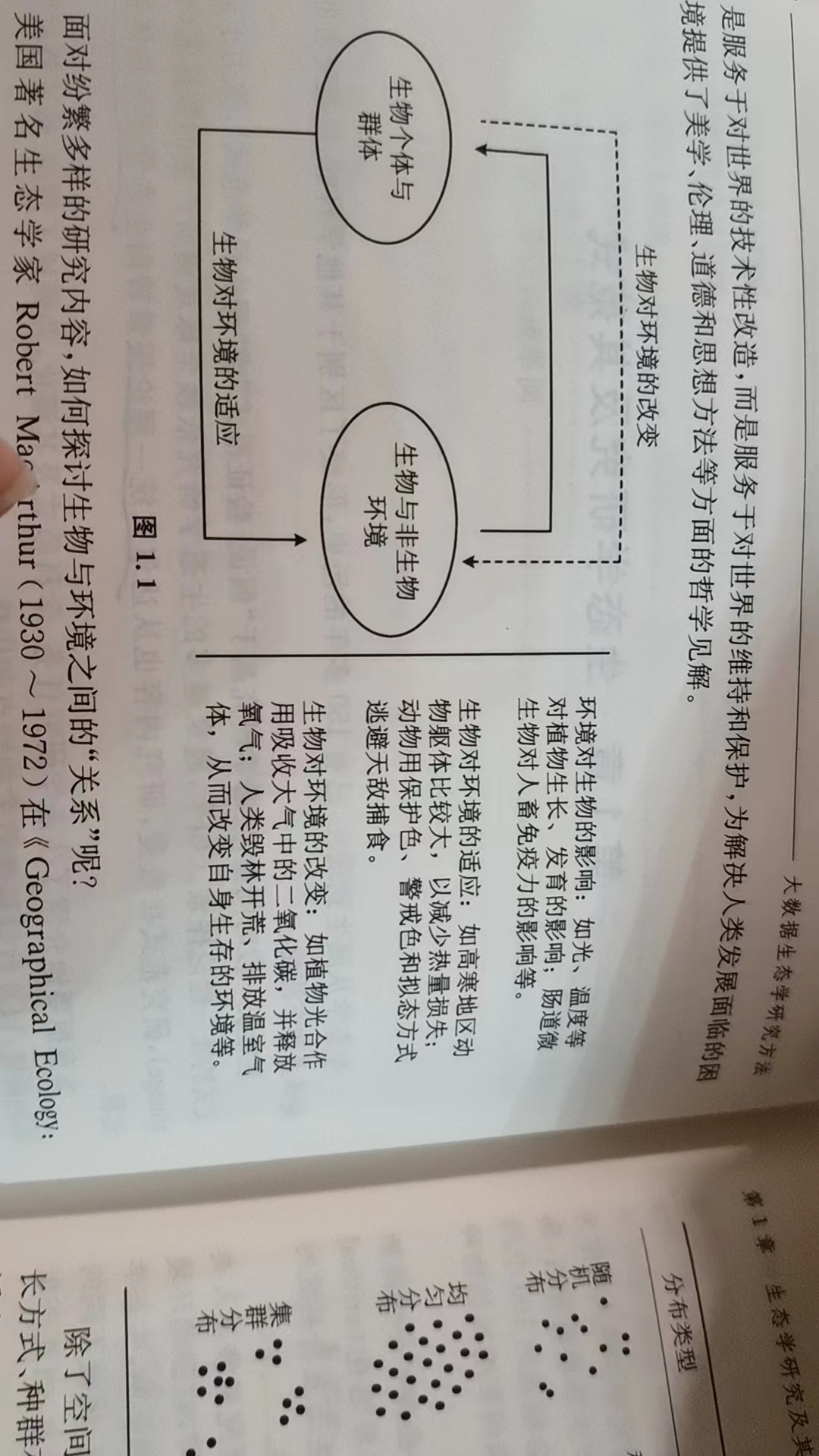
1. 生态学研究及其范式

1.生态学从诞生到现在已有150多年的历史，形成了区别于其他学科的，比较完善的理论体系和方法论。在很长一段时间里，生态学研究基手"假说证式”或“那论科学范式”，致力于达尔文式的统一理论探索。如今越来越多的生态学研究依赖全球规测所产生的数据流，研究范式在改变，研究内从达尔文统一论探索转向生态管理和具体生态过程。

生物和环境之间的关系：

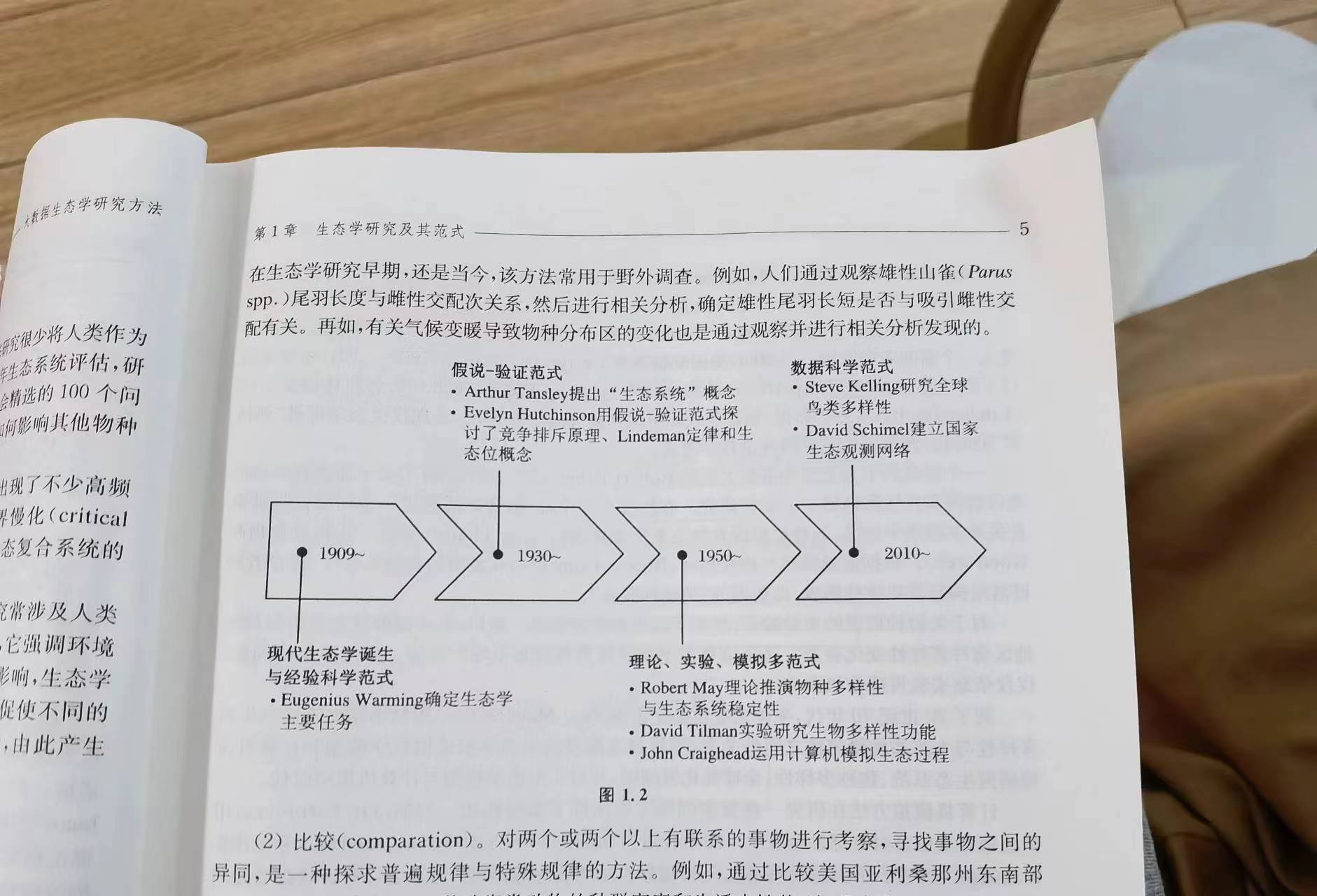


2.什么是“范式”

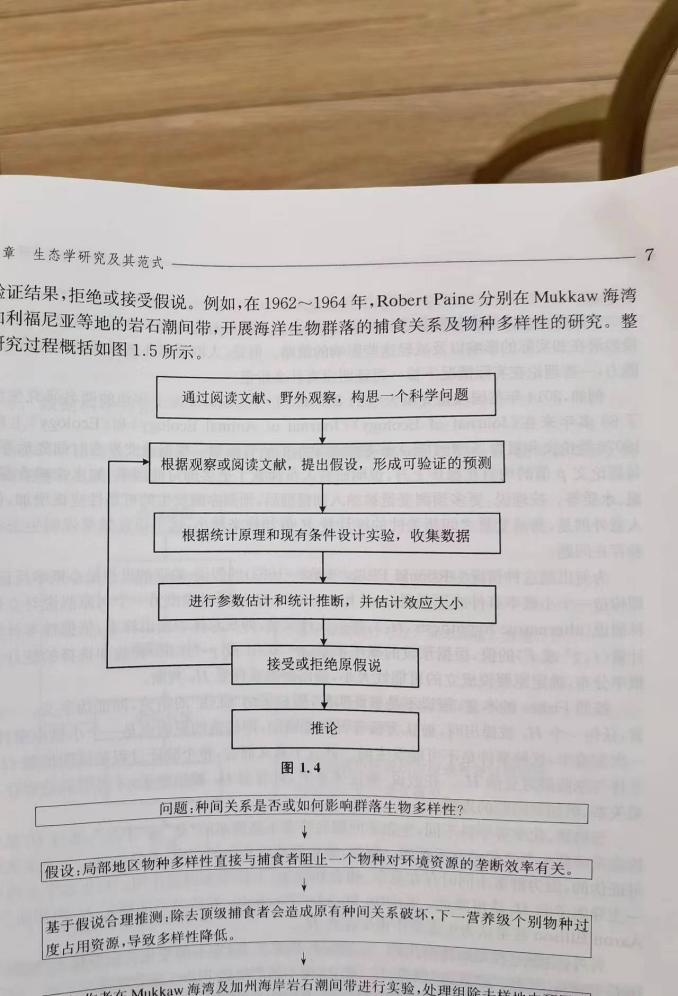
范式(paradigm)是一个哲学术语,来自ThomasKuhn(1922~1996)的著作《TheStructure of Scientific Revolutions》。通俗地讲，它是一个学科的理论基础和实践规范的全部，包括定律、理论体系以及研究的仪器设备和具体方法，是从事某个学科研究成员公认的模式。

研究范式是学科范式的方法论部分。按照JimGray的观点，科学研究范式可分为四类/以观察和实验描述自然规律为主的经验科学，用模型推演得到结论的理论科学，利用电子计算机对科学实验进行模拟仿真的计算科学，以及直接分析海量数据发现相关关系而获得新知识的数据密集型科学。

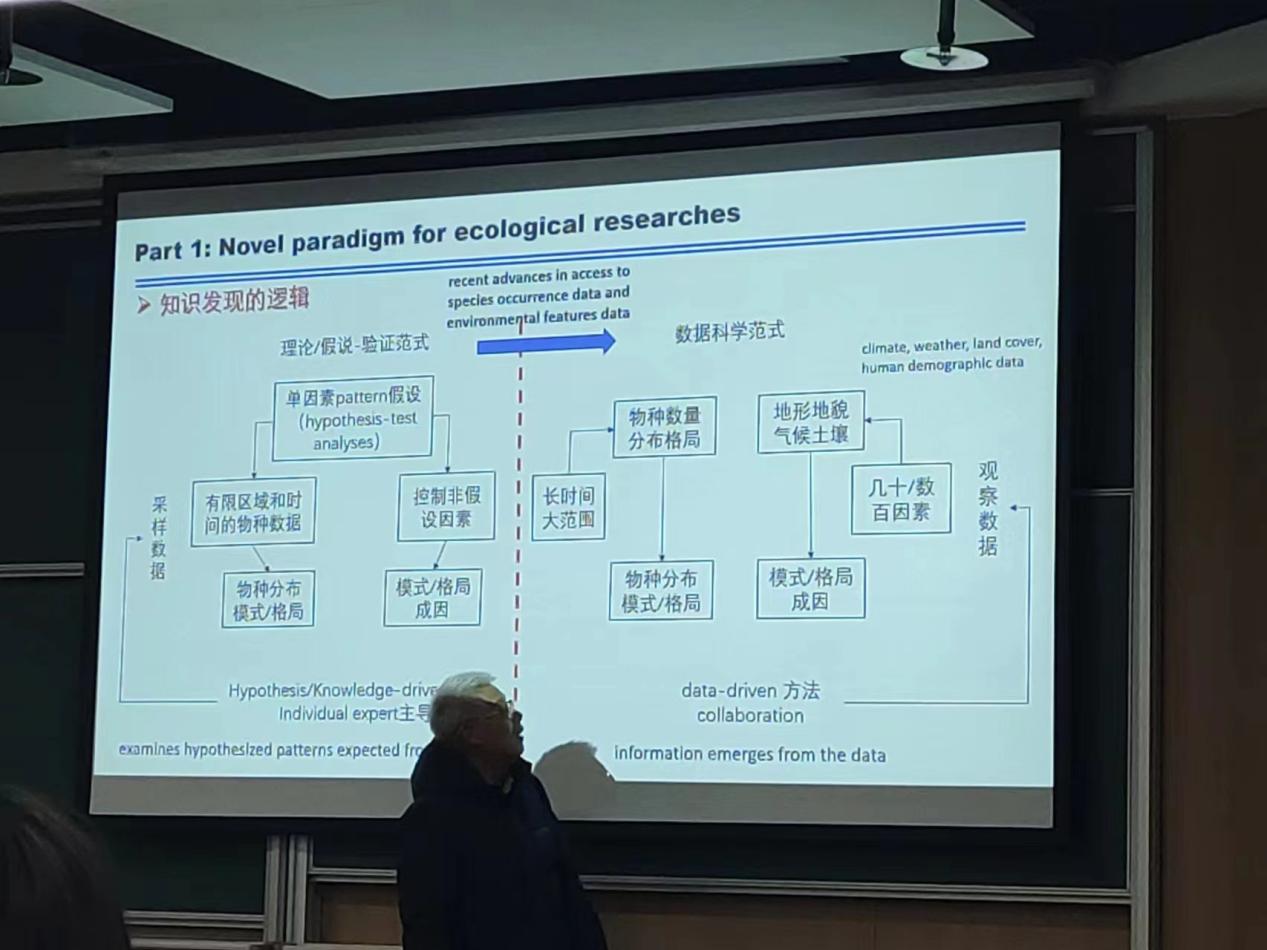
发展历程：



3.假说-演绎科学研究模型：



4.从理论-验证范式到数据科学范式的转变



5.相关研究案例：

1.Steve kelling及其同事:北美鸟类生物多样性及分布

https://dol.org/10.1525/bio 2009 59.7 12

①获取多源异构数据②数据集成与再利用③探索性数据分析④验证假说真实性

6.数据科学范式的优势:

①相关性分析②全数据归纳:样本=总体，避免经验科学范式相关≠因果③利用非参数或半参数模型将数科学范式与理论科学范式融合，避免归纳偏见

第二章R语言与数据挖掘

1.安装R语言

Install R：Widows:https://cran.r-project org/bin/windows/base/

Install Rstudio：<https://positco/products/open-source/rstudio/download>

Install Rtools：<http://cran.r-project>

Install R and RStudio on ubuntu：

使用VMware，用户可在单一桌面上同时运行不同的操作系统，如Linux系统，部署不同的应用程序。

①下载vMware workstarion，网上资源比较多，最好是完整版的，可以从如下链接下载16.0版本，文件形式如VMware\_162319376536 Setup.exe。

链接:hitps//pan.baldu.com/s/1G8WPFZyOhxzP]4zfeInLsQ，提取码:4h17参考网上资源安装Vmware，接着，安装时一般要榆入密钥或许可证号，VMware workstation16.0层本的序列号可以从网上搜索。

②VMwaro里安装Ubuntu22.04，关于Ubuntu的安装，网上资源比较多，可以参考这个网站链接:hitps//blog,csdn.net/shenliu128/article/details/127262743

视频:https//www.youtube.com/watch?v=Nhlh]FKmzpk

③安装RSrudio：

wget https://dawnload1.rstudio.org/electronjammy/amd64/rstudio-2023.06,1-524-amd64.deb sudo apt install .1/istudio -2023.061-524-amd64.deb

Eg：在window系统中安装较为简单，安装虚拟机较为困难，首先进行了虚拟机的安装，一切顺利完成，遗憾的是后续运行过程中经常出问题，就改为了window系统使用了。

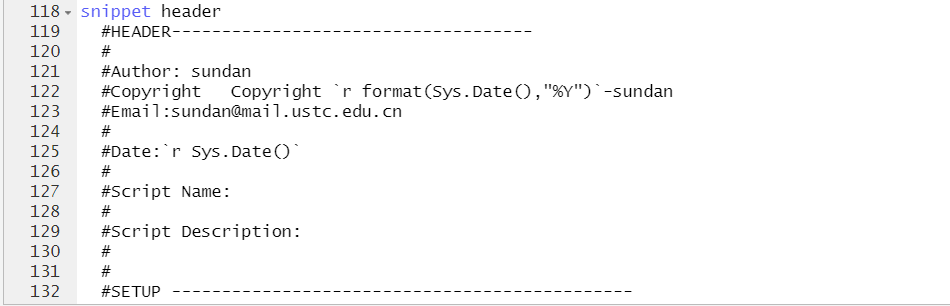
2.R语言安装包：

|  |
| --- |
| **加载数据** |
| DBI: 数据库接口，用于在R中连接和操作数据库。 |
| odbc: 提供了一个连接到ODBC（开放数据库连接）的统一接口，用于在R中访问各种数据库。 |
| RMySQL: 用于在R中连接和操作MySQL数据库的软件包。 |
| RPostgreSQL: 用于在R中连接和操作PostgreSQL数据库的软件包。 |
| RSQLite: 提供了在R中连接和操作SQLite数据库的功能。 |
| XLConnect: 用于在R中读取和写入Excel文件的软件包。 |
| xlsx: 提供了在R中读取和写入.xlsx格式文件的功能。 |
| foreign: 用于在R中读取和写入其他统计软件（如SPSS、SAS、Stata）的文件格式的软件包。 |
| haven: 提供了在R中读取和写入SPSS、SAS和Stata文件的功能。 |
| **数据处理** |
| tidyverse: 一个包括了多个数据处理和可视化包的集合，其中包括了 dplyr、tidyr 等等。 |
| plyr: 用于数据操作和变换的软件包。 |
| stringr: 提供了用于处理字符串的函数。 |
| lubridate: 用于处理日期和时间数据的软件包。 |
| 数据可视化： |
| ggplot2: 提供了用于制作高度定制化图形的功能。 |
| ggvis: 提供了交互式数据可视化功能。 |
| rgl: 用于制作三维图形的软件包。 |
| htmlwidgets: 提供了将JavaScript交互式图形嵌入到R Markdown文档中的功能。 |
| googleVis: 提供了用于创建Google Charts交互式图形的功能。 |
| **数据建模** |
| car: 包含了一系列用于应用统计模型和计算统计量的函数。 |
| mgcv: 用于广义加性模型（GAM）的软件包。 |
| lme4/nlme: 用于线性混合效应模型（LMMs）和广义线性混合模型（GLMMs）的软件包。 |
| randomForest: 提供了随机森林算法的实现。 |
| multcomp: 用于多重比较和多重检验的软件包。 |
| vegan: 用于生态学和环境数据分析的软件包。 |
| glmnet: 提供了用于广义线性模型的Lasso和弹性网络的功能。 |
| survival: 用于生存分析的软件包。 |
| caret: 提供了分类和回归训练（classification and regression training）的功能。 |
| tidymodels: 一个集合了用于建模和预测的软件包，包括模型拟合、交叉验证等。 |
| **报告结果** |
| shiny: 用于创建交互式Web应用程序的软件包。 |
| R Markdown: 用于创建动态文档的软件包。 |
| xtable: 用于创建可以在LaTeX或HTML中呈现的表格的软件包。 |
| **空间数据处理** |
| sp: 提供了在R中处理空间数据（地理信息系统）的功能。 |
| maptools: 提供了用于读取和操作地图数据的功能。 |
| maps: 提供了世界地图和国家地图的细节信息。 |
| ggmap: 提供了在ggplot2中绘制Google地图的功能。 |
| **时间序列数据处理** |
| zoo: 提供了处理时间序列数据的功能。 |
| xts: 提供了用于处理时间序列数据的扩展数据结构。 |
| quantmod: 提供了用于金融时间序列分析的功能。 |
| **高性能R代码编写** |
| Rcpp: 用于在R中编写C++代码的软件包。 |
| data.table: 提供了用于快速处理大型数据集的功能。 |
| parallel: 提供了在R中进行并行计算的功能。 |
| **与网络交互** |
| XML: 提供了在R中处理XML数据的功能。 |
| jsonlite: 提供了在R中处理JSON数据的功能。 |
| httr: 提供了在R中进行HTTP请求的功能。 |
| **编写自己的R包** |
| devtools: 提供了用于创建、安装和加载R软件包的工具。 |
| testthat: 提供了用于编写和运行测试的功能。 |
| roxygen2: 提供了用于自动生成R包文档的工具。 |

安装R包主要参考CARN，在Rstudio中命令为insatll.package(“\*\*\*”)

3.snippet

创建自己的header



4.实现github和R的连接

Git installed: <https://nit-scm.com/downloads>

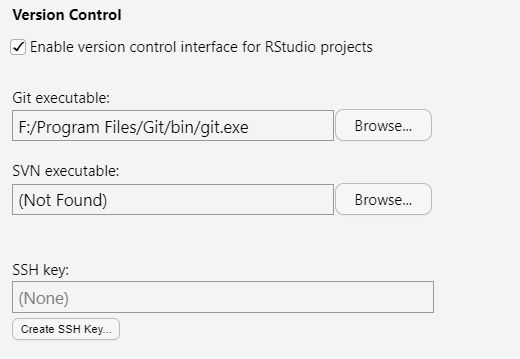
GitHub-sign up: httns://github com

①Open RStudio and go to Tools > Global Options click on Git/VIN

②Check Enable version control interface for RStudio projects

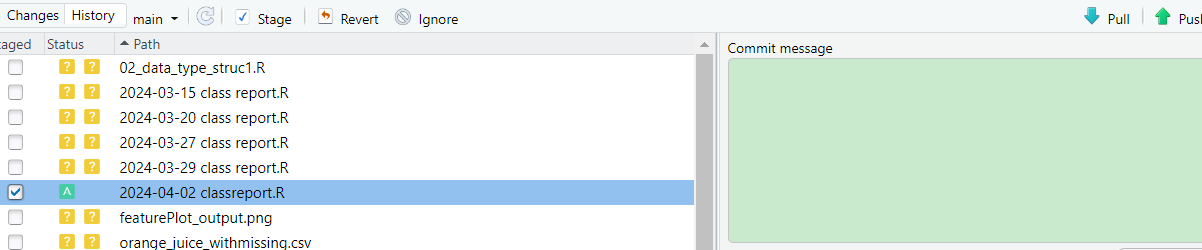
③Set the path to the Git executable that you just installed

④Restart Rstudio

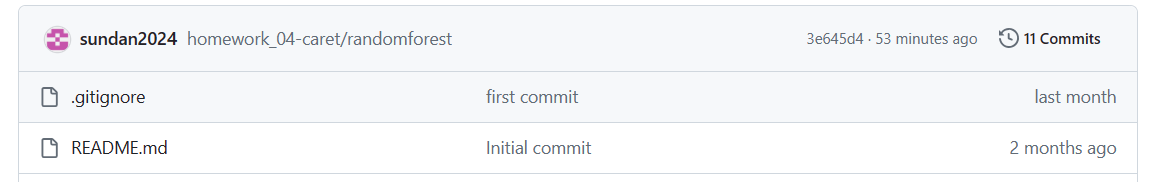


⑤Create a new project based on a remote Git repositon,: Select file > New Project. and from the opening menu select to create a new project from version control, choose Git, then provide the repository url (use the https link of the url if you want to avoid all the ssh trouble) from the repository you want to clone and create the project.

⑥在GitHub里面复制链接，一定记得选择readme！！



⑦实现commit与push



5.Rstudio四个区域：

**编辑器区域：**

这是您编写R代码和文档的主要区域。您可以在这里创建新的脚本文件、编辑现有文件，并在RStudio的编辑器中执行代码。编辑器区域通常包括代码高亮、自动补全、代码折叠等功能，以帮助您提高编码效率。

**控制台区域：**

控制台区域是您与R解释器进行交互的地方。您可以在控制台中执行单行代码、查看代码执行的结果，并与R进行交互式会话。控制台还提供了历史记录功能，可以查看之前执行过的命令。

**环境/历史区域：**

这个区域显示了当前R会话的环境变量以及您在控制台中执行的命令的历史记录。您可以查看当前定义的对象、数据框等，并通过点击它们来查看其内容。历史记录也可以用来重新执行之前执行过的命令。

**文件/目录/工具区域：**

这个区域包含了文件浏览器、Git版本控制、包管理器、作业管理等工具。您可以在这里浏览和管理项目文件、进行版本控制、安装和加载软件包、执行并行作业等

6.R语言的基础命令-整理大全

Library（）加载

input file <- "data/data.csv" 导入

output file <-"data/results.csv" 导出

sample\_number <-nrow(input\_data) 赋值

**①实际案例-计算符及基础画图**

01-Assignment Operators (赋值运算符):

<-：将右侧的值赋给左侧的变量。

->：与<-相反，将左侧的变量赋值为右侧的值。

=：也用于赋值操作，但通常在函数调用中使用。

02-Vector Operations (向量操作):

加法、减法、乘法、除法：对应位置上的元素进行相应的数学运算。

比较运算符（如<, >, ==, !=）：逐个比较两个向量中的元素，并返回逻辑值向量。

03-Arithmetic Operators (算术运算符):

+：加法。

-：减法。

\*：乘法。

/：除法。

^：指数运算。

%/%：整数除法（向下取整）。

%%：取余数。

04-Relational Operators (关系运算符):

<：小于。

>：大于。

<=：小于等于。

>=：大于等于。

==：等于。

!=：不等于。

05-Logical Operators (逻辑运算符):

!：逻辑非。

|：逻辑或。

&：逻辑与。

||：短路逻辑或。

&&：短路逻辑与。

06-plot() 函数:

plot()函数用于绘制散点图、折线图等，可以指定x轴和y轴的数据，以及其他参数来控制图形的外观。

07-设置标记属性:

pch参数用于设置标记的形状（如圆圈、三角形等）。

cex参数用于设置标记的大小。

col参数用于设置标记的颜色。

08-连接点:

type参数用于指定连接点的方式，如折线、点等。

lwd参数用于设置线条的宽度。

lty参数用于设置线条的类型（如实线、虚线等）。

09-添加标签和标题:

main参数用于设置图表的标题。

xlab参数和ylab参数用于设置x轴和y轴的标签。

10-图例（Legend）:

legend()函数用于添加图例，可以指定位置、标签、线型、颜色等参数。

11-Overlaying Graphs:

lines()函数用于在现有的图形上添加额外的线条，可以用于叠加多个数据集。

12-Adding Lines:

abline()函数用于添加直线，可设置斜率和截距，或者添加水平线和垂直线。

**②实际案例-R语言中包的安装与使用**

01-从CRAN安装包：

使用install.packages()函数安装指定的包，如readr、ggplot2和tidyr。

02-从GitHub安装包：

首先安装devtools包，然后使用devtools::install\_github()函数安装指定GitHub仓库中的包，如rstudio/shiny。

03-使用install.packages()函数从特定的仓库中安装包，如furrr从指定的CRAN仓库。

04-从Zip文件安装包：

使用install.packages()函数从本地Zip文件安装包，需要指定文件路径和参数。

05-使用pak包管理器安装包

首先安装pak包：

使用install.packages()函数安装pak包。

使用pak::pkg\_install()函数从CRAN或Bioconductor安装包，或从GitHub安装包。

06-使用帮助文档：

使用help()函数查看特定包的帮助文档，如tidyverse。

使用vignette()函数查看包的示例文档。

使用browseVignettes()函数浏览包的示例文档。

使用demo()函数查看包的演示。

07-搜索帮助文档：

使用apropos()函数搜索包含特定字符串的函数。

使用help.search()函数搜索包中包含特定字符串的帮助文档。

08-创建和调用自定义函数

使用function()关键字创建函数，可以带有参数。

使用函数名和参数调用函数，如new.function()和new.function(10)。

09-Rstudio配置和使用

安装和配置gptstudio：

使用install.packages()函数安装pak包和gptstudio包。

使用pak::pak()函数安装usethis和MichelNivard/gptstudio包。

使用usethis::edit\_r\_environ()函数编辑Rstudio环境变量配置文件。

使用Sys.getenv()函数获取环境变量的值，如OPENAI\_API\_KEY。

使用gptstudio:::gptstudio\_chat()函数调用gptstudio包中的聊天功能。

**③实际案例-数据结构类型及ggplot**

**01-变量赋值**

使用等号（=）进行赋值，如 variable.1 = 124

使用左箭头（<-）进行赋值，如 variable.2 <- "Learn R Programming"

使用右箭头（->）进行赋值，如 133L -> variable.3

打印这些变量的值，使用 cat() 函数输出信息。

**02-数据类型和数据结构**

浮点数数据类型：定义了一个浮点数 pi，并用 class() 函数检查其类别，用 typeof() 函数查看其类型。

向量（Vectors）：通过 c() 函数创建了几个向量，包括数值向量（numeric vector）和字符向量（character vector）。

序列（Sequences）：使用 seq() 函数创建了一个序列。

列表（Lists）：通过 list() 函数创建了一个列表，其中包含向量、矩阵和其他列表。

矩阵（Matrices）：使用 matrix() 函数创建了一个矩阵，并使用 dimnames 参数设置了行和列的名称。

数组（Arrays）：使用 array() 函数创建了一个三维数组。

应用函数（apply function）：使用 apply() 函数对数组的某个维度进行操作，如求和。

数学运算：对矩阵进行了加法运算。

**03-创建数据框（Data Frame）：**

使用 data.frame() 函数创建一个包含员工信息的数据框 emp.data，其中包括员工ID、姓名、薪水和入职日期等信息。

使用 print() 函数显示数据框的内容。

使用 class() 函数查看数据框的类别。

**04-提取数据框的子集：**

通过索引提取数据框的子集，例如提取员工ID和薪水信息到新的数据框 final。

使用 [row\_index, ] 提取特定行，如提取第一行到新的数据框 final1。

使用 [row\_index\_range, ] 提取特定行的范围，如提取第4到5行到新的数据框 final2。

使用 [-row\_index, ] 排除特定行，如排除第一行创建新的数据框 emp.data.1。

使用 [ , column\_index] 提取特定列，如提取员工姓名到新的向量 emp.data.2。

直接使用 $ 运算符提取特定列，如提取员工姓名到新的向量 emp.data.3。

使用 [row\_index, column\_index] 提取特定行和列的单个元素，如提取第3行第2列的值。

**05-创建因子（Factor）：**

创建一个包含员工姓名的向量 data。

使用 factor() 函数将向量转换为因子，其中姓名被视为类别。

使用 is.factor() 函数检查对象是否为因子。

使用 levels 参数重新指定因子的级别顺序。

使用 t() 函数对矩阵或数据框进行转置操作

**06-矩阵转换为数据框：**

创建一个矩阵。

使用as.data.frame()函数将矩阵转换为数据框。

**07-数据框转换为矩阵：**

创建一个数据框。

使用data.matrix()函数将数据框中的一部分或全部转换为矩阵。

**08-数据框转换为数组：**

创建多个数据框。

使用array()函数将数据框中的数据转换为数组。

**09-从本地目录读取和保存数据文件：**

创建一个包含员工信息的数据框。

使用write.xlsx()和write.csv()函数将数据框保存为 Excel 和 CSV 文件。

使用read.xlsx()和read.csv()函数从文件中读取数据。

**10-学习数据结构和基本操作：**

从 CSV 文件中读取数据并存储在数据框中。

进行列操作，包括选择列并了解不同方式的结果。

操作因子变量，包括调整因子水平的顺序和将字符变量转换为因子变量。

1. **几个函数的操作：**

使用lubridate中的函数将年份、月份和日期拼接成日期向量，并将其转换为日期格式：

library(lubridate)

surveys$date <- ymd(paste(surveys$year, surveys$month, surveys$day, sep = "-"))

使用apply()函数对矩阵进行行或列操作，计算每列的和：

a\_m1 <- apply(m1, 2, sum)

使用lapply()函数对列表中的每个元素执行相同的操作：

movies\_lower <- lapply(movies, tolower)

使用sapply()函数对向量中的每个元素执行相同的操作，并返回向量：

smn\_cars <- sapply(dt, min)

使用tapply()函数对数据进行拆分、应用和组合操作：

tapply(iris$Sepal.Width, iris$Species, median)

使用tidyverse包中的函数进行数据操作，例如select()、filter()、mutate()、group\_by()、summarize()等：

surveys\_sml <- surveys %>%

filter(weight < 5) %>%

select(species\_id, sex, weight)

surveys %>%

group\_by(sex) %>%

summarize(mean\_weight = mean(weight, na.rm = TRUE))

使用gather()和spread()函数进行数据重塑：

surveys\_long <- surveys\_wide %>%

gather(key = species\_abbrev, value = count, -(month:plot\_id))

spread(surveys\_long, key = species\_abbrev, value = count)

最后，将处理后的数据导出为CSV文件：

write\_csv(surveys\_complete, "results/surveys\_complete.csv")

1. Ggplot

**散点图（Scatter Plot）：**

使用geom\_point()函数绘制散点图。

通过aes()函数定义了x轴和y轴的变量。

使用alpha参数调整透明度。

使用color参数按照物种（species\_id）对点进行着色。

**箱线图（Boxplot）：**

使用geom\_boxplot()函数绘制箱线图。

使用geom\_jitter()函数在箱线图上添加点以展示数据分布。

**时间序列数据可视化：**

对数据进行汇总计数后，使用geom\_line()函数绘制时间序列线图。

使用facet\_wrap()函数将同一图分为多个子图，每个子图根据不同的物种（species\_id）。

**主题定制（Theme Customization）：**

使用theme\_bw()函数将绘图主题设置为白色背景。

使用theme()函数进行更高级的主题自定义，如修改文本大小、轴标签的颜色、角度等。

**标签和标题（Labels and Titles）：**

使用labs()函数添加标题和轴标签。

**字体大小（Font Size）：**

使用theme()函数中的text=element\_text(size = 16)参数设置文本的字体大小。

**自定义主题（Custom Theme）：**

创建了一个自定义的主题grey\_theme，可以定义轴标签的颜色、字体大小等。

**组合多个图形（Arranging Plots）：**

使用gridExtra包中的grid.arrange()函数将多个图形组合到一个图形中。

使用ncol参数指定列数，widths参数指定每列的宽度比例。

**导出图形（Exporting Plots）：**

使用ggsave()函数将图形保存为图片文件。

第一个参数是文件路径，第二个参数是要保存的图形对象。

使用width和height参数指定输出图形的宽度和高度。

可以指定dpi参数来设置输出图形的分辨率。

**④实际案例-机器学习**

01-决策树回归模型 (rpart)

数据加载：使用read.csv()函数加载数据，并通过na.omit()函数去除缺失值。

标准化数据：使用scale()函数对数据进行标准化。

转换为虚拟变量：使用fastDummies::dummy\_cols()函数将分类变量转换为虚拟变量。

数据集拆分：将数据集分为训练集和测试集。

建立决策树模型：使用rpart()函数建立回归树模型。

可视化决策树：使用rpart.plot()函数可视化生成的决策树。

特征重要性评估：使用caret::varImp()函数评估变量的重要性。

模型评估：使用测试数据集评估模型性能，计算均方根误差（RMSE）。

02-随机森林回归模型 (randomForest)

建立随机森林回归模型：使用randomForest()函数建立随机森林回归模型。

特征重要性评估：使用caret::varImp()函数评估变量的重要性。

模型评估：使用测试数据集评估模型性能，计算均方根误差（RMSE）。

03-梯度提升回归树模型 (gbm)

建立梯度提升回归树模型：使用gbm()函数建立梯度提升回归树模型。

模型参数设置：设置梯度提升模型的参数，如树的数量、学习率等。

模型评估：使用测试数据集评估模型性能，计算均方根误差（RMSE）。

04-三种模型对比

模型介绍

训练决策树模型

使用 train() 函数训练决策树模型（method = "rpart"）。

在 trainControl 参数中指定交叉验证的方法和重复次数。

使用 preProcess 参数对数据进行预处理，包括标准化（缩放和中心化）。

通过 tuneLength 参数指定寻找最佳参数的长度。

预测测试集数据，并使用 Metrics::rmse() 函数评估模型性能。

训练随机森林模型

使用 train() 函数训练随机森林模型（method = "rf"）。

其他步骤与训练决策树模型类似。

训练梯度提升模型

使用 train() 函数训练梯度提升模型（method = "gbm"）。

其他步骤与训练决策树模型类似。

模型比较和评估

使用 resamples() 函数对模型进行比较。

使用 summary() 函数查看模型性能摘要。

使用 bwplot() 函数绘制箱线图比较模型性能。