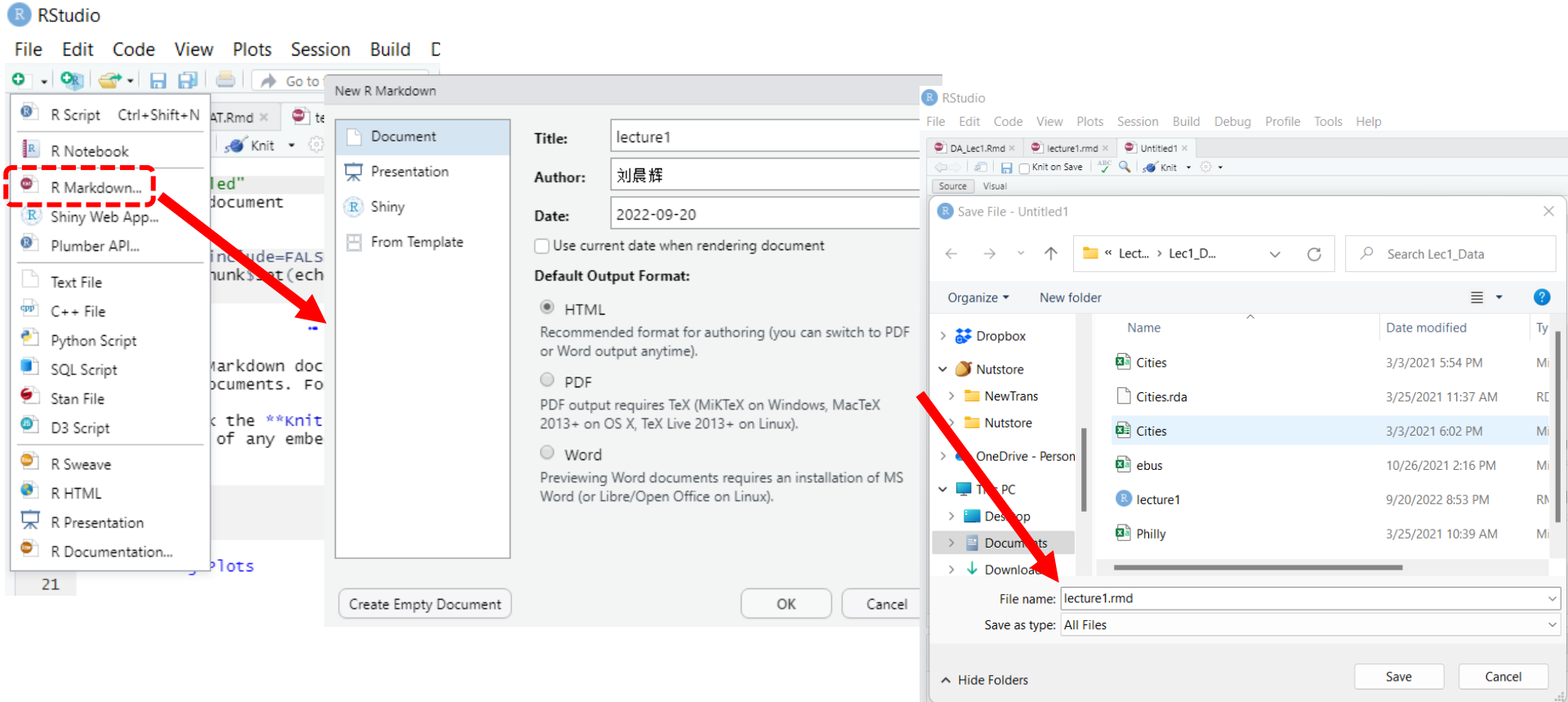
Step1: 设置工作目录 – 将工作目录设置为Lec1_Data文件夹





4. R&RStudio介绍

Step2: 创建R Markdown文件



5. R基本知识

5.1 常用操作 (Common Operations)

命令 (Command)	功能 (Function)	命令 (Command)	功能 (Function)	命令 (Command)	功能 (Function)
print()	显示内容	data()	上载数据	=/<-	赋值
help()/?	帮助工具	load()	上载文件	==	逻辑等于
getwd()	目前工作目录	save()	保存文件	! =	逻辑不等
setwd()	目前工作目录	format()	格式化文件	>	大于
list.files()	列出当前目录所有内容	rm()	删除文件	<	小于
install.packages()	安装工具包	class()	数据类型	!x	逻辑非
library()	上载工具包	ls()	列出环境所有变量	X & Y	逻辑与

5. R基本知识

5.2 变量类型 (Variables)

- 数字(numeric): 数字
- 文本(character): 用单(双)引号包含
- 逻辑(TRUE/FALSE): 区分大小写
- 因子(factor): 分类数据，levels。

常用操作： class()，判断变量类型。

变量命名规则

- 由英文字母、数字、英文连接符(.,_)组成
- 英文字母区分大小写：ab与AB不同
- 名称必须以英文字母为头
- 特殊字母不可使用: TRUE, FALSE, T, F, if, else, etc.

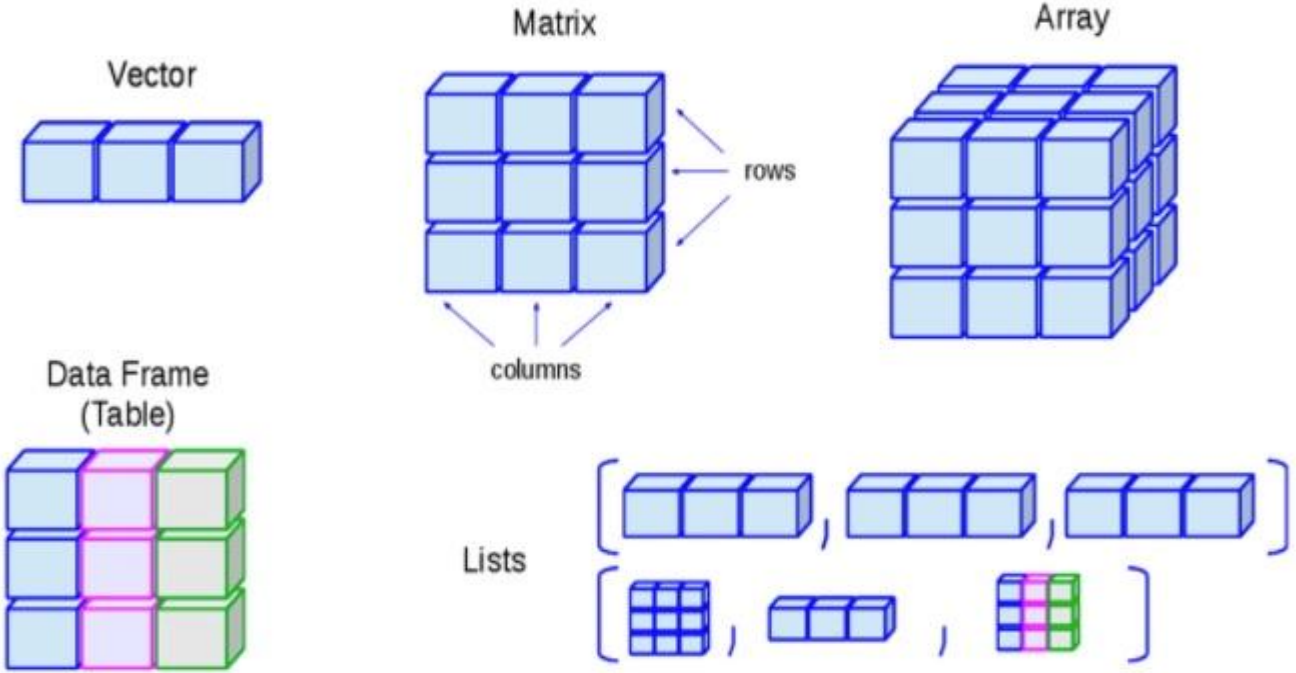
```
> a=1
> print(a)
[1] 1
>
> b<- "HNU"
> print(b)
[1] "HNU"
>
> c=TRUE
> print(c)
[1] TRUE
>
> d<- factor(c("HNU"))
> print(d)
[1] HNU
Levels: HNU
```

```
> class(a)
[1] "numeric"
```

5. R基本知识

5.3 数据结构类型 (Data Structure)

- 向量(vector)
- 矩阵(matrix)
- 数组(array)
- 数据框(data frame)
- 列表(list)



5. R基本知识

5.3 数据结构类型 (Data Structure)

向量(vector): 由元素(element)组成的一维数组

创造向量

```
> ### Vector
> a = 1
> print(a[1])
[1] 1
>
> b = c(1,10,100)
> print(b[1])
[1] 1
> print(b[1:3])
[1]   1  10 100
> print(b[c(1,3)])
[1]   1 100
```

向量操作

```
> ### Vector Operation
> b = c(1,100,10)
> print(class(b))
[1] "numeric"
> print(length(b))
[1] 3
> print(sort(b))
[1]   1  10 100
> print(sort(b,decreasing = TRUE))
[1] 100  10   1
> print(order(b))
[1] 1 3 2
> print(b[order(b)])
[1]   1  10 100
```

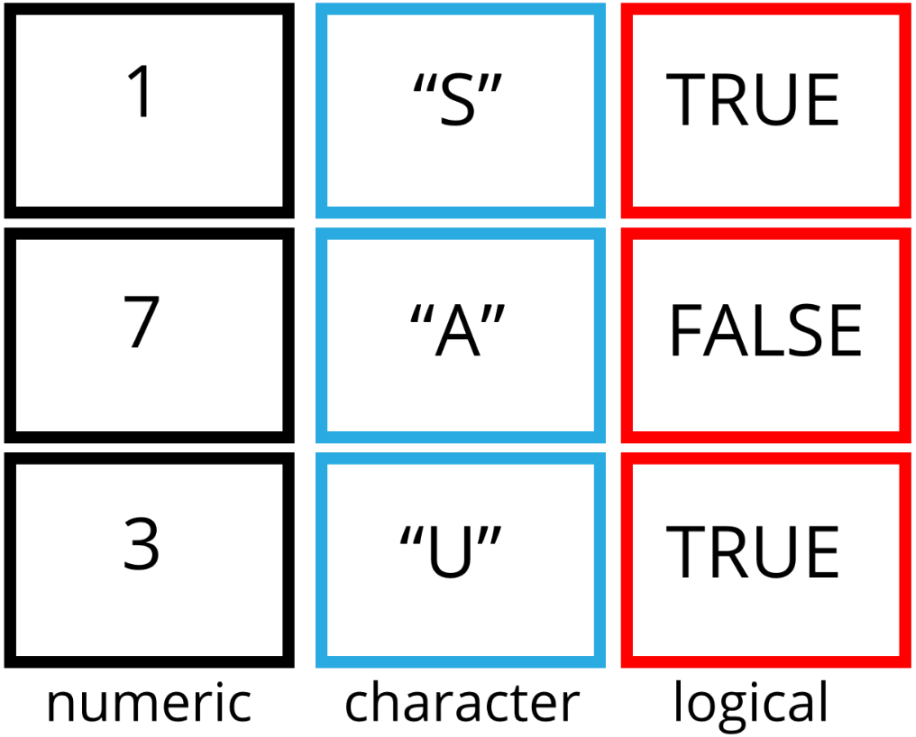
5. R基本知识

5.3 数据结构类型 (Data Structure)

data frame

数据框(data.frame)：

多维vector



5. R基本知识

5.3 数据结构类型 (Data Structure)

数据框：

由列创造

```
> ### data.frame
> city<- c("Changsha","Wuhan")
> province<- c("Hunan","Hubei")
> population<- c(839.45,1121.2)
> gdp<- c(12142.52,15616.1)
>
> dat<- data.frame(cbind(city,province,population,gdp))
>
> print(dat)
      city province population      gdp
1 Changsha    Hunan     839.45 12142.52
2    Wuhan    Hubei     1121.2  15616.1
> print(colnames(dat))
[1] "city"       "province"   "population" "gdp"
> print(rownames(dat))
[1] "1" "2"
> print(nrow(dat))
[1] 2
> print(ncol(dat))
[1] 4
```

5. R基本知识

5.3 数据结构类型 (Data Structure)

数据框：

由行创造

```
> ### data.frame: rbind
> city<- c("Changsha","Wuhan")
> province<- c("Hunan","Hubei")
> population<- c(839.45,1121.2)
> gdp<- c(12142.52,15616.1)
>
> record1<- c("Changsha","Hunan",839.45,12142.52)
> record2<- c("Wuhan","Hubei",1121.2,15616.1)
>
> dat<- data.frame(rbind(record1,record2))
> print(dat)
                X1     X2     X3       X4
record1 Changsha Hunan 839.45 12142.52
record2    Wuhan Hubei 1121.2  15616.1
>
> colnames(dat)<- c("city","province","population","gdp")
> print(dat)
            city province population      gdp
record1 Changsha    Hunan     839.45 12142.52
record2    Wuhan    Hubei     1121.2  15616.1
```

5. R基本知识

5.3 数据结构类型 (Data Structure)

数据框常用操作

```
> ### data.frame: operation
> print(class(dat))
[1] "data.frame"
> print(dim(dat))
[1] 2 4
> print(nrow(dat))
[1] 2
> print(ncol(dat))
[1] 4
> print(str(dat))
'data.frame':   2 obs. of  4 variables:
 $ city      : chr  "Changsha" "Wuhan"
 $ province  : chr  "Hunan" "Hubei"
 $ population: chr  "839.45" "1121.2"
 $ gdp       : chr  "12142.52" "15616.1"
NULL
> print(names(dat))
[1] "city"       "province"   "population" "gdp"
> print(dat[1,1])
[1] "Changsha"
```

- 查看第一行的元素：dat[1,]
- 查看第一列的元素：dat[,1]
- 查看第一行，第一列的元素：dat[1,1]

5. R基本知识

5.4 数据输入输出 (Data Input & Output)

R支持批量的从主流的表格存储格式文件（例如CSV、XLSX/XLS、XML等）中输入数据：

- CSV文件：逗号分隔值（comma-separated values），以纯文本形式存储表格数据的简单文件格式。
- XLSX/XLS文件：二进制的文件。
- .Rdata
- XML
- …

5. R基本知识

5.4 数据输入输出 (Data Input & Output)

（1）CSV文件

- 本质为文本文件：行为记录，列为字段。
- 适合存储中小型数据

文本打开

EXCEL打开

	A	B	C	D	E
1	Rank	City	Province	GDP	Population
2	1	Shanghai	Shanghai	38700.58	2428.14
3	2	Beijing	Beijing	36102.6	2153.6
4	3	Shenzhen	Guangdong	27670.24	1343.88

5.4 数据输入输出 (Data Input & Output)

(1) CSV文件：直接读入，无需任何工具包。

```
> ### Import csv file
> dat<- read.csv("Cities.csv")
> print(dat)
   Rank         City     Province       GDP Population
1     1     Shanghai     Shanghai 38700.58    2428.14
2     2      Beijing      Beijing 36102.60    2153.60
3     3     Shenzhen    Guangdong 27670.24    1343.88
```

```
> ### csv data process
> dat<- read.csv("Cities.csv")
> print(class(dat)) #查看数据类型
[1] "data.frame"
> print(ncol(dat)) #列数
[1] 5
> print(nrow(dat)) #行数
[1] 50
```

CSV文件输入后，数据类型为数据框。

5. R基本知识

5.4 数据输入输出 (Data Input & Output)

（1）CSV文件：直接读出，无需任何工具包。

```
> #### Export CSV file
> write.csv(dat,file="Cities_New1.CSV")
> write.csv(dat,file="Cities_New2.CSV",row.names = FALSE)
> dat_new1<- read.csv("Cities_New1.CSV")
> dat_new2<- read.csv("Cities_New2.CSV")
> print(head(dat_new1,3))
  X Rank     City  Province      GDP Population
1 1    1 Shanghai  Shanghai 38700.58    2428.14
2 2    2  Beijing   Beijing 36102.60    2153.60
3 3    3 Shenzhen Guangdong 27670.24    1343.88
> print(head(dat_new2,3))
  Rank     City  Province      GDP Population
1    1 Shanghai  Shanghai 38700.58    2428.14
2    2  Beijing   Beijing 36102.60    2153.60
3    3 Shenzhen Guangdong 27670.24    1343.88
```

注意行名字的设定！

5.4 数据输入输出 (Data Input & Output)

(2) XLSL文件

需要工具包”readxl”,”writexl”。

- install.packages(“readxl”)
- install.packages(“writexl”)

```
library(readxl)
library(writexl)

### Import xlsx file
dat<- readxl::read_xlsx(path = "Cities.xlsx",sheet = 1)
print(head(dat,3))

### Output xlsx file
writexl::write_xlsx(dat,path = "Cities_New.xlsx")
```

5.4 数据输入输出 (Data Input & Output)

（3）Rdata文件：R自带的数据存储格式，load & save

```
> load("Cities.rda")
>
> print(head(Cities))
  Rank      City  Province      GDP Population
1    1  Shanghai  Shanghai 38700.58    2428.14
2    2   Beijing   Beijing 36102.60    2153.60
3    3  Shenzhen Guangdong 27670.24    1343.88
4    4 Guangzhou Guangdong 25019.11    1530.59
5    5 Chongqing Chongqing 25002.79    3124.32
6    6    Suzhou   Jiangsu 20170.50    1074.99
>
> Cities_New<- Cities
> save(Cities_New,file = "Cities_New.rda")
```

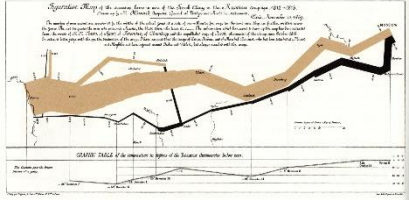
5. R基本知识

5.5 数据可视化 (Data Visualization)

国家综合交通
立体网络规划
(2021)

5. R基本知识

5.5 数据可视化 (D



Based on Charles Minard's graph of Napoleon's Russian campaign of 1812.

拿破仑1812侵俄战争
Source: Charles-Joseph Minard

5. R基本知识

5.5 数据可视化 (Data Visualization)

(1) R自带画图工具：**plot**, line, points, hist, pie, …

Usage

```
plot(x, y, ...)
```

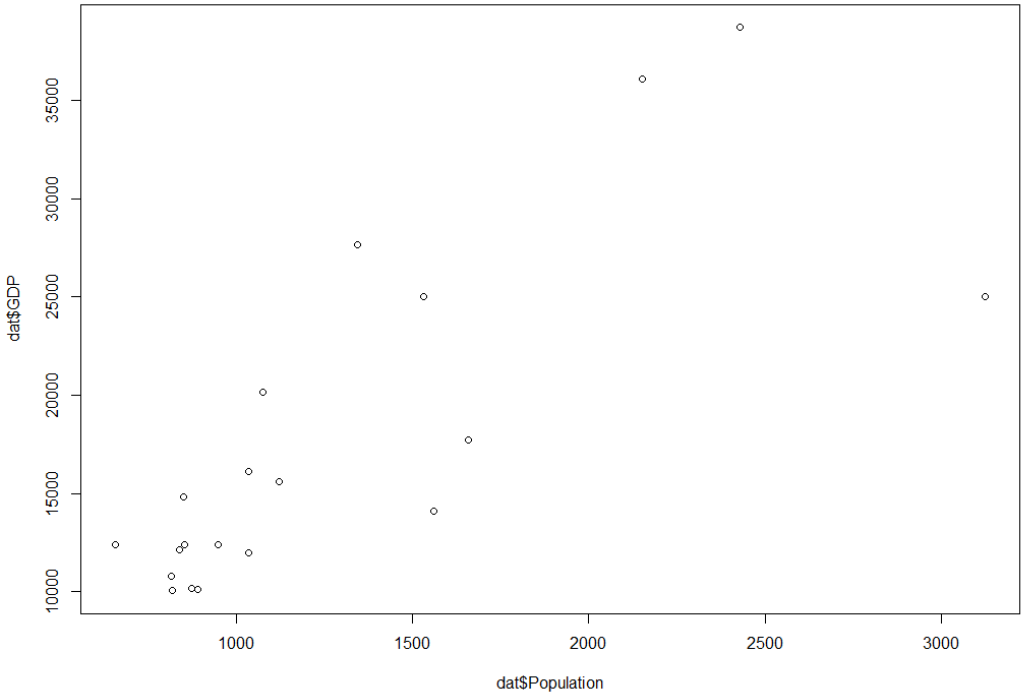
Arguments

`x`	the coordinates of points in the plot. Alternatively, a single plotting structure, fun
`y`	the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.
`...`	Arguments to be passed to methods, such as graphical parameters (see `par`).

5. R基本知识

5.5 数据可视化 (Data Visualization)

（1） R自带画图工具：**plot**



人口 vs. GDP散点图

```
> ### Basic plot functions
> dat<- read.csv("Cities.csv")
> plot(dat$Population,dat$GDP)
```

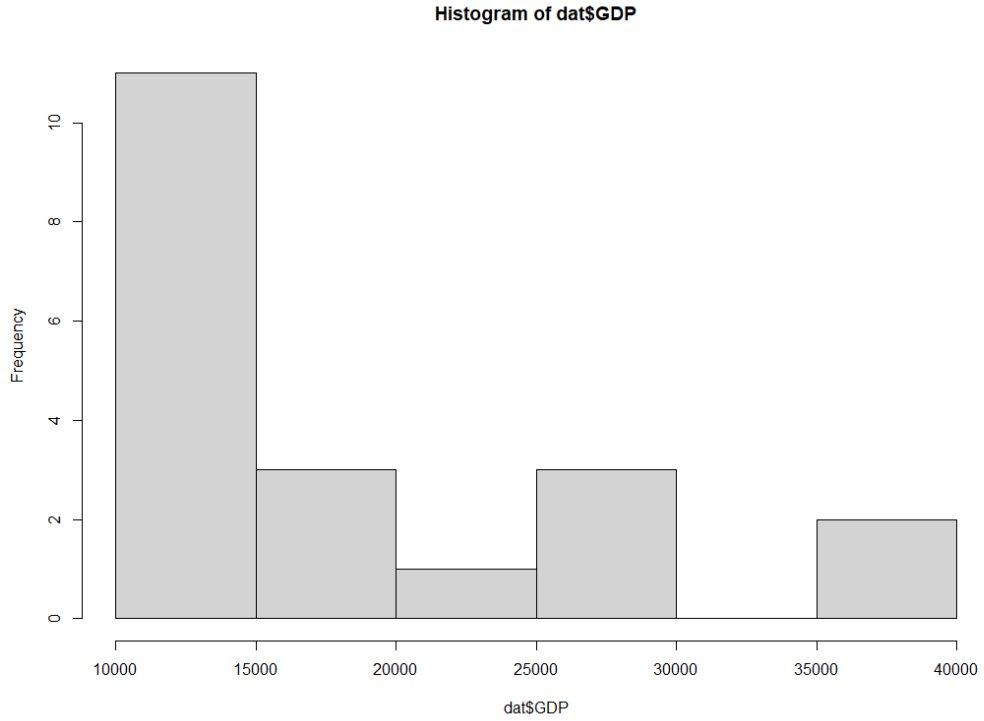
5. R基本知识

5.5 数据可视化 (Data Visualization)

（1） R自带画图工具：**hist**

hist: 直方图

```
> ### Histogram
> hist(dat$GDP)
```



5. R基本知识

5.5 数据可视化 (Data Visualization)

(2) ggplot工具

ggplot2: R中最常用的画图工具包。

- install.packages("ggplot2")
- library(ggplot2)

`ggplot()` is used to construct the initial plot object, and is almost always followed by + to add component to the plot.

- `ggplot(df, aes(x, y, other aesthetics))`
- `ggplot(df)`
- `ggplot()`

5. R基本知识

5.5 数据可视化 (Data Visualization)

(2) ggplot工具

Hadley Wickham

职位：
Chief Scientist in RStudio, Inc.

出生：
1979/10/14
Hamilton, New Zealand

博士毕业学校：
Iowa State University

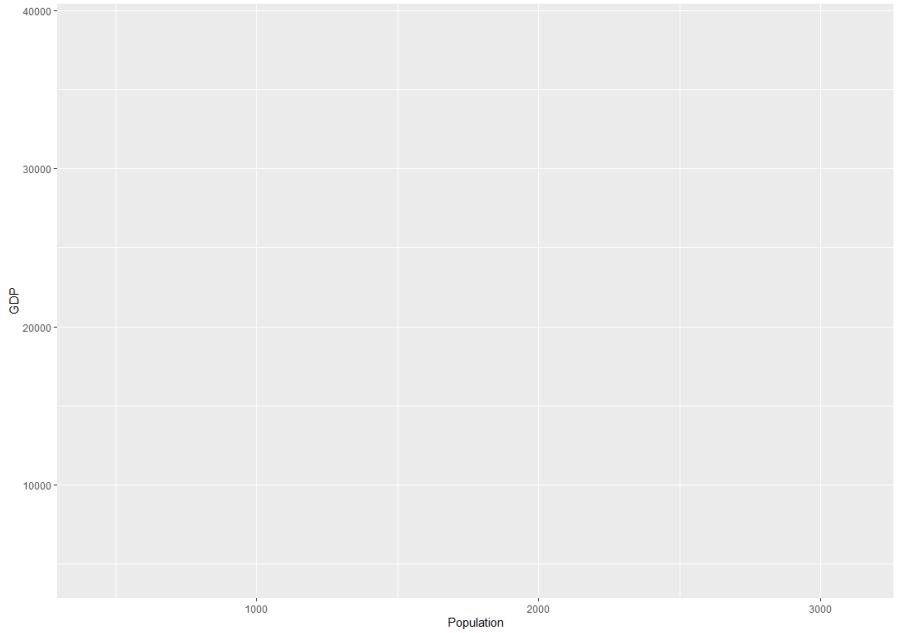
博士生导师：
Di Cook & Heike Hofmann

奖项：
2019 COPSS Presidents' Award

5.5 数据可视化 (Data Visualization)

(2) ggplot工具

```
> ### Plot the population vs. GDP
> ggplot(data=dat,aes(x=Population,y=GDP))
```



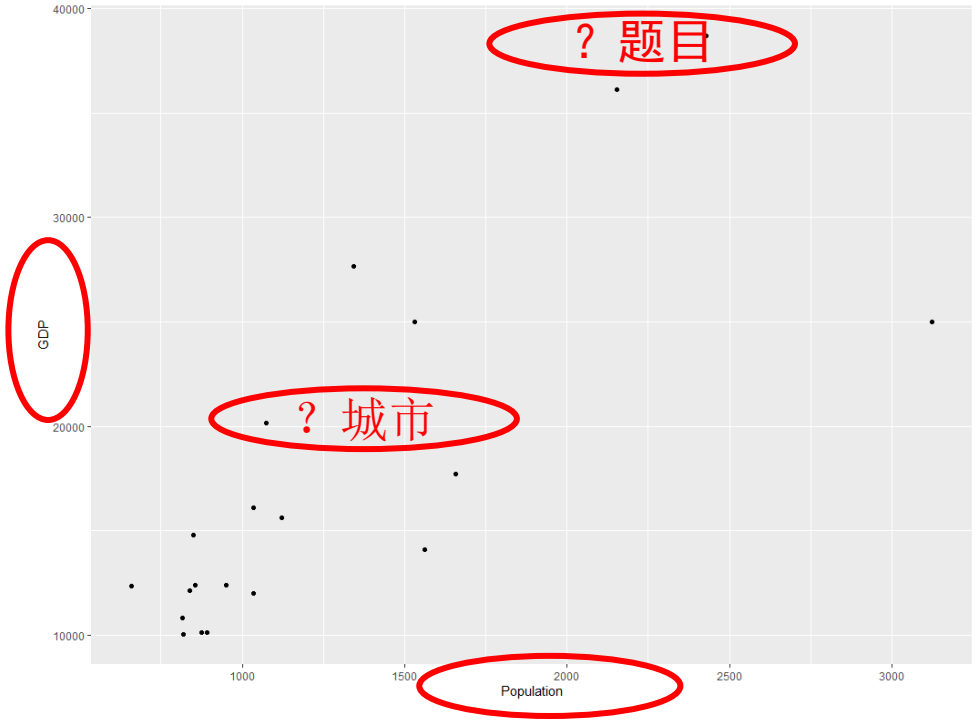
人口 vs. GDP散点图

5.5 数据可视化 (Data Visualization)

（2）ggplot工具

```
> ### Plot the population vs. GDP
> ggplot(data=dat,aes(x=Population,y=GDP))+
+   geom_point()
```

需要明确说明图像形式！

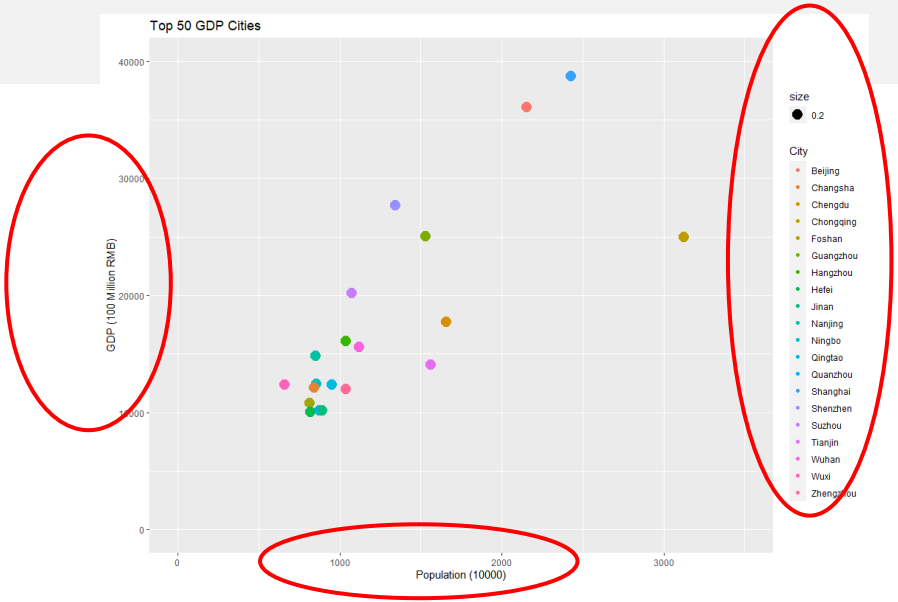


5. R基本知识

5.5 数据可视化 (Data Visualization)

（2）ggplot工具

```
### Plot the population vs. GDP, more features
ggplot(data=dat,aes(x=Population,y=GDP))+
  labs(x="Population (10000)",y="GDP (100 Million RMB)",title = "Top 20 GDP Cities")+
  geom_point(aes(colour=City,size=0.2))+
  xlim(c(0,3500))+
  ylim(c(0,40000))
```



5. R基本知识

5.6 数据基本运算操作 (Operations)

1.求和(sum): sum(dat$GDP)

2.最大值(Max): max(dat$GDP)

3.最小值(Min): min(dat$GDP)

4.中位值(Median): median(dat$GDP)

5.百分位值(Percentile): quantile(dat$GDP)

6.平均值(Average/Mean): mean(dat$GDP)

7.方差(Variance): var(dat$GDP)

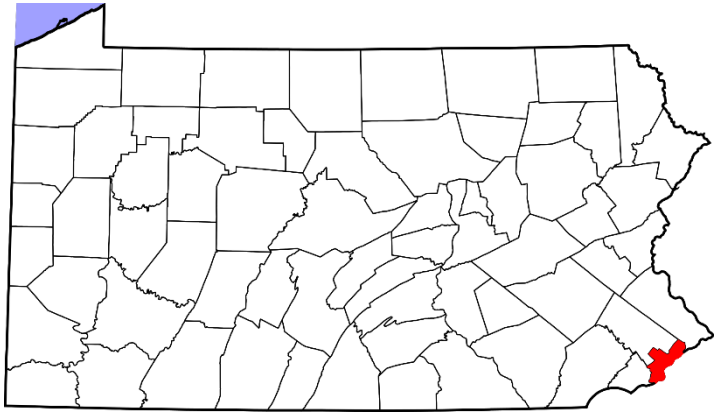
8.标准差(Standard Deviation): sd(dat$GDP)

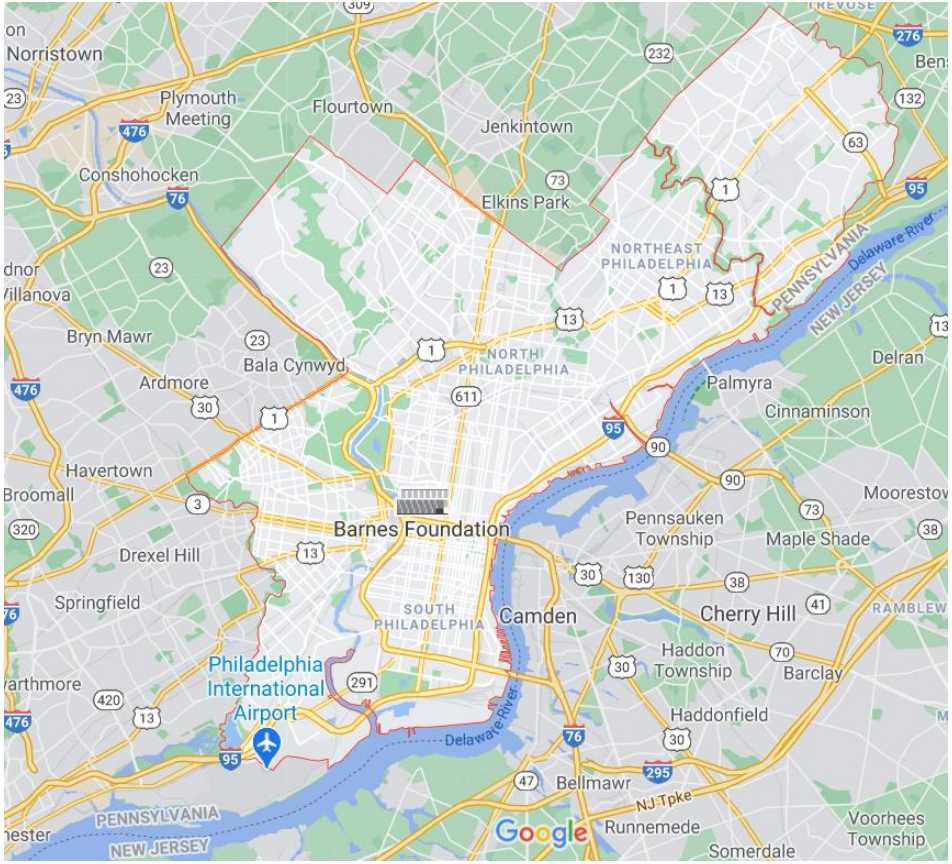
9.汇总: summary(dat$GDP)

```
> ### Basic Functions
> print(summary(dat$GDP))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   4546    5940    7811   10979   12393   38701
```

6. 实例分析 – 费城交通事故分析

费城(Philadelphia): 157.9万人口(2019), 367平方公里





6. 实例分析 – 费城交通事故分析

费城交通事故记录（2008-2017）

➢一共有多少交通事故发生？造成多少死亡？多少受伤？

```
#################### Example ############################
philly<- read.csv("Philly.csv")
### How many crashes occur? How many injuries and deaths?
print(nrow(philly))
# 85712
print(sum(philly$Death))
# 799
print(sum(philly$Injury))
# 93646
```

6. 实例分析 – 费城交通事故分析

➢事故组成：交叉口，天气，与碰撞类型

```
### Crash composition #########
### Intersection
print(table(philly$Intersection))
print(prop.table(table(philly$Intersection)))
#     0     1
# 36555 49155

### Weather
print(table(philly$Weather))
print(prop.table(table(philly$Weather)))
#  Good  Rain  Snow
# 71416 12606  1690

### Collision
print(table(philly$Collision))
print(prop.table(table(philly$Collision)))
# Angle Hit_fixed_object   Hit_pedestrian         Rear_end
# 33542            12708            14926            24536
```

6. 实例分析 – 费城交通事故分析

- 发生的时间分布：年份，月份，分别以表格和曲线形式画出来？

```
> print(table(philly$Year,philly$Month))

      1   2   3   4   5   6   7   8   9  10  11  12
2008 620 602 690 777 777 730 666 472 669 502 661 652
2009 546 570 640 732 805 735 689 519 683 782 695 638
2010 611 456 712 774 836 794 757 751 756 834 735 592
2011 519 542 662 740 795 758 688 743 684 762 709 763
2012 679 674 827 832 871 815 704 663 696 756 630 681
2013 618 596 761 789 870 781 712 707 775 678 713 623
2014 554 440 660 696 680 703 680 714 725 781 730 733
2015 593 582 645 737 807 861 806 788 793 857 796 830
2016 729 727 777 857 892 842 761 794 786 832 853 724
2017 656 626 707 776 880 704 476 788 717 805 810 726
```

6. 实例分析 – 费城交通事故分析

```
t1<- as.data.frame(table(philly$Year,philly$Month))
names(t1)<- c("Year","Month","Freq")
t1$Month<- as.numeric(as.character(t1$Month))

g1<- ggplot(data = t1)+
  geom_line(aes(x=Month,y=Freq,color=Year,linetype=Year))

print(g1)
```

谢 谢!