

# 用R语言解读传染病模型

---

张丹, 20220618

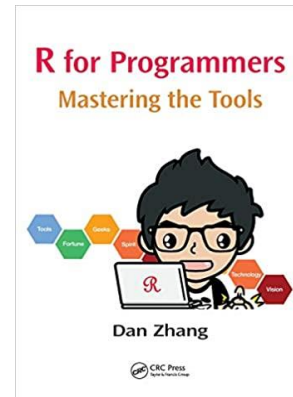
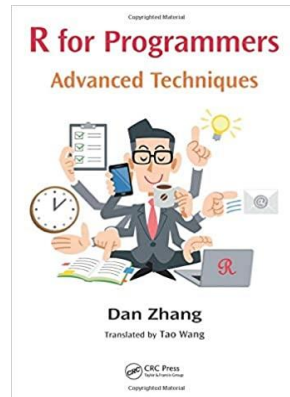
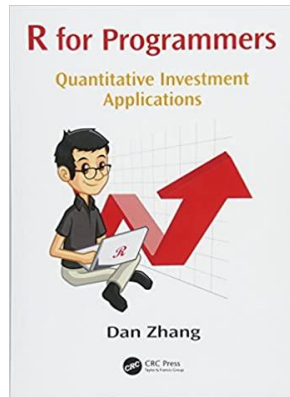
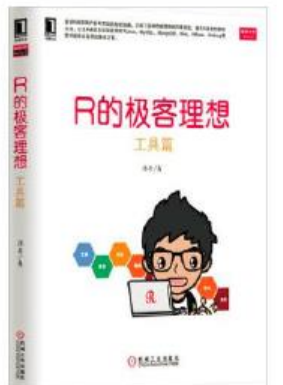
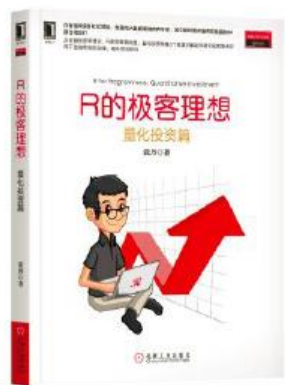
# 个人介绍

**张丹**， R语言实践者，北京青萌数海科技有限公司CTO，微软MVP。

10年以上互联网应用架构经验，在R、Java、NodeJS、大数据、数据挖掘等方面有深厚的积累。

精通量化投资交易策略，熟悉中国金融二级市场、交易规则和投研体系。熟悉数据学科方法论，在外汇、海关、区块链等领域均有落地的应用。

著有《R的极客理想：量化投资篇》、《R的极客理想：工具篇》、《R的极客理想：高级开发篇》，英文版图书被CRC出版集团引进，在美国发行。个人博客：<http://fens.me>。



## 背景

新冠疫情几次变异，极大地影响着我们的正常生活和工作。特别是2022年2月以来的Delta变异株感染，在上海和北京这种人口超大型城市中，有着超强的传染力。政府防疫工作的强力介入，隔离和居家已经是常态了，有新闻指出Delta变异株感染1人可传9人。

在流行病学领域，有几种不同传染病的传播模型，可以模拟病毒的传播过程。本次分享将使用R语言，来给大家演示病毒传播的过程。了解了病毒传播的逻辑，能让我们更加坚定战胜病毒的决心。

本次分享材料和代码：

<https://github.com/bsspirit/infect>

# 目录

1. 传染病模型原理：自由增长模型、SI模型、SIS模型、SIR模型
2. 用R语言手动实现
3. 基于EpiModel包的自动化实现
4. 如何获取新冠数据
5. 北京的数据带入模型预测

# 自由增长模型

## 模型假设：

- $I(t)$ 为 $t$ 时刻的感染者人数， $I(t)$ 连续、可微函数。
- $\rho$ 是常数，为每个病人，每天传染的人数， $\rho > 0$

## 模型构建：

从 $t$ 到 $t+\Delta t$ 的时间里，病人增加的人数为  $I(t+\Delta t) - I(t) = \rho * I(t) * \Delta t$

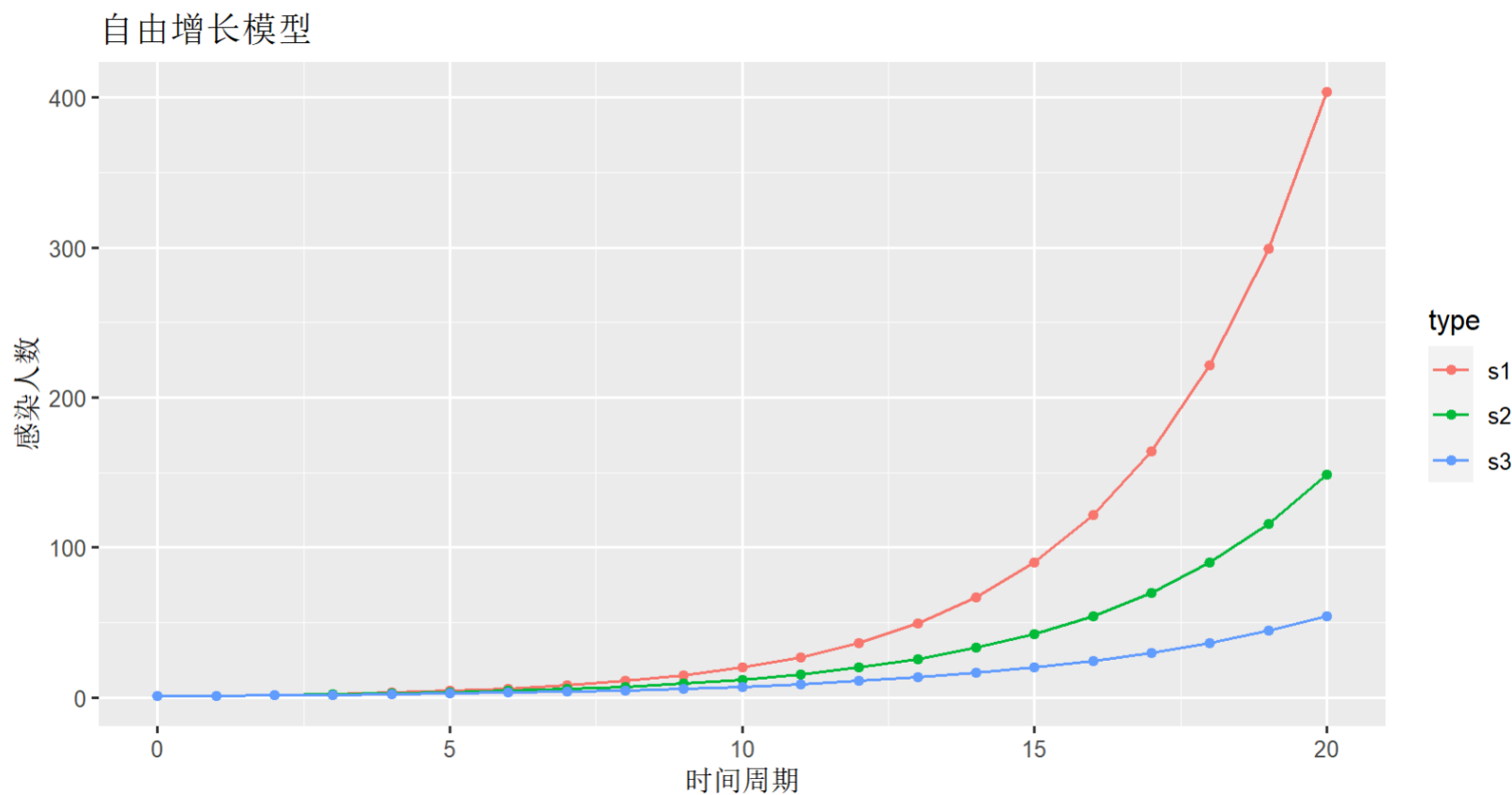
当 $t=0$ 时，初始病人人数为  $I_0$  得，

$$\frac{dI}{dt} = \rho * I$$

$$I(0) = I_0$$

# 自由增长模型：模拟的传染人数

初始：日接触率  $\rho = 0.3、0.25、0.2$ ，已感染者  $I_0 = 1$



自由增长模型整体上呈现指数增长趋势，对参数 $\rho$ 非常敏感。

模型适合于古代医疗条件不发达、不懂得对病人进行防疫隔离时，发生恶性瘟疫的情形。

# SI模型

**模型假设：**人群分为易感染者（Susceptible）和已感染者（Infective）两类（取两个词的首字母，称之为SI模型）；

- 在疾病传播期内所考察地区的总人数不变，不考虑生死和迁移。
- 当病人与健康者有效接触时，使健康者受感染变为病人。

**模型构建：**

已感染人数为  $N * i(t)$ ，所有已感染人每天有效接触人数为  $\rho * N * i(t)$ ，其中被传染的易感染人的比例  $s(t)$ ，因此每天增加的感染者人数为  $\rho * N * i(t) * s(t)$

- $N$ ：总人数，不变
- $s(t)$ ：t时刻易感染者在总人数中所占比例。
- $i(t)$ ：t时刻已感染者在总人数中所占比例。
- $\rho$ 是常数，每个病人每天有效接触的平均人数是常数，称为日接触率， $\rho > 0$

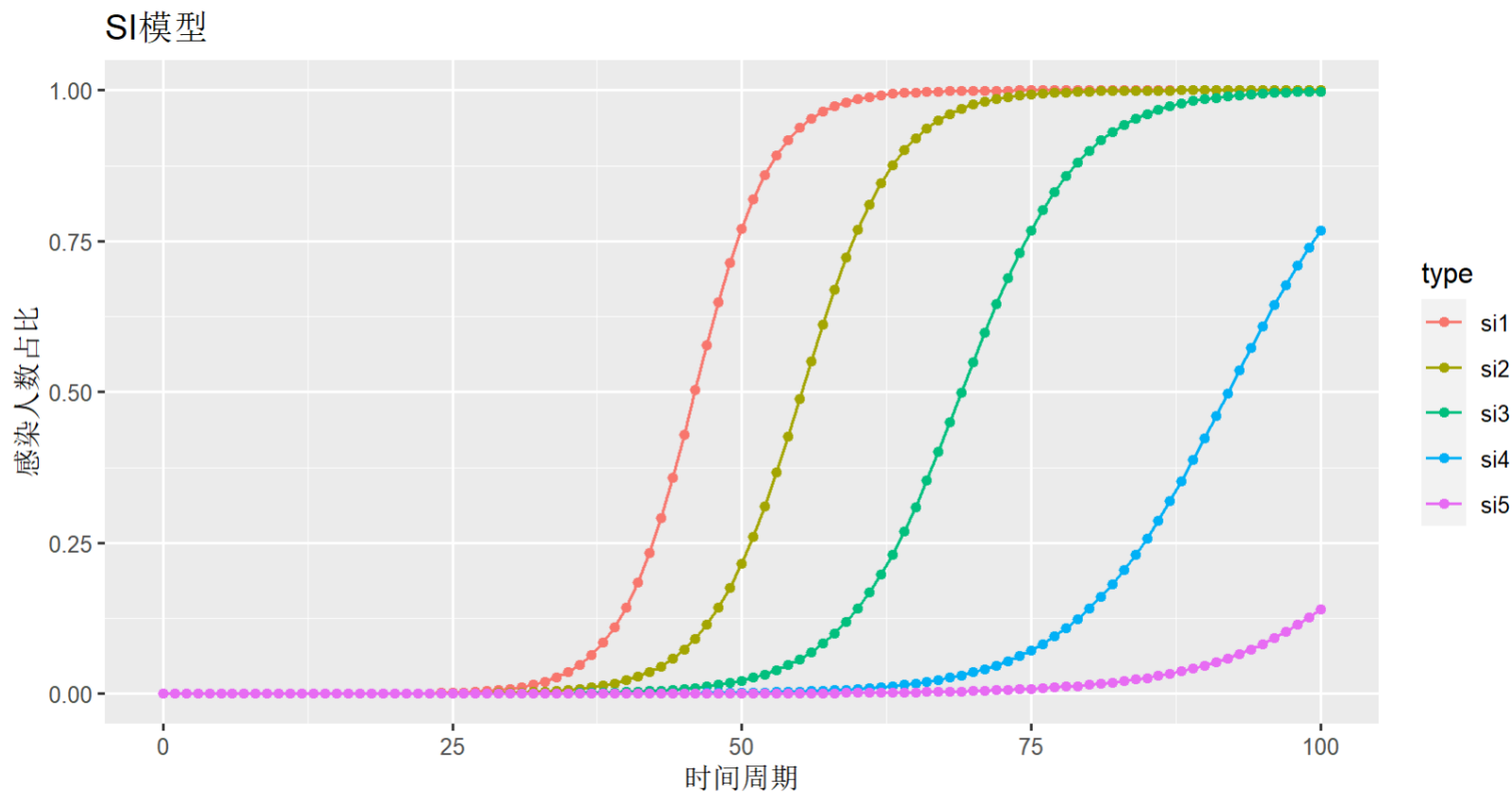
$$N * (i(t+\Delta t) - i(t)) = \rho * i(t) * s(t) * \Delta t$$

$N$ 总人数不变，因此  $s(t) + i(t) = 1$  得，

$$\frac{di}{dt} = \rho * i * (1-i); \quad i(0) = i_0$$

## SI模型：模拟的传染人数占比

初始：日接触率  $\rho = 0.3、0.25、0.2、0.15、0.12$ ，已感染者比例  $i_0 = 0.000001$ (百万分之一)



SI模型对参数 $\rho$ 非常敏感，而且只要我们把每个病人、每天接触的人数有效降低，传染病的传染速度就会变得非常慢。

只要防疫力度够大，控制住传染病是完全可能的。这种模型比较适合，被传染后无法恢复健康的，比如HIV等情形。



# SIS模型

SIS模型与SI模型的差异，在于SIS模型假设已感染者（Infective）可以被治愈，重新变成易感染者（Susceptible），比如季节性流感等。

**模型假设：**人群分为易感染者（Susceptible）和已感染者（Infective）两类，已感染者（Infective）可以被治愈，变成易感染者（Susceptible）（称之为SIS模型）

- 在疾病传播期内所考察地区的总人数不变，不考虑生死和迁移。
- 当病人与健康者有效接触时，使健康者受感染变为病人。
- 人被全部治愈，所需要的天数为 $1/\mu$ ，为传染病的平均传染期。

**模型构建：**

已感染人数为  $N * i(t)$ ，所有已感染人每天有效接触人数为  $\rho * N * i(t)$ ，其中被传染的易感染人的比例  $s(t)$ ，因此每天增加的感染者人数为  $\rho * N * i(t) * s(t)$ 。被治愈的人数为  $\mu * N * i(t)$ 。

$$N * (i(t+\Delta t) - i(t)) = \rho * i(t) * s(t) * \Delta t - \mu * N * i(t) * \Delta t$$

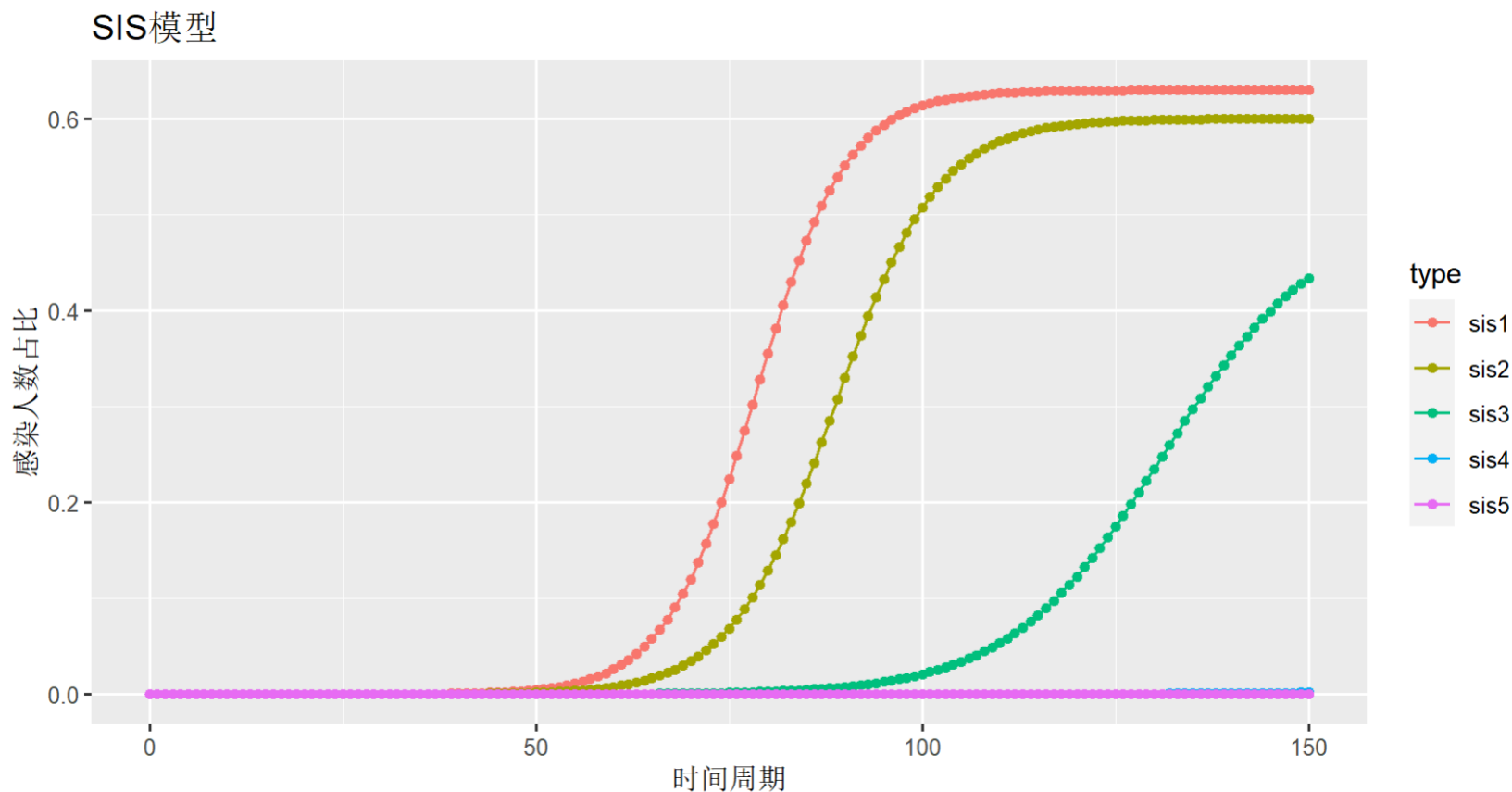
- $N$ ：总人数，不变
- $s(t)$ ：t时刻易感染者在总人数中所占比例。
- $i(t)$ ：t时刻已感染者在总人数中所占比例。
- $\rho$ 是常数，每个病人每天有效接触的平均人数是常数，为日接触率， $\rho > 0$
- $\mu$ 是常数，每天被治愈的病人数占病人总数的比例，为日治愈率

$N$ 总人数不变，因此  $s(t) + i(t) = 1$  得，是

$$\frac{di}{dt} = \rho * i * (1-i) - \mu * i ; \quad i(0) = i_0$$

# SIS模型：模拟的传染人数占比

初始：日接触率  $\rho = 0.3、0.25、0.2、0.15、0.12$ ，日治愈率  $\mu = 0.1$ ，已感染者比例  $i_0 = 0.000001$  (百万分之一)



1、传染的速度，取决于病人的传染速度，与病人的治愈速度之间的相对水平，如果病人的治愈速度，大于病人的传染速度，即 $\mu > \rho$ ，那么该传染病最终会消失；

2、即便病人的传染速度高于治愈速度，最终也只有一部分人群会被感染；

3、最终感染的人群比例，为 $1 - 1/\sigma$ ， $\sigma = \rho / \mu$ ， $\sigma$ 是整个传染期内每个病人的有效接触人数，可以理解为病人在整个生病期内，接触的总人数。

# SIR模型

SIR模型与SIS模型的差异，在于SIR模型假设已感染者（Infective）被治愈后会具备免疫力，不会被感染，从而成为恢复者（Removed），SIR模型已经初步接近实际的传染病模型。

**模型假设：**人群分为易感染者（Susceptible）、已感染者（Infective）和恢复者（Removed）三类；

- 在疾病传播期内所考察地区的总人数不变，不考虑生死和迁移。
- 当病人与健康者有效接触时，使健康者受感染变为病人。
- 人被全部治愈，所需要的天数为 $1/\mu$ ，为传染病的平均传染期。

**模型构建：**

已感染人数为  $N * i(t)$ ，已感染人每天有效接触人数为  $\rho * I(t)$ ，被传染的易感染人的比例  $S(t)/N$ ，因此每天增加的感染者人数为  $\rho * I(t) * S(t)/N$ ，被治愈的人数为  $\mu * I(t)$ 。

- $N$ ：总人数，不变
- $S(t)$ ： $t$ 时刻易感染者在总人数中所占比例。
- $I(t)$ ： $t$ 时刻已感染者在总人数中所占比例。
- $R(t)$ ： $t$ 时刻恢复者在总人数中所占比例。
- $\rho$ 是常数，每个病人每天有效接触的平均人数是常数，为日接触率， $\rho > 0$
- $\mu$ 是常数，每天被治愈的病人数占病人总数的比例，为日治愈率

$$S(t + \Delta t) - S(t) = -\rho * I(t) * \Delta t / N$$

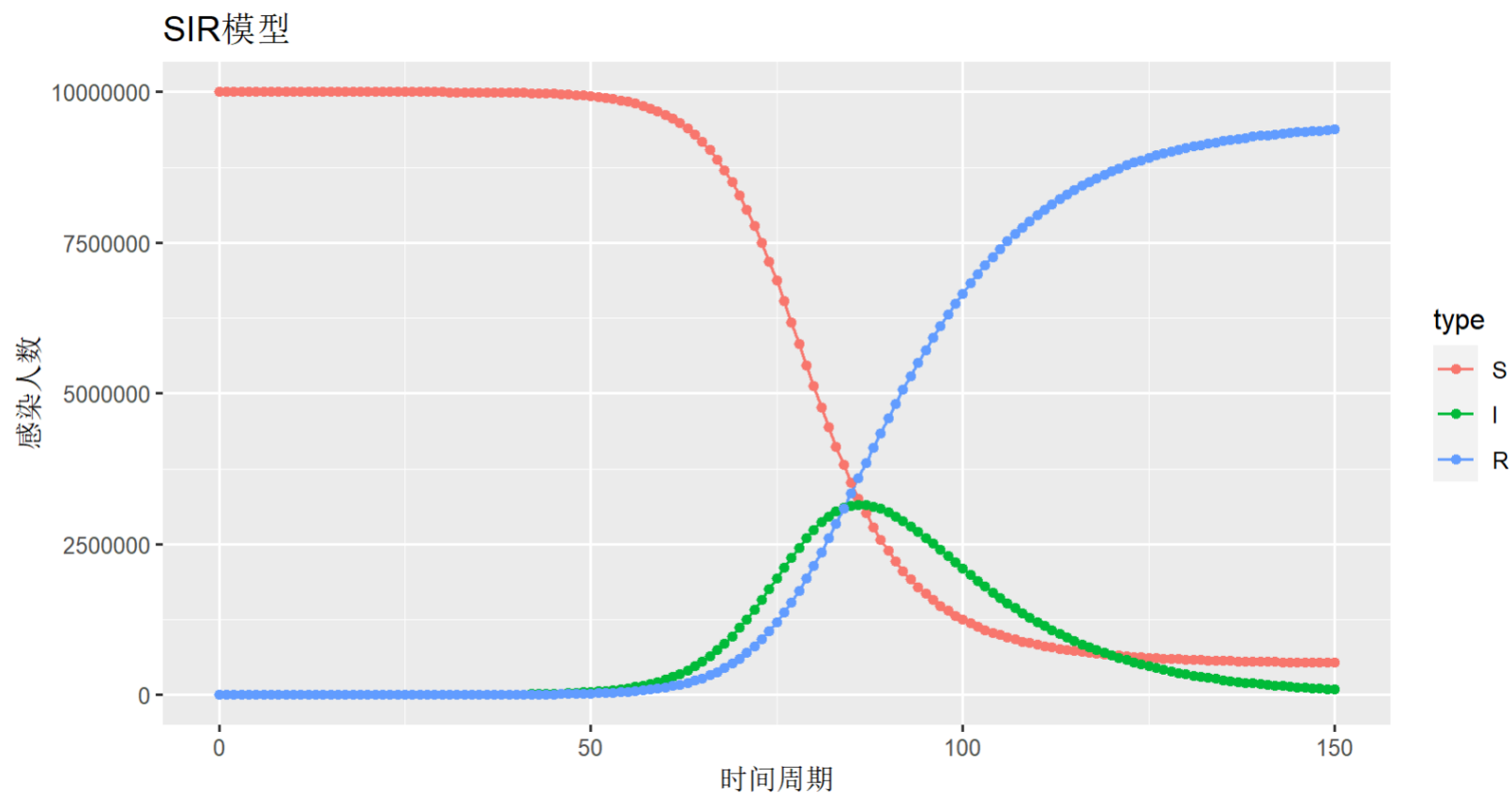
$$I(t + \Delta t) - I(t) = \rho * I(t) * S(t) * \Delta t / N - \mu * I(t) * \Delta t$$

$$R(t + \Delta t) - R(t) = \mu * I(t) * \Delta t$$

$N$ 总人数不变，因此  $s(t) + i(t) = 1$  得，是  $\frac{dS}{dt} = -\rho * I * S/N$ ;  $\frac{dI}{dt} = \rho * I * S/N - \mu * I$ ;  $\frac{dR}{dt} = \mu * I$

## SIR模型：模拟的传染人数

初始：日接触率 $\rho = 0.25$ ，日治愈率 $\mu = 0.1$ ，易感染者 $S = 1000 \times 1000 \times 10$ ，已感染者 $I = 10$ ，移出者 $R = 100$



# 目录

1. 传染病模型原理：自由增长模型、SI模型、SIS模型、SIR模型
2. 用R语言手动实现
3. 基于EpiModel包的自动化实现
4. 如何获取新冠数据
5. 北京的数据带入模型预测

# R程序实现：程序结构

自由增长模型为例

初始：日接触率  $\rho = 0.3、0.25、0.2$ ，已感染者  $I_0 = 1$

$$\frac{dI}{dt} = \rho * I$$

$$I(0) = I_0$$

```
> calc1 <- function(rho = 0.3,           # 传染率系数,每个病人每天传染的人数
+                               times = 0:20, # 病情发展时间
+                               I = 1) {     # 已感染者1人
+   # 公式
+   s_equations <- function(times, vars, params) {
+     with(as.list(c(vars, params)), {
+       dI <- rho * I
+       return(list(dI))
+     })
+   }
+   # 解微分方程
+   ode(
+     func = s_equations,
+     y = c(I = I),
+     times = times,
+     parms = c(rho = rho)
+   ) %>% as.data.frame
+ }
> s1 <- calc1(rho = 0.3)
> s2 <- calc1(rho = 0.25)
> s3 <- calc1(rho = 0.2)
> sdf1<-rbinddf(list(s1,s2,s3),names=c("s1","s2","s3"))
> head(sdf1)
  time      I type
1    0 1.000000  s1
2    1 1.349861  s1
3    2 1.822121  s1
4    3 2.459609  s1
5    4 3.320123  s1
6    5 4.481699  s1
> names(sdf1)<-c("time","value","type")
> draw(sdf1,title="自由增长模型",ylab="感染人数",xlab="时间周期")
```

## R程序实现：公式实现

SI模型公式:  $\frac{di}{dt} = \rho * i * (1-i)$

SIS模型公式:  $\frac{di}{dt} = \rho * i * (1-i) - \mu * i$

SIR模型公式:  $\frac{dS}{dt} = -\rho * I * S/N$   
 $\frac{dI}{dt} = \rho * I * S/N - \mu * I$   
 $\frac{dR}{dt} = \mu * I$

```
# 公式
si_equations <- function(time, vars, params) {
  with(as.list(c(vars, params)), {
    di <- rho * i * (1 - i)
    return(list(di))
  })
}
```

```
sis_equations <- function(time, vars, params) {
  with(as.list(c(vars, params)), {
    di <- rho * i * (1 - i) - mu * i
    return(list(c(di)))
  })
}
```

```
sir_equations <- function(time, vars, params) {
  with(as.list(c(vars, params)), {
    N <- S + I + R
    dS <- -rho * I * S / N
    dI <- rho * I * S / N - mu * I
    dR <- mu * I
    return(list(c(dS, dI, dR)))
  })
}
```

# 目录

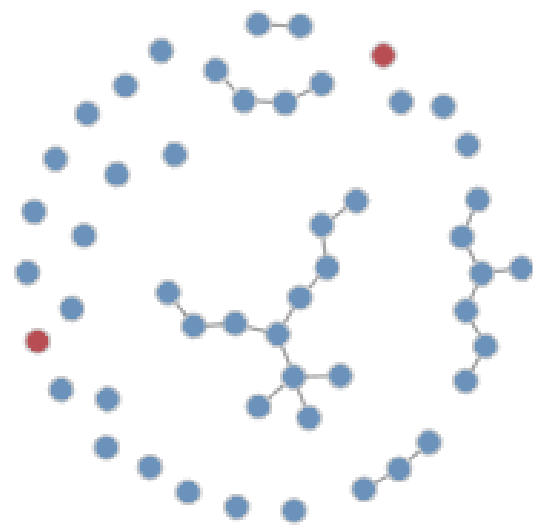
1. 传染病模型原理：自由增长模型、SI模型、SIS模型、SIR模型
2. 用R语言手动实现
3. 基于EpiModel包的自动化实现
4. 如何获取新冠数据
5. 北京的数据带入模型预测



# EpiModel包介绍

EpiModel 提供了用于模拟和分析传染病动力学数学模型的工具，支持的流行病模型类包括确定性隔间模型、随机个体接触模型和随机网络模型。疾病类型包括有和没有人口统计的 SI、SIR 和 SIS 流行病，具有可用于扩展的实用程序，以构建和模拟任意复杂性的流行病模型。网络模型类基于在 R 的 Statnet 软件套件中实现的时间指数随机图模型 (ERGM) 的统计框架。

官方网站: <http://www.epimodel.org/index.html>



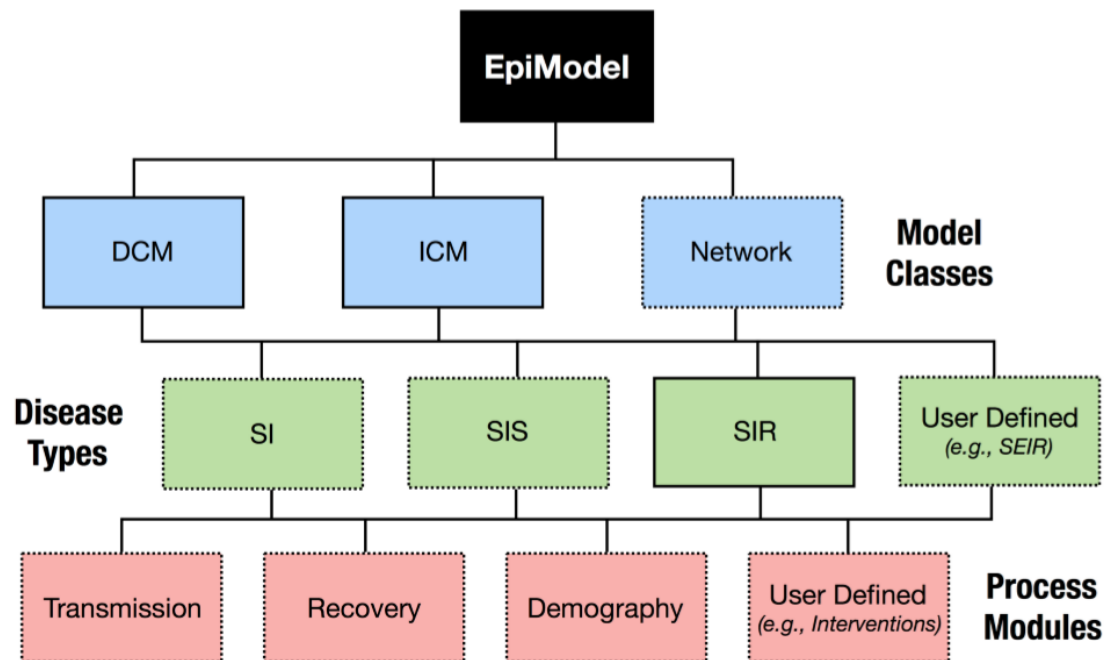
# EpiModel包结构

三类流行病模型提供功能：

- **确定性隔间模型** (Deterministic Compartmental Models)：模型是确定性的，时间是连续的，解是输入参数和初始条件的固定数学函数，在疾病和人口过渡过程中没有随机变异性。
- 随机个体接触模型 (Stochastic Individual Contact Models)：参数是随机抽取，其中控制状态间转移的所有比率和风险都是随机抽取的，它们来自由这些比率或概率总结的分布，包括正态分布、泊松分布和二项分布。时间是离散的，ICMs是在离散时间内模拟的，而DCMs是在连续时间内模拟的。
- 随机网络模型 (Stochastic Network Models)：两个节点之间连边与否不再是确定的事情，而是根据一个概率决定。网络动力学和疾病传播动力学，这两个动态过程是被视为独立的或者相互依赖的。独立的网络模型假设疾病模拟对时间网络的结构没有影响，尽管时间网络的结构肯定会影响疾病。依赖网络模型允许流行病学和人口统计过程影响网络结构。

三种传染病类型：

- SI: Susceptible-Infective, 一种两种状态的疾病，其中存在终生感染而无法恢复。HIV/AIDS 就是一个例子，尽管在这种情况下，通常将感染阶段建模为单独的隔间。
- SIR: Susceptible-Infective-Removed, 一种三期疾病，感染后可终生恢复并产生免疫力。麻疹就是一个例子，但该疾病的现代模型也需要考虑人群中的疫苗接种模式。
- SIS: Susceptible-Infective- Susceptible, 一种两阶段疾病，其中一个人可能在一生中从易感状态到受感染状态来回转换。例子包括细菌性传播疾病，如淋病。



# EpiModel : SI模型

初始：日感染率 inf.prob=0.5，日传播率 act.rate=0.5

初始易感染者s.num=500，初始已感染者i.num=1，观察周期nsteps=100

```
> param <- param.dcm(inf.prob = 0.5, act.rate = 0.5)
> init <- init.dcm(s.num = 500, i.num = 1)
> control <- control.dcm(type = "SI", nsteps = 100)
> mod <- dcm(param, init, control)
> mod
```

EpiModel Simulation

=====

Model class: dcm

Simulation Summary

-----

Model type: SI

No. runs: 1

No. time steps: 100

No. groups: 1

Model Parameters

-----

inf.prob = 0.5

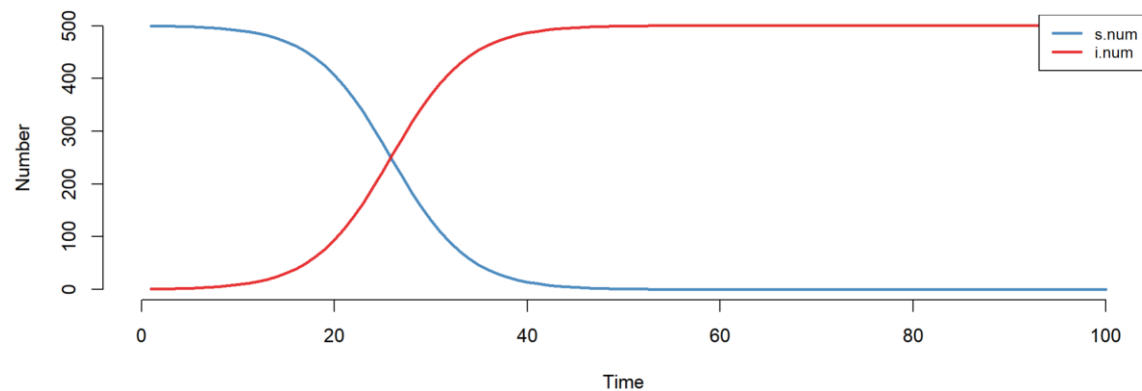
act.rate = 0.5

Model Output

-----

Variables: s.num i.num si.flow num

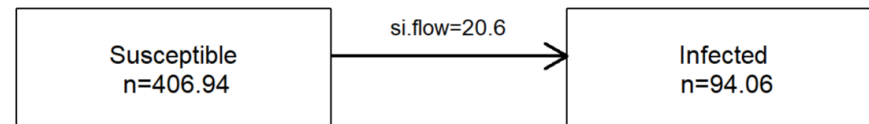
```
> plot(mod)
```



```
> comp_plot(mod, at = 20, digits = 2)
```

## SI Model Diagram

time=20 | run=1



# EpiModel 核心函数

```
> param.dcm
function (inf.prob, inter.eff, inter.start, act.rate, rec.rate,
          a.rate, ds.rate, di.rate, dr.rate, inf.prob.g2, act.rate.g2,
          rec.rate.g2, a.rate.g2, ds.rate.g2, di.rate.g2, dr.rate.g2,
          balance, ...)
```

参数	描述
inf.prob	感染率，易感者和感染者之间每次可传播行为的感染概率。
inter.eff	影响每行为感染概率的干预措施的效果。
inter.start	干预开始的时间步长，在 1 和模型中指定的时间步长数之间。
act.rate	传播率，每人每单位时间的平均可传播行为数
rec.rate	治愈率，具有免疫力（在 SIR 模型中）或再易感性（在 SIS 模型中）的平均治愈率。
a.rate	到达率或出生率。对于一组模型，到达率是每人每单位时间的新到达率。
ds.rate	易感人者离开率或死亡率
di.rate	已感染者离开率或死亡率
dr.rate	治愈者离开率或死亡率，仅用于 SIR 模型。
inf.prob.g2	第 2 组感染率
act.rate.g2	第 2 组传播率，每组平均传播行为数 2 人/单位时间
rec.rate.g2	第 2 组治愈率，具有免疫（在 SIR 模型中）或再易感性（在 SIS 模型中）的平均恢复率。
a.rate.g2	第 2 组到达率。
ds.rate.g2	第 2 组死亡率。
di.rate.g2	第 2 组疾病死亡率。
dr.rate.g2	第 2 组的已治愈的离开率，仅用于 SIR 模型类型。
balance	对于两组模型，将 act.rate 平衡为组 1（balance="g1"）或组 2（balance="g2"）设置的速率

```
> init.dcm
function (s.num, i.num, r.num, s.num.g2, i.num.g2, r.num.g2,
          ...)
```

参数	描述
s.num	初始易感染者人数
i.num	初始已感染者人数。
r.num	初始已治愈者人数，仅用于 SIR
s.num.g2	第 2 组初始易感染者人数
i.num.g2	第 2 组初始已感染者人数。
r.num.g2	第 2 组初始已治愈者人数，仅用于 SIR

# EpiModel : SIR模型人口分析

在易感-感染-恢复 (SIR) 模型中, 受感染的个体从疾病转变为终生恢复状态, 在这种状态下他们将不再易感。

初始: 日感染率 $\text{inf.prob}=0.2$ , 日传播率 $\text{act.rate}=1$ , 恢复率 $\text{rec.rate}=1/20$ , 出生率 $\text{a.rate}=1/95$ , 死亡率 $\text{ds.rate}=1/100$ , 疾病死亡率 $\text{di.rate}=1/80$ , 恢复率 $\text{dr.rate}=1/100$ 。

易感染者 $\text{s.num}=1000$ , 已感染者 $\text{i.num}=1$ , 治愈 $\text{r.num}=0$ , 观察周期 $\text{nsteps}=500$ 。

```
> param <- param.dcm(inf.prob = 0.2, act.rate = 1, rec.rate = 1/20,
+                   a.rate = 1/95, ds.rate = 1/100, di.rate = 1/80, dr.rate = 1/100)
> init <- init.dcm(s.num = 1000, i.num = 1, r.num = 0)
> control <- control.dcm(type = "SIR", nsteps = 500, dt = 0.5)
> mod <- dcm(param, init, control)
> mod
```

EpiModel Simulation

Model class: dcm

Simulation Summary

Model type: SIR

No. runs: 1

No. time steps: 500

No. groups: 1

Model Parameters

$\text{inf.prob} = 0.2$

$\text{act.rate} = 1$

$\text{rec.rate} = 0.05$

$\text{a.rate} = 0.01052632$

$\text{ds.rate} = 0.01$

$\text{di.rate} = 0.0125$

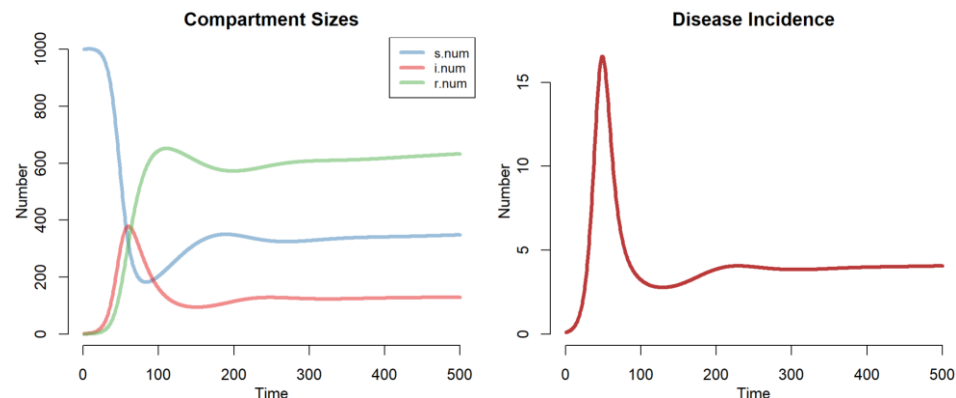
$\text{dr.rate} = 0.01$

Model Output

Variables: s.num i.num r.num si.flow ir.flow a.flow

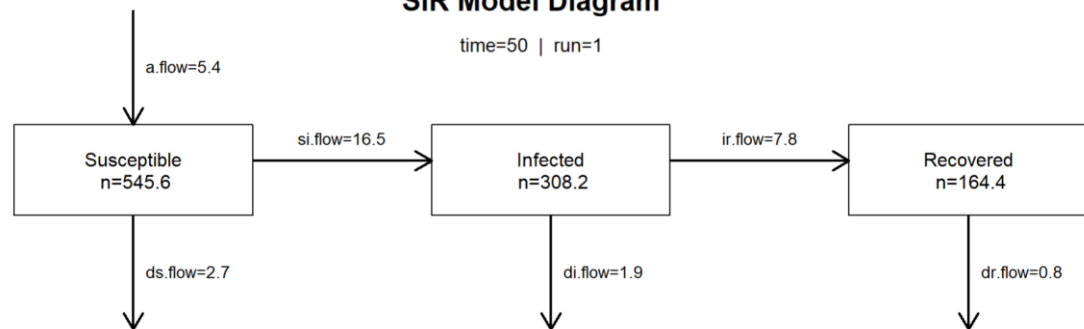
ds.flow di.flow dr.flow num

```
> par(mar = c(3.2, 3, 2, 1), mgp = c(2, 1, 0), mfrow = c(1, 2))
> plot(mod, popfrac = FALSE, alpha = 0.5,
+      lwd = 4, main = "Compartment Sizes")
> plot(mod, y = "si.flow", lwd = 4, col = "firebrick",
+      main = "Disease Incidence", legend = "n")
```



SIR Model Diagram

time=50 | run=1



# EpiModel : SIS敏感性分析

在易感-感染-恢复 (SIS) 模型中，我们在一个封闭的群体中对此进行，感染力和恢复等式相互反映，因为个体在状态之间来回流动。

建模初始：日感染率 $\text{inf.prob}=(0.1\ 0.1\ 0.2\ 0.1\ 0.2\ 0.2)$ ，日传播率 $\text{act.rate}=(0.2\ 0.4\ 0.4\ 0.6\ 0.6\ 1)$ ，日治愈率 $\text{rec.rate}=0.02$

初始易感染者 $\text{s.num}=500$ ，初始已感染者 $\text{i.num}=1$ ，观察周期 $\text{nsteps}=350$

```
> inf<-c(0.1, 0.1, 0.2, 0.1, 0.2,0.2)
> act<-c(0.2, 0.4, 0.4, 0.6, 0.6,1)
> param <- param.dcm(inf.prob = inf, act.rate = act, rec.rate = 0.02)
> init <- init.dcm(s.num = 500, i.num = 1)
> control <- control.dcm(type = "SIS", nsteps = 350)
> mod <- dcm(param, init, control)
> mod
```

EpiModel simulation

Model class: dcm

Simulation Summary

Model type: SIS

No. runs: 6

No. time steps: 350

No. groups: 1

Model Parameters

$\text{inf.prob} = 0.1\ 0.1\ 0.2\ 0.1\ 0.2\ 0.2$

$\text{act.rate} = 0.2\ 0.4\ 0.4\ 0.6\ 0.6\ 1$

$\text{rec.rate} = 0.02$

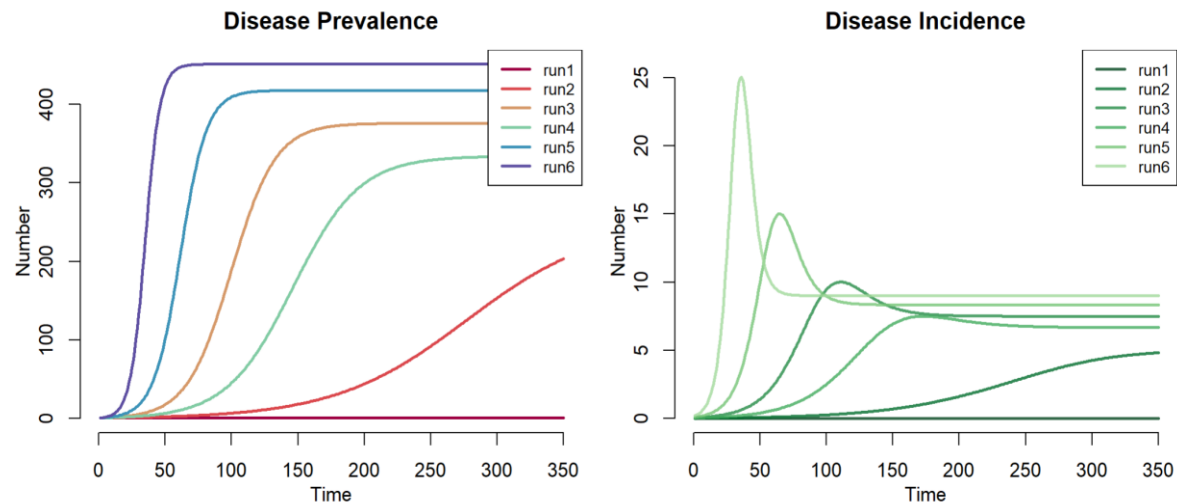
Model Output

Variables: s.num i.num si.flow is.flow num

```
> par(mfrow = c(1,2), mar = c(3.2,3,2.5,1))
```

```
> plot(mod, alpha = 1, main = "Disease Prevalence", legend = "full")
```

```
> plot(mod, y = "si.flow", col = "Greens", alpha = 0.8, main = "Disease Incidence", legend = "full")
```



# EpiModel: 2组混合SI模型异性恋疾病传播分析

在当前结构的两组模型中，组之间的混合纯粹是异质的：一个组只与另一组有联系，没有组内联系。

异性恋疾病传播，其中群体代表性别，我们可以将第 1 组建模为女性，将第 2 组建模为男性。

建模初始：日感染率女性 $\text{inf.prob}=0.4$ ，男性 $\text{inf.prob.g2}=0.1$

易感染者女性 $\text{s.num}=500$ 男性 $\text{s.num.g2}=500$ ，已感染者女性 $\text{i.num}=1$ ，男性 $\text{i.num.g2}=0$ ，观察周期 $\text{nsteps}=500$

```
> param <- param.dcm(inf.prob = 0.4, inf.prob.g2 = 0.1, act.rate = 0.25, balance = "g1",
+                   a.rate = 1/100, a.rate.g2 = NA, ds.rate = 1/100, ds.rate.g2 = 1/100,
+                   di.rate = 1/50, di.rate.g2 = 1/50)
> init <- init.dcm(s.num = 500, i.num = 1, s.num.g2 = 500, i.num.g2 = 0)
> control <- control.dcm(type = "SI", nsteps = 500)
> mod <- dcm(param, init, control)
> mod
```

```
EpiModel Simulation
=====
```

```
Model class: dcm
```

```
Simulation Summary
```

```
-----
Model type: SI
```

```
No. runs: 1
```

```
No. time steps: 500
```

```
No. groups: 2
```

```
Model Parameters
```

```
-----
inf.prob = 0.4
```

```
act.rate = 0.25
```

```
a.rate = 0.01
```

```
ds.rate = 0.01
```

```
di.rate = 0.02
```

```
inf.prob.g2 = 0.1
```

```
a.rate.g2 = NA
```

```
ds.rate.g2 = 0.01
```

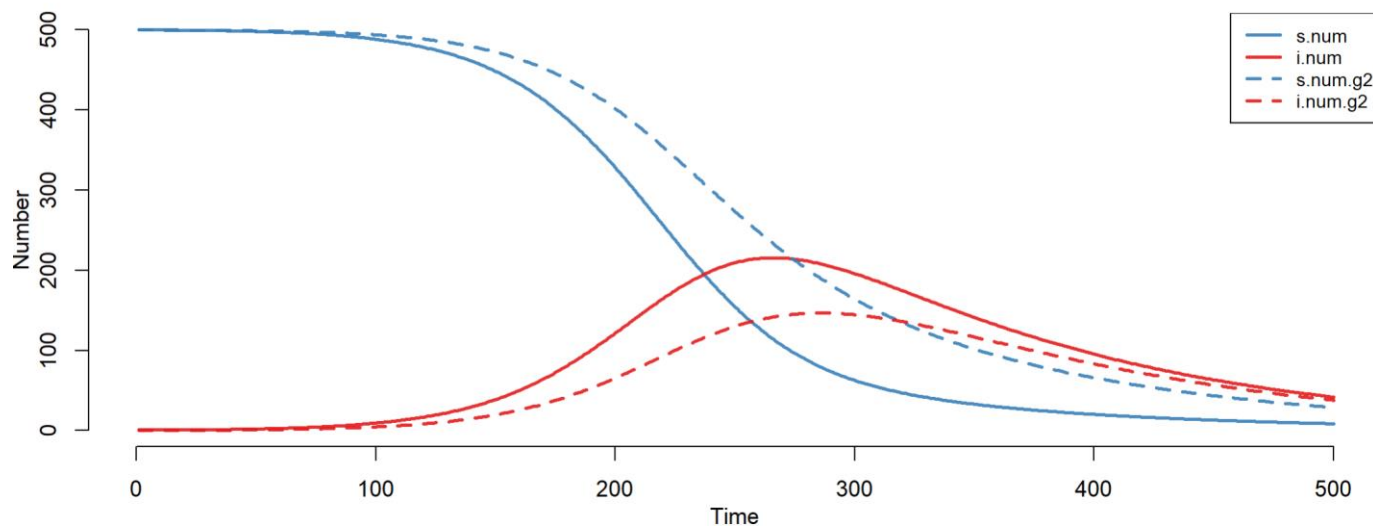
```
di.rate.g2 = 0.02
```

```
balance = g1
```

```
Model Output
```

```
-----
Variables: s.num i.num s.num.g2 i.num.g2 si.flow a.flow
ds.flow di.flow si.flow.g2 a.flow.g2 ds.flow.g2 di.flow.g2
num num.g2
```

```
> plot(mod)
```





# EpiModel: 随机网络传播模型

官方例子：血清监测：疾病状况影响伴侣选择——人口统计过程(出生、死亡和迁移)——接触网络过程必须适应人口规模和组成的变化。

- 对于独立模型，在每次疫情模拟开始时对全动态网络进行模拟。
- 对于相关模型，在每个时间步长的情况下，网络被重新模拟成变化的种群大小和变化的节点属性的函数。

```
# 模拟流行病传播模型
param <- param.net(inf.prob = 0.1, act.rate = 5, rec.rate = 0.02)
status.vector <- c(rbinom(500, 1, 0.1), rep(0, 500)) # 二项分布
status.vector <- ifelse(status.vector == 1, "i", "s")
init <- init.net(status.vector = status.vector)
control <- control.net(type = "SIS", nsteps = 500, nsims = 10, epi.by = "race")
```

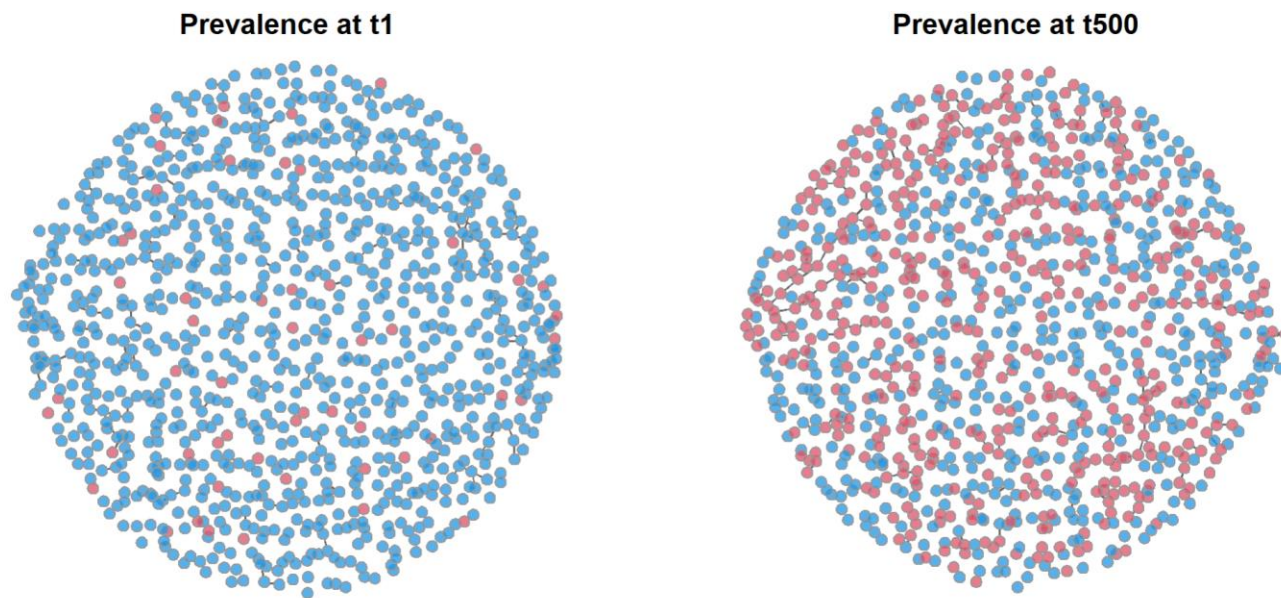
```
#模型构建
sim1 <- netsim(est1, param, init, control)
```

```
#数据结果
head(as.data.frame(sim1), 10)
```

```
#获得网络状态
nw <- get_network(sim1, sim = 1)
```

```
#获得每个时间点的时间状态
get_transmat(sim1, sim = 1)
```

```
#绘制任意时间点的网络状态
par(mfrow = c(1,2), mar = c(0,0,1,0))
plot(sim1, type = "network", at = 1, col.status = TRUE,
     main = "Prevalence at t1")
plot(sim1, type = "network", at = 500, col.status = TRUE,
     main = "Prevalence at t500")
```



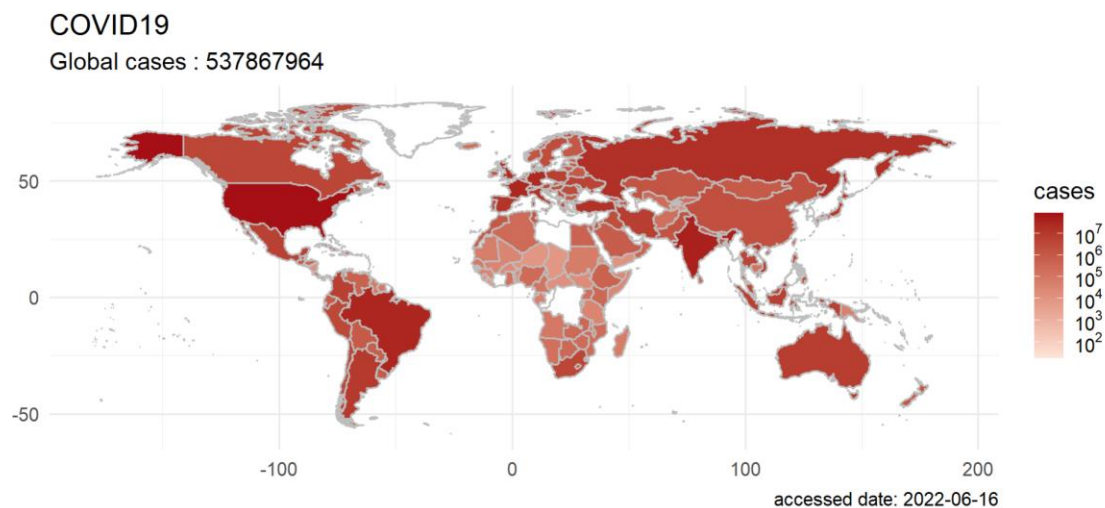


# 目录

1. 传染病模型原理：自由增长模型、SI模型、SIS模型、SIR模型
2. 用R语言手动实现
3. 基于EpiModel包的自动化实现
4. 如何获取新冠数据
5. 北京的数据带入模型预测

# 疫情数据获取: nCov2019

```
remotes::install_github("GuangchuangYu/nCov2019")
library(nCov2019)
x <- load_nCov2019()
plot(x)
```



```
> str(x)
List of 3
 $ table      : 'data.frame':      174523 obs. of  5 variables:
   ..$ country   : chr [1:174523] "Afghanistan" "Afghanistan" "Afghanistan" "A
   ..$ date      : Date[1:174523], format: "2020-01-22" "2020-01-23" "2020-01-
   ..$ cases     : int [1:174523] 0 0 0 0 0 0 0 0 0 0 ...
   ..$ deaths    : int [1:174523] 0 0 0 0 0 0 0 0 0 0 ...
   ..$ recovered : int [1:174523] 0 0 0 0 0 0 0 0 0 0 ...
 $ province: 'data.frame':      78053 obs. of  6 variables:
   ..$ country   : chr [1:78053] "Australia" "Australia" "Australia" "Australi
   ..$ province  : chr [1:78053] "australian capital territory" "new south wale
   ..$ date      : Date[1:78053], format: "2020-01-22" "2020-01-22" "2020-01-2
   ..$ cases     : int [1:78053] 0 0 0 0 0 0 0 0 0 0 ...
   ..$ deaths    : int [1:78053] 0 0 0 0 0 0 0 0 0 0 ...
   ..$ recovered : int [1:78053] 0 0 0 0 0 0 0 0 0 0 ...
 $ time       : chr "2022-06-16"
 - attr(*, "class")= chr "nCov2019History"
> chinadf<-x$province[which(x$province$country=='China'),]
> head(chinadf)
   country province      date cases deaths recovered
3   China   anhui 2020-01-22     1      0          0
6   China  beijing 2020-01-22    14      0          0
13  China chongqing 2020-01-22     6      0          0
21  China   fujian 2020-01-22     1      0          0
22  China   gansu 2020-01-22     0      0          0
27  China guangdong 2020-01-22    26      0          0
> beijingdf<-chinadf[which(chinadf$province=='beijing'),]
> head(chinadf)
   country province      date cases deaths recovered
3   China   anhui 2020-01-22     1      0          0
6   China  beijing 2020-01-22    14      0          0
13  China chongqing 2020-01-22     6      0          0
21  China   fujian 2020-01-22     1      0          0
22  China   gansu 2020-01-22     0      0          0
27  China guangdong 2020-01-22    26      0          0
```

# 疫情数据可视化

```
#install.packages(c("shiny","shinycssloaders","shinydashboard","plotly"))
open_dashboard()
```



# 目录

1. 传染病模型原理：自由增长模型、SI模型、SIS模型、SIR模型
2. 用R语言手动实现
3. 基于EpiModel包的自动化实现
4. 如何获取新冠数据
5. 北京的数据带入模型预测

# R0基本传染指数

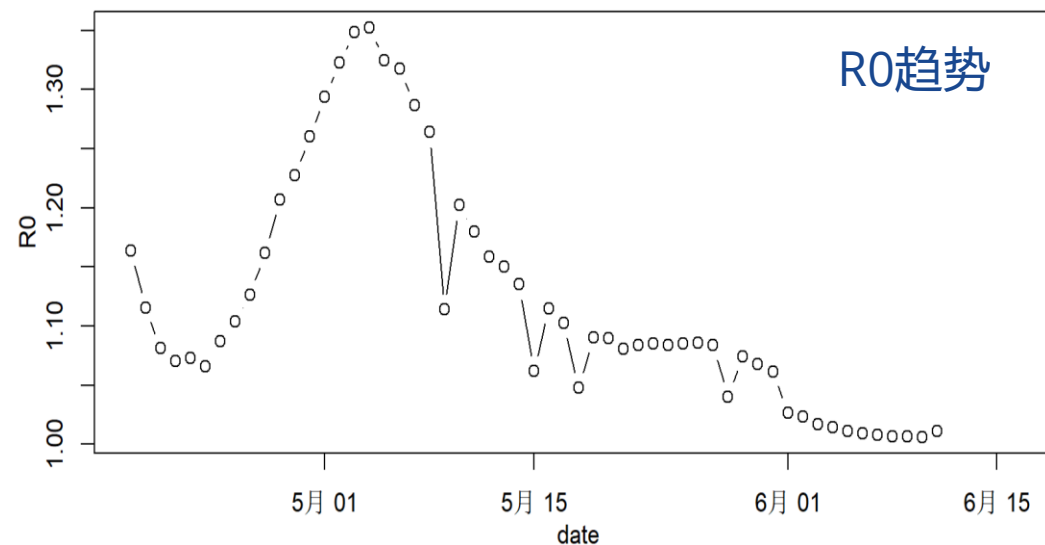
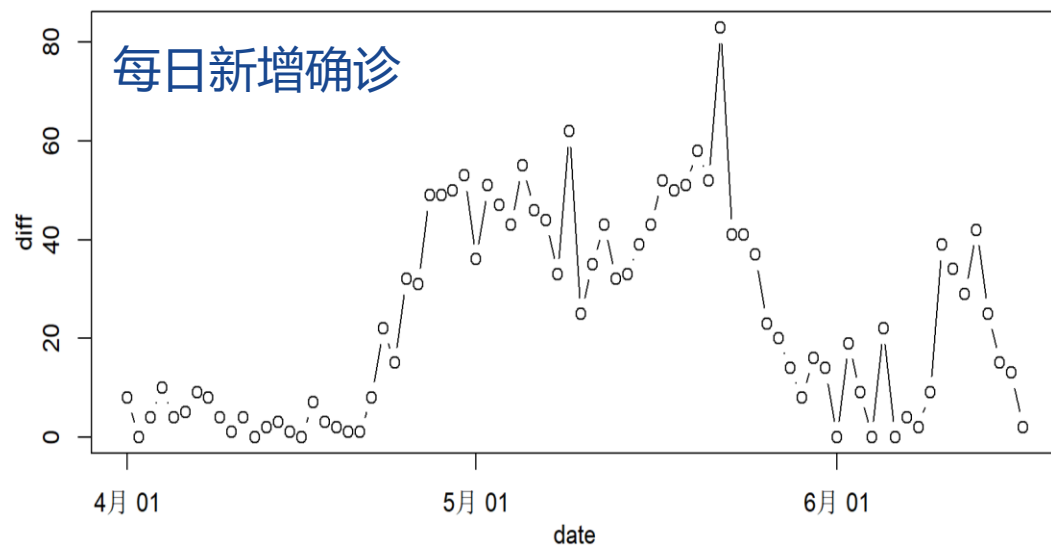
R0（基本传染指数）指的是，在完全易感人群中，由典型病例引起的二次感染人数的平均数量。

R0可以帮助我们衡量病毒的传染性：

- 如果 $R_0 < 1$ ，则意味着疾病得到控制或传播速度不太快；
- 如果 $R_0 = 1$ ，则平均1人可以传播到另1人；
- 如果 $R_0 > 1$ ，那么这种疾病可以从1个人传播到更广泛的人群中。

$$R_0 = \text{传染期} * \text{每人每天的接触者数} * \text{每个接触者的感染概率}$$

## 取北京2022-04-01至今的数据



# EpiModel: SEIR模型

易感-潜伏-感染-恢复 (SEIR)：模型扩展了 SIR 模型以包括潜伏期但非传染性的类别，模型虑了开放人群中易感人群、潜伏人群、感染者的比例。R0 是初始繁殖数，e.dur 是潜伏状态的持续时间，i.dur 是传染状态的持续时间，cfr 是病死率。

```
> param <- param.dcm(R0 = 1.9, e.dur = 10, i.dur = 14, cfr = c(0.5, 0.7, 0.9))
> init <- init.dcm(s.num = 1e6, e.num = 10, i.num = 0, r.num = 0,
+               se.flow = 0, ei.flow = 0, ir.flow = 0, d.flow = 0)
> control <- control.dcm(nsteps = 500, dt = 1, new.mod = SEIR)
> mod <- dcm(param, init, control)
> mod
```

EpiModel simulation

=====

Model class: dcm

Simulation Summary

-----

No. runs: 3

No. time steps: 500

Model Parameters

-----

R0 = 1.9

e.dur = 10

i.dur = 14

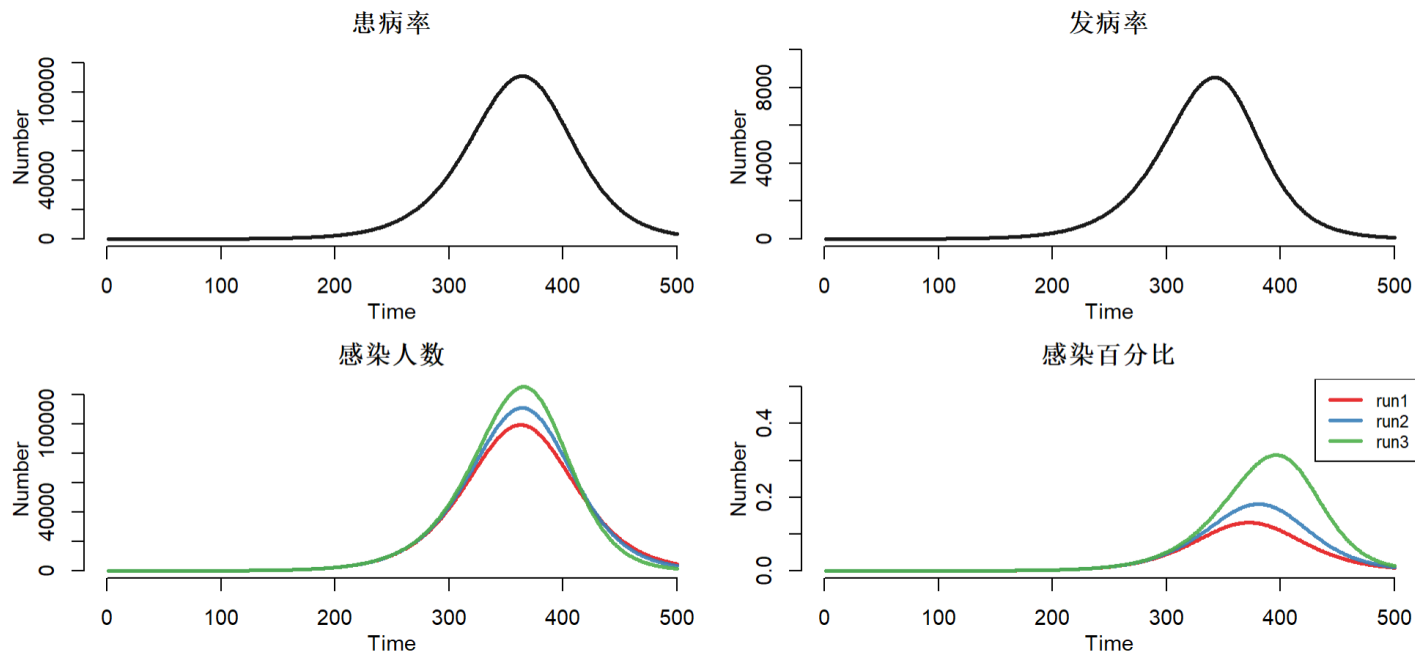
cfr = 0.5 0.7 0.9

Model Output

-----

Variables: s.num e.num i.num r.num se.flow ei.flow ir.flow  
d.flow num i.prev ei.prev

```
> par(mfrow = c(2, 2))
> plot(mod, y = "i.num", run = 2, main = "患病率")
> plot(mod, y = "se.flow", run = 2, main = "发病率")
> plot(mod, y = "i.num", main = "感染人数")
> plot(mod, y = "i.prev", main = "感染百分比", ylim = c(0, 0.5), legend = "full")
```



## 参考资料

1. <https://zhuanlan.zhihu.com/p/104439779>
2. <https://cloud.tencent.com/developer/article/1607390>
3. <http://statnet.org/tut/BasicDCMs.html>
4. <https://github.com/GuangchuangYu/nCov2019>

# THANK YOU!

本次分享材料和代码:

<https://github.com/bsspirit/infect>