计算机数学建模

第一讲 建立数学模型

周毓明

南京大学计算机科学与技术系

课程内容

1. 课程概述
2. 基本概念
3. 建模步骤
4. 模型分类

1. 课程概述

- 成绩计算
- 课程目标
- 课程内容
- 参考资料

成绩计算

- 大作业： 60%
  - ① 选一个题目，进行数学建模，提交符合要求的建模论文 或者
  - ② 写一篇综述论文，对数学建模在计算机科学中的应用进行介绍和分析
- 到课情况 40%

课程目标

- 掌握数学建模的基本方法
  - 简单优化、线性规划、图论模型、统计回归
  - 层次/网络分析法、马尔科夫、微分方程、…
- 能够建立解决实际问题的数学模型
- 能够熟练使用计算机求解数学模型
  - R, Matlab, SPSS, LINGO,…
- 能够撰写符合要求的数学建模论文
  - Word, Latex, Origin, Smartdraw,…

课程内容

- 简单优化专题
  - 第2章：初等模型 + 第3章：简单的优化模型
- 线性规划专题
  - 第4章：数学规划模型+第12章：动态优化模型
- 图论模型专题
  - 附加资料
- 统计回归专题
  - 第10章：统计回归模型
- 层次/网络分析法专题
  - 第8章：离散模型

课程内容

- 马尔科夫模型专题
  - 第12章：马氏链模型
- 微分方程模型专题
  - 第5章：微分方程模型+第6章：差分模型 +第7章：稳定性模型
- 智能计算专题
  - 附加资料
- 傅里叶变换专题
  - 附加资料
- 小波分析专题
  - 附加资料
- 博弈论专题

课程内容

常用模型

按专题组织

- 简单优化
- 数学规划
- 图论模型
- 统计回归
- 层次分析法
- 马尔科夫模型
- 微分方程模型
- 智能计算模型
- 傅里叶变换

基本模型

+

案例分析

+

计算机典型应用

突出应用

课程内容

- 每个专题结构

课程内容

1. 数学概念与模型
2. 实际案例与分析
3. 计算机典型应用

1. 数学概念与模型

① 线性规划
② 整数规划
③ 非线性规划

2. 实际案例与分析

① 奶制品的生产与销售
② 汽车生产与原油采购
③ 接力队选拔

3. 计算机典型应用

软件工程

① 下一版本问题(NRP)
② 测试用例集约简
③ 其他应用…

课程内容

不同方向

拓宽视野

参考资料

- 姜启源, 谢 育出版社, 2011年
- F.R. Giord **mathemat** 启源译, 数学建模.
- M.M. Mee dition. 刘来福, 杉 械工业出版社, 2(
- 赵静, 但琺 , 2008
- 谢金星, 薛 华大学出版社, 2005.

参考资料

- **课程公共邮箱：课件下载+作业下载**

  用户名：cs2018mm@21cn.com

  密码：xian1202MM

- **我的联系方式**

  办公室：计算机系楼703

  电话：89682450

  邮箱：zhouyuming@nju.edu.cn

致谢

本课件使用了如下来源的材料：

- 姜启源, 谢金星, 叶俊. 数学模型电子教案, 修订版. 清华大学
- 浙江大学数学建模基地. 数学模型概论.
- 刘利刚. 数学模型. 浙江大学.
- …

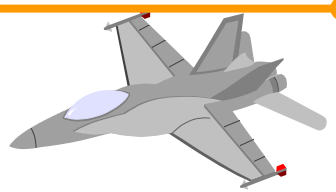
2. 基本概念

- 数学模型
- 数学建模
- 建模示例

数学模型

一千个读者就有一千个哈姆雷特

我们常见的模型



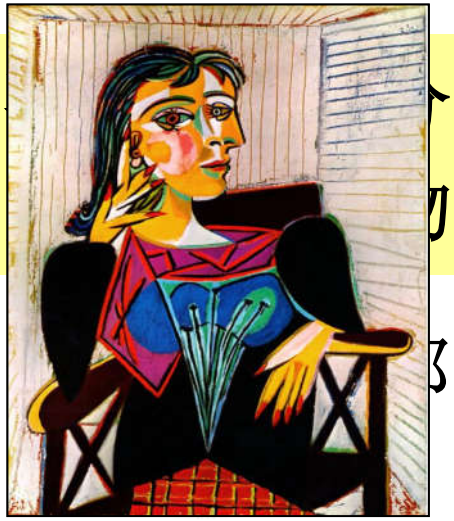
玩具、照片、飞机、火箭模型… …　～实物模型

水箱中的舰艇、风洞中的飞机… …　～物理模型

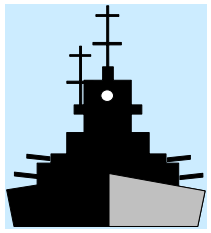
地图、电路图、分子结构图… …　～符号模型

模型是为了一定目的，对客观
进行简缩、抽象、提炼出来的

模型集中反映了原型中人们



数学建模



你碰到过的数学模型——“航行问题”

甲乙两地相距750千米，船从甲到乙顺水航行需30小时，从乙到甲逆水航行需50小时，问船的速度是多少?

用 x 表示船速，y 表示水速，列出方程：

$$\begin{cases}(x+y)\times 30 = 750\\(x-y)\times 50 = 750\end{cases}$$

求解 ⇨ $x=20$, $y=5$

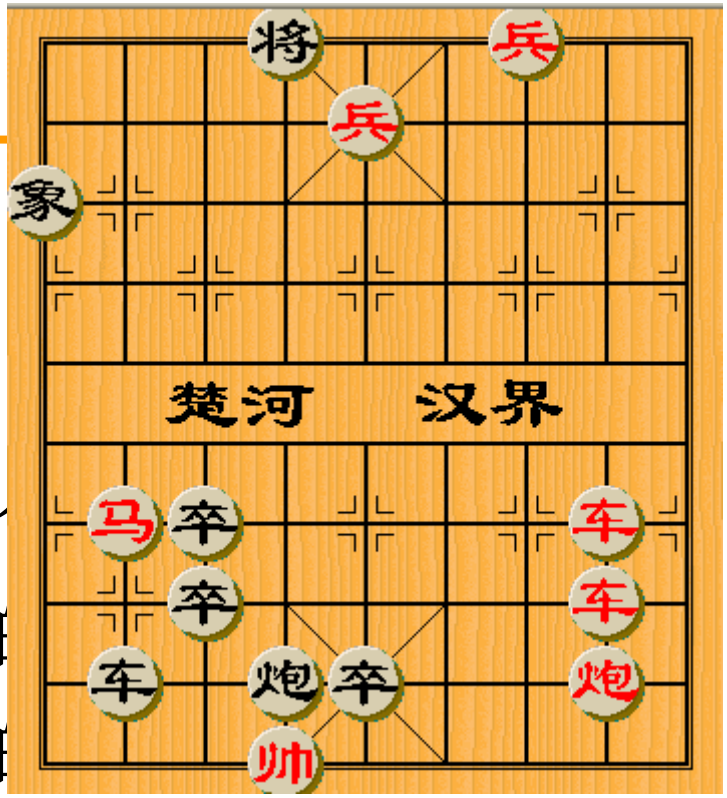
答：船速每小时20千米/小时.

数学建模

航行问题建立数学模型的基本步骤

- 作出简化假设（船速、水速为常数）；
- 用符号表示有关量（x, y表示船速和水速）；
- 用物理定律（匀速运动的距离等于速度乘以时间）列出数学式子（二元一次方程）；
- 求解得到数学解答（x=20, y=5）；
- 回答原问题（船速每小时20千米/小时）。

数学建模



数学模型

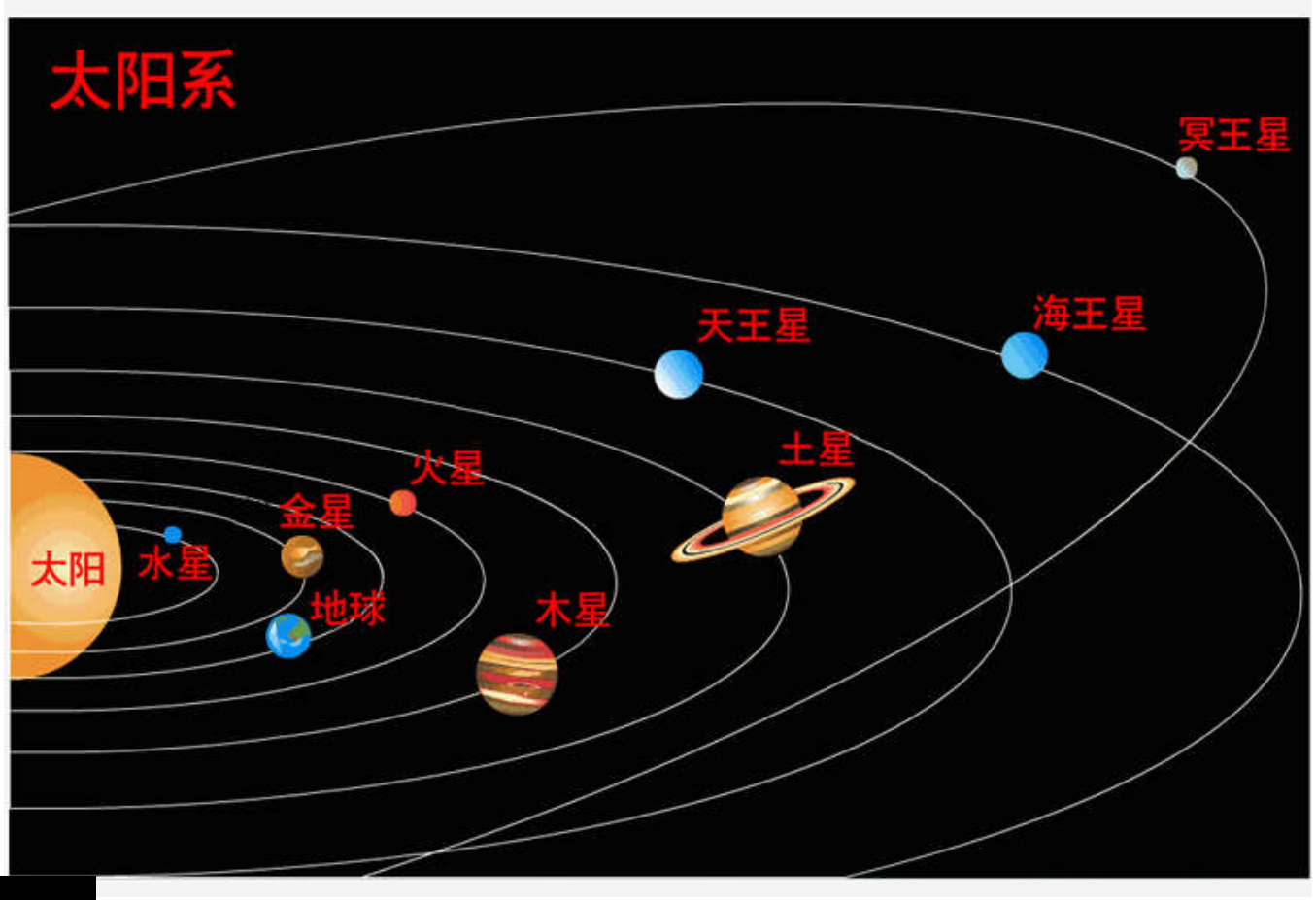
对于一个现实对象，为了一
根据其内在规律，作出必要
运用适当的数学工具，得到

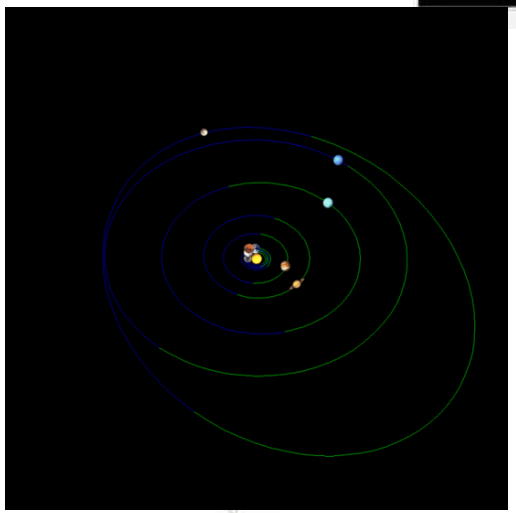
数学建模

建立数学模型的全过程

（包括表述、求解、解释、检验等）

计算机模拟：如兵棋推演





海王星的发现过程

数学建模

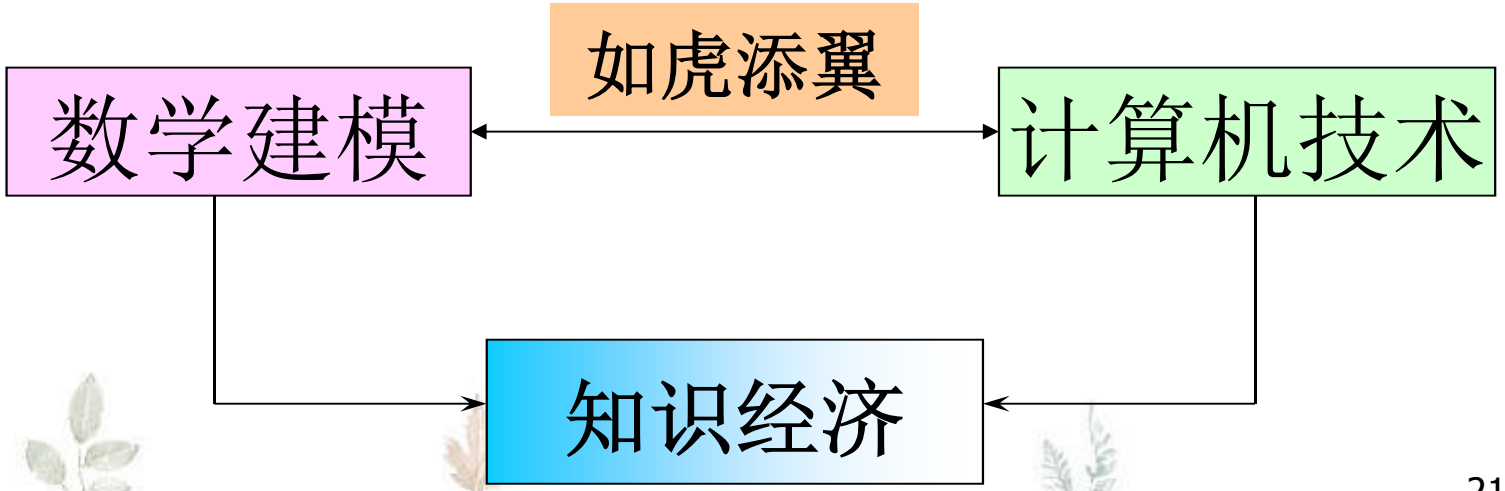
重要意义

- 数学建模
- 在高新技术领域数学建模几乎是必不可少的工具；
- 数学进入一些新领域，为数学建模开辟了许多处女地。

数学建模

数学建模的具体应用

- 分析与设计
- 预报与决策
- 控制与优化
- 规划与管理



建模示例1：椅子问题

(1) 椅子能在不平的地面上放稳吗

问题分析　通常～三只脚着地　　放稳～四只脚着地

模型假设

- 四条腿一样长，椅脚与地面点接触，四脚连线呈正方形；
- 地面高度连续变化，可视为数学上的连续曲面；
- 地面相对平坦，使椅子在任意位置至少三只脚同时着地。

建模示例1：椅子问题

模型构建

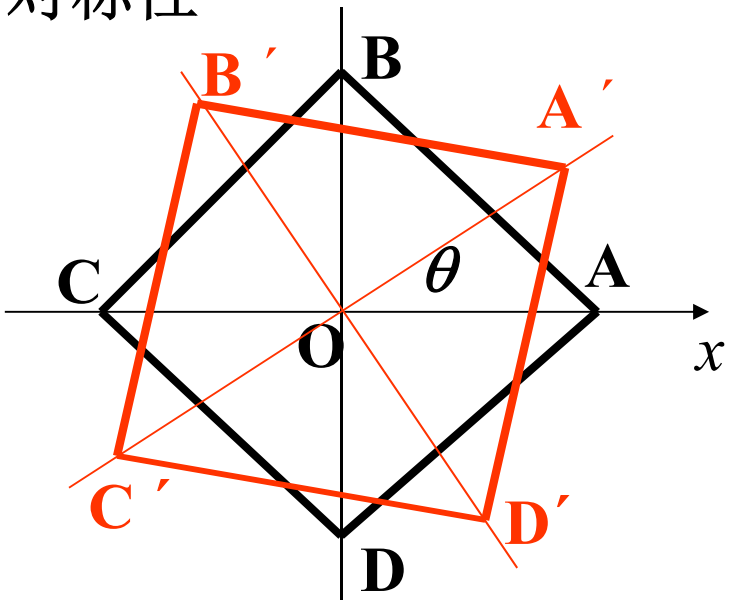
用数学语言把椅子位置和四只脚着地的关系表示出来

• 椅子位置　利用正方形(椅脚连线)的对称性

用θ(对角线与x轴的夹角)表示椅子位置

• 四只脚着地　椅脚与地面距离为零

距离是θ的函数



四个距离
(四只脚)

⇒ 正方形对称性

两个距离

A,C 两脚与地面距离之和 ~ $f(\theta)$

B,D 两脚与地面距离之和 ~ $g(\theta)$

正方形ABCD
绕O点旋转

建模示例1：椅子问题

模型构建

用数学语言把椅子位置和四只脚着地的关系表示出来

地面为连续曲面 ⇨ $f(\theta)$, $g(\theta)$是连续函数

椅子在任意位置至少三只脚着地 ⇨ 对任意θ, $f(\theta)$, $g(\theta)$至少一个为0

数学问题

已知：$f(\theta)$, $g(\theta)$是连续函数；

对任意θ， $f(\theta) \cdot g(\theta)=0$；

且 $g(0)=0$， $f(0) > 0$.

证明：存在θ_0，使$f(\theta_0) = g(\theta_0) = 0$.

建模示例1：椅子问题

模型求解

给出一种简单、粗糙的证明方法

将椅子旋转90^0，对角线AC和BD互换。

由$g(0)=0$，　$f(0)>0$，知$f(\pi/2)=0$，　$g(\pi/2)>0$.

令$h(\theta)=f(\theta)-g(\theta)$，则$h(0)>0$和$h(\pi/2)<0$.

由　f，　g的连续性知　h为连续函数，　据连续函数的基本性质，必存在θ_0，使$h(\theta_0)=0$，　即$f(\theta_0)=g(\theta_0)$.

因为$f(\theta)\cdot g(\theta)=0$，所以$f(\theta_0)=g(\theta_0)=0$.

评注和思考　建模的关键 ~ θ和　$f(\theta)$，　$g(\theta)$的确定

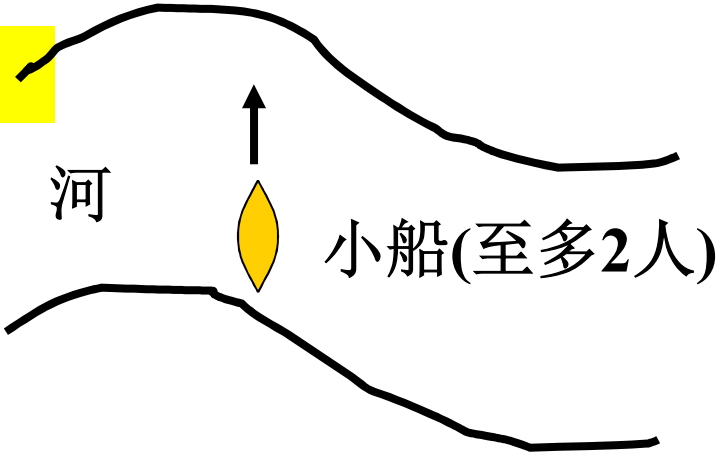
假设条件的本质与非本质　考察四脚呈长方形的椅子

建模示例2：过河问题

（2）商人们怎样安全过河

问题(智力游戏)

随从们密约, 在河的任一岸,
一旦随从的人数比商人多, 就
杀人越货.
但是乘船渡河的方案由商人决定.商
人们怎样才能安全过河?



ΔΔΔ 3名商人

××× 3名随从

问题分析

多步决策过程

决策~ 每一步(此岸到彼岸或彼岸到此岸)船上的人员
要求~在安全的前提下(两岸的随从数不比商人多),经有限步使
全体人员过河.

建模示例2：过河问题

模型构建

x_k~第k次渡河前此岸的商人数　　x_k, y_k=0,1,2,3;

y_k~第k次渡河前此岸的随从数　　k=1,2,… …

$s_k=(x_k, y_k)$~过程的状态　　S ~ 允许状态集合

S={(x, y)| x=0, y=0,1,2,3; x=3, y=0,1,2,3; x=y=1,2}

u_k~第k次渡船上的商人数　　u_k, v_k=0,1,2;

v_k~第k次渡船上的随从数　　k=1,2,… …

$d_k=(u_k, v_k)$~决策　　D={(u, v)| $u+v$=1, 2} ~允许决策集合

$s_{k+1}=s_k+(-1)^k d_k$　　~状态转移律

多步决策问题

求d_k∈D(k=1,2, …n), 使s_k∈S, 并按转移律由 s_1=(3,3)到达 s_{n+1}=(0,0).

建模示例2：过河问题

模型求解

$S=\{(x,y)\mid x=0, y=0,1,2,3;$

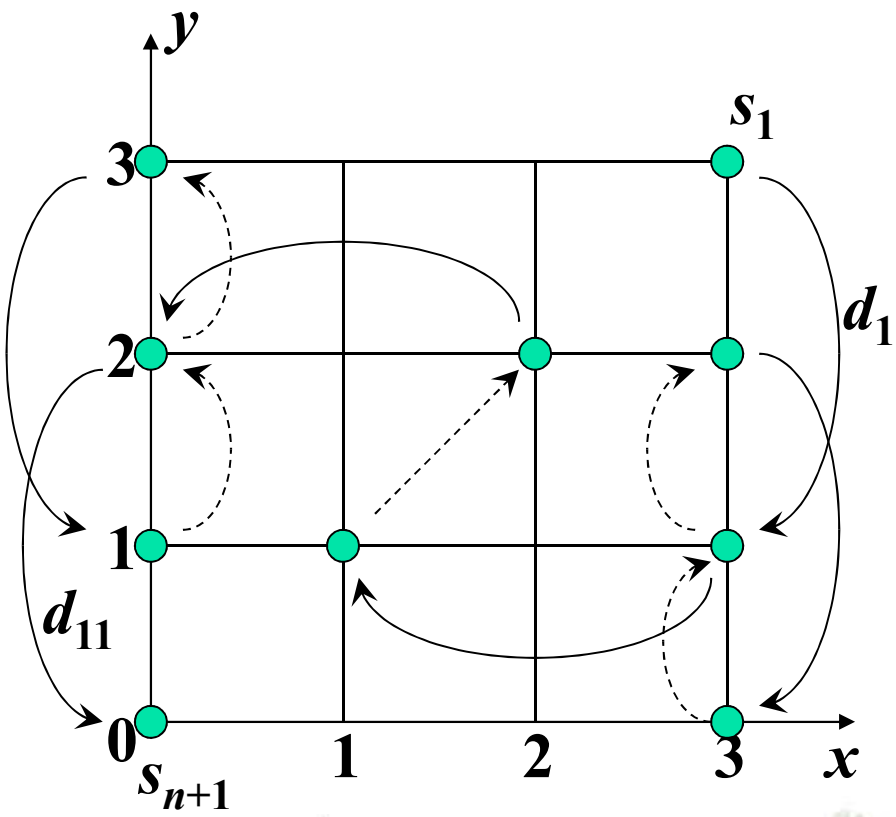
$x=3, y=0,1,2,3;\ x=y=1,2\}$

- 穷举法～ 编程上机
- 图解法

状态$s=(x,y)$～16个格点

允许状态　　～10个 ● 点

允许决策～移动1或2格; k奇, 左下移; k偶,右上移.



$d_1, \ldots, d_{11}$给出安全渡河方案

评注和思考

规格化方法,易于推广

考虑4名商人各带一随从的情况

建模示例3：缺陷定位

(3) 缺陷语句定位

给定一个有bug的程序和一批测试用例。当这些测试用例在程序上执行时，有些测试用例passed，有些测试用例failed。如何仅通过程序结构信息和测试用例的执行信息自动识别出有bug的语句？

mid() { int x,y,z,m;	3, 3, 5					
1: read(x, y, z);						
2: m = z;						
3: if (y<z)						
4: if (x<y)						
5: m = y;						
6: else if (x<z)						
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;						
3: if (y<z)						
4: if (x<y)						
5: m = y;						
6: else if (x<z)						
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3, 3, 5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)						
4: if (x<y)						
5: m = y;						
6: else if (x<z)						
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)						
5: m = y;						
6: else if (x<z)						
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)	•					
5: m = y;						
6: else if (x<z)						
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)	•					
5: m = y;						
6: else if (x<z)	•					
7: m = y;						
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3, 3, 5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)	•					
5: m = y;						
6: else if (x<z)	•					
7: m = y;	•					
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);						
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)	•					
5: m = y;						
6: else if (x<z)	•					
7: m = y;	•					
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);	•(3)					
}						

mid() { int x,y,z,m;	3,3,5					
1: read(x, y, z);	•					
2: m = z;	•					
3: if (y<z)	•					
4: if (x<y)	•					
5: m = y;						
6: else if (x<z)	•					
7: m = y;	•					
8: else						
9: if (x>y)						
10: m = y;						
11: else if (x>z)						
12: m = x;						
13: print(m);	•(3)					
}	P					

mid() { int x,y,z,m;	3,3,5	1,2,3	3,2,1	5,5,5	5,3,4	2,1,3
1: read(x, y, z);	•	•	•	•	•	•
2: m = z;	•	•	•	•	•	•
3: if (y<z)	•	•	•	•	•	•
4: if (x<y)	•	•			•	•
5: m = y;		•				
6: else if (x<z)	•				•	•
7: m = y; // m = x	•					•
8: else			•	•		
9: if (x>y)			•	•		
10: m = y;			•			
11: else if (x>z)				•		
12: m = x;						
13: print(m);	•(3)	•(2)	•(2)	•(5)	•(4)	•(1)
}	P	P	P	P	P	F

建模示例3：缺陷定位

基本方法

- 构造执行频谱矩阵
  - 行: 语句
  - 列: 测试用例
  - 值: 测试用例执行时是否覆盖到了该语句
- 利用各种方法计算语句的可疑程度
  - 被越多的passing测试用例执行，越不可能有bug
  - 被越多的failing测试用例执行，越有可能包含bug
  - ...

建模示例3：缺陷定位

语句可疑度度量示例

$$suspiciousness_S(s) = \frac{failed(s)}{passed(s) + failed(s)}$$

$$suspiciousness_J(s) = \frac{failed(s)}{totalfailed + passed(s)}$$

$$suspiciousness_O(s) = \frac{failed(s)}{\sqrt{totalfailed * (failed(s) + passed(s))}}$$

$$suspiciousness_T(s) = \frac{\%\text{failed}(s)}{\%\text{failed}(s) + \%\text{passed}(s)}$$

建模示例3：缺陷定位

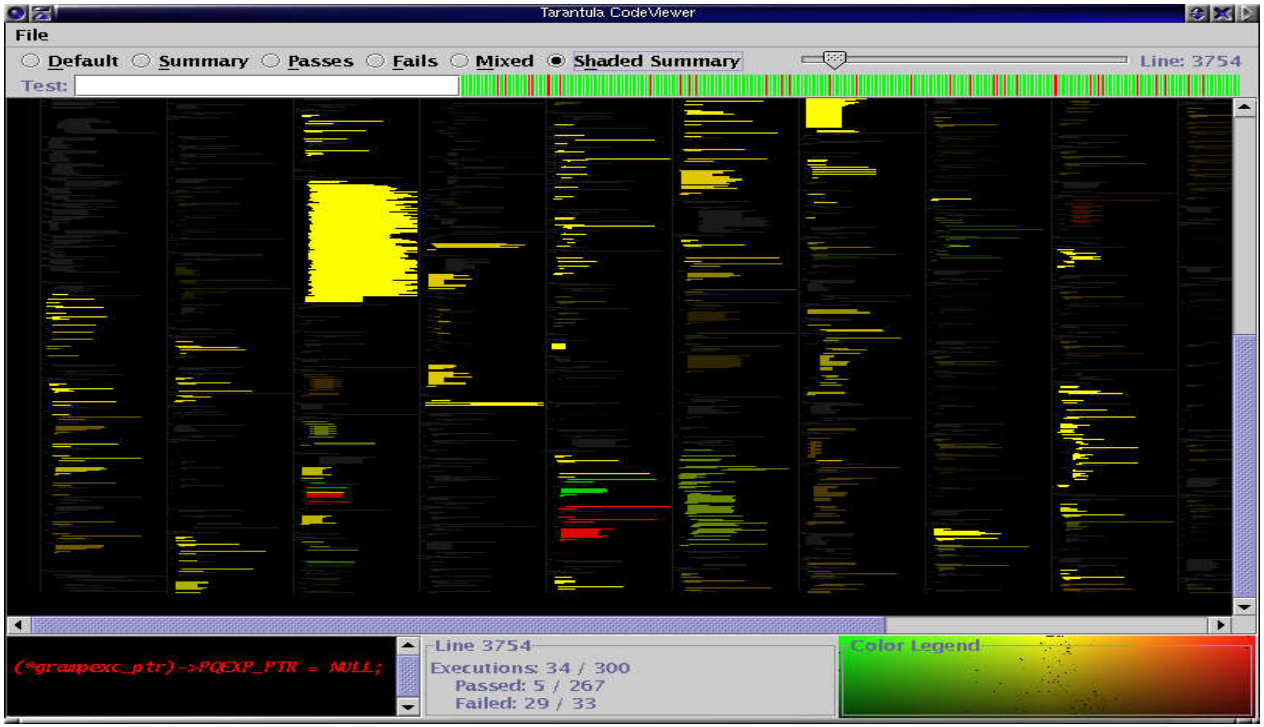
语句可疑度度量示例

	Test Cases								Tarantula			SBI		Jaccard		Ochiai	
	t1	t2	t3	t4	t5	t6	t7	t8									
mid() { int x,y,z,m;	3,3,5	1,2,3	3,2,1	5,5,5	5,3,4	7,5,4	2,1,3	4,3,5	$suspiciousness_T$	confidence	rank	$suspiciousness_S$	rank	$suspiciousness_J$	rank	$suspiciousness_O$	rank
1: read("Enter 3 numbers:",x,y,z);	●	●	●	●	●	●	●	●	0.5	1.0	7	0.25	7	0.25	7	0.5	7
2: m = z;	●	●	●	●	●	●	●	●	0.5	1.0	7	0.25	7	0.25	7	0.5	7
3: if (y<z)	●	●	●	●	●	●	●	●	0.5	1.0	7	0.25	7	0.25	7	0.5	7
4: if (x<y)	●	●			●		●	●	0.67	1.0	3	0.4	3	0.4	3	0.63	3
5: m = y;		●							0.0	0.17	12	0.0	13	0.0	13	0.0	13
6: else if (x<z)	●				●		●	●	0.75	1.0	2	0.5	2	0.5	2	0.71	2
7: m = y; // *** bug ***	●						●	●	0.86	1.0	1	0.67	1	0.67	1	0.82	1
8: else			●	●		●			0.0	0.5	9	0.0	13	0.0	13	0.0	13
9: if (x>y)			●	●		●			0.0	0.5	9	0.0	13	0.0	13	0.0	13
10: m = y;			●			●			0.0	0.33	10	0.0	13	0.0	13	0.0	13
11: else if (x>z)				●					0.0	0.17	12	0.0	13	0.0	13	0.0	13
12: m = x;									0.0	0.0	13	0.0	13	0.0	13	0.0	13
13: print("Middle number is:",m);	●	●	●	●	●	●	●	●	0.5	1.0	7	0.25	7	0.25	7	0.5	7
} Pass/Fail Status	P	P	P	P	P	P	F	F									

建模示例3：缺陷定位

Mary Jean Harrold

工具支持



建模示例3：缺陷定位

mid() { int x,y,z,m;	3,3,5	1,2,3	3,2,1	5,5,5	5,3,4	2,1,3
1: read(x,y,z);	●	●	●	●	●	●
2: m=z;	●	●	●	●	●	●
3: if(y<z)	●	●	●	●	●	●
4: if(x<y)		●				
5: m=y;		●				
6: elseif(x<z)	●				●	●
7: m=y;	●					●
8: else	●		●	●		
9: if(x>y)			●			
10: m=y;			●			
11: elseif(x>z)						
12: m=x;						
13: print(m);	●	●	●	●	●	●
}	P	P	P	P	P	F

建模示例3：缺陷定位

实验结果

R. Santelices et al.
ICSE 2009

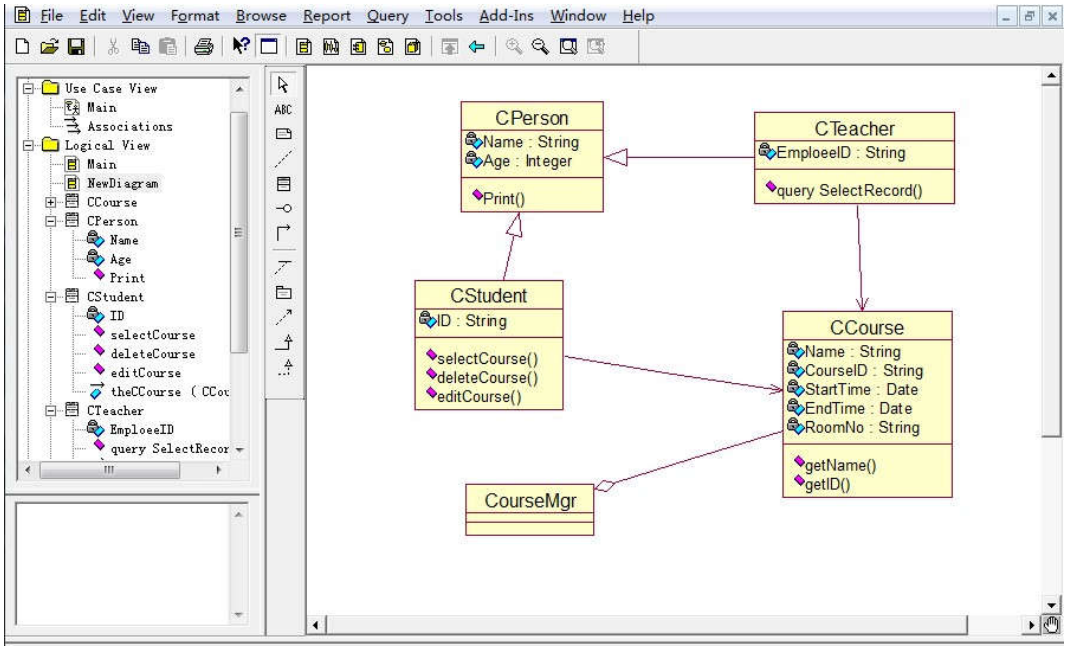
subject	LOC	tests	faults
Tcas	131	1608	10
Tot_info	283	1052	10
Schedule1	290	2650	9
Schedule2	317	2710	7
Print_tokens1	478	4130	5
Print_tokens2	410	4115	10
NanoXML v1	3497	214	7
NanoXML v2	4009	214	7
NanoXML v3	4608	216	8
NanoXML v5	4782	216	7
XML-security v1	21613	92	7
XML-security v2	22318	94	7
XML-security v3	19895	84	2
JABA	37966	677	11
TOTAL FAULTS			107

For all 107 faults:

measure	statement	branch	du-pair
average cost	11.49%	10.24%	9.02%
standard dev.	*16.25%*	*15.40%*	*12.04%*

建模示例4：规模预测

(4) 软件系统规模(SLOC)预测



背景介绍

软件系统的分析做好后，经理想知道系统的规模有多大，以便

(1) 估算开发工作量

(2) 进行进度控制

建模示例4：规模预测

问题分析

- 给定系统的UML类图
- 可以从UML类图上收集如下结构度量信息

Measure name	Measure definition
Number of Classes(NC)	类图所有类的数目。
Number of Attributes(NA)	类图中所有属性的数目。
Number of Methods(NM)	类图中所有方法的数目。
Number of Associations(NAssoc)	类图中关联关系的数量。
Number of Aggregation(NAgg)	类图中聚合关系的数量。
Number of Dependencies(NDep)	类图中依赖关系的个数。
Number of Generalizations(NGen)	类图中继承关系的个数。
Number of Generalizations Hierarchies(NGenH)	类图中泛化体系的数量。
Max Dit(Depth of Inheritance Tree)	泛化体系的最大深度。
Number of Aggregations Hierarchies(NAggH)	聚合体系的个数。
Max HAgg	聚合结构的最大深度。

建模示例4：规模预测

模型假设

- 系统的规模用SLOC来计算
- UML类图上的结构度量与SLOC呈线性关系

$$y_i = a_0 + a_1 x_{i1} + a_2 x_{i2} + a_3 x_{i3} + \cdots + a_k x_{ik} + \epsilon_i$$

建模示例4：规模预测

模型构建与求解

- 从**sourceforge.net**下载**100**个开源系统源码
- 利用工具逆向恢复**UML**类图
- 收集结构度量信息
- 建立线性回归模型**(Weka)**

$$SLOC = 12.6869 \times NA + 6.5028 \times NM - 11.072 \times NGen - 467.5534$$

建模示例4：规模预测

模型评价

- 10-fold交叉验证

  将数据集分成10份，轮流将其中9份做训练1份做测试，然后将所有的测试结果合并在一起评价模型的性能

$$MRE_i = \frac{|y_i - \hat{y}_i|}{y_i}$$

$$MMRE = \frac{1}{n}\sum_{i=1}^{i=n} MRE_i$$

$$\mathrm{Pred}(q) = \frac{\| \{i \mid MRE_i \le q\} \|}{n}$$

MMRE = 0.228　　**Pred(0.25) = 0.60**

按软件工程标准，相当准确

建模示例4：规模预测

模型检验

跨版本验证

- 从sourceforge.net下载100个新版本的源码
- 逆向恢复UML类图，得到新版本上的结构度量
- 利用老版本上得到的方程进行预测，与实际结果比较

$$SLOC = 12.6869 \times NA + 6.5028 \times NM - 11.072 \times NGen - 467.5534$$

MMRE = 0.207 **Pred(0.25) = 0.65**

按软件工程标准，相当准确

3. 建模步骤

建模步骤

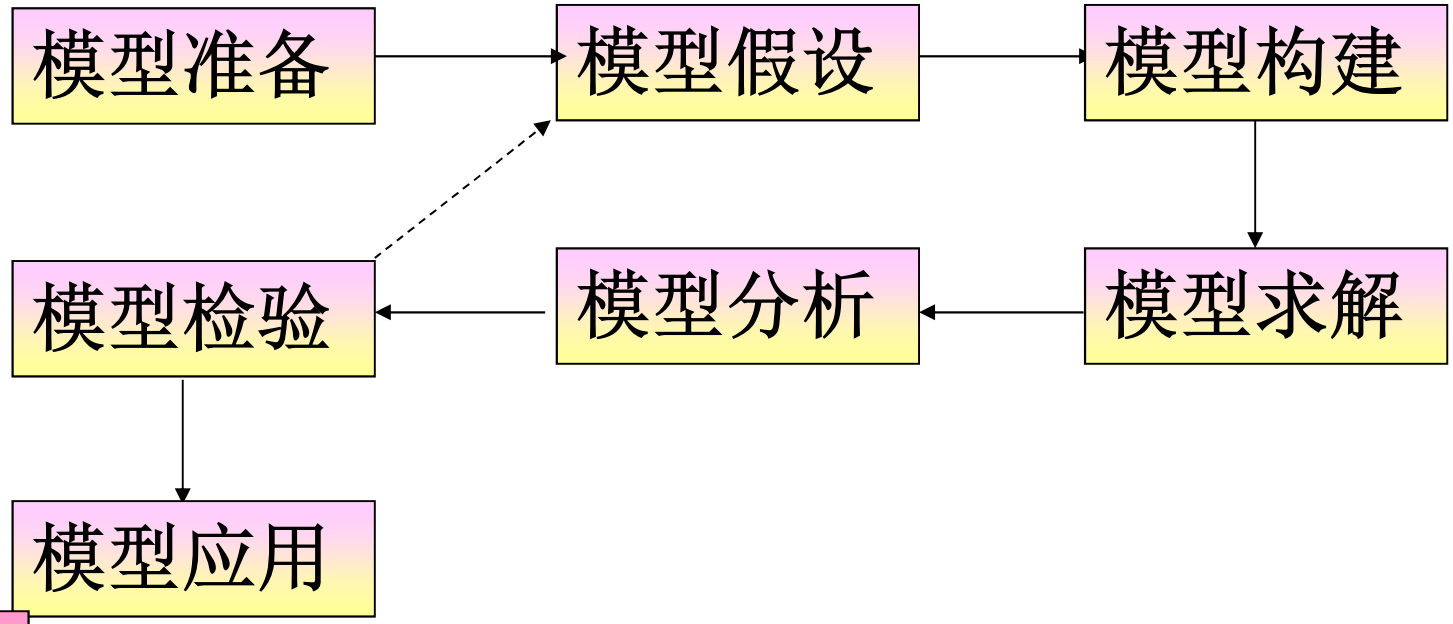
数学建模的基本方法

- 机理分析：根据对客观事物特性的认识，找出反映内部机理的数量规律
- 测试分析：将对象看作“黑箱”,通过对量测数据的统计分析，找出与数据拟合最好的模型
- 二者结合：用机理分析建立模型结构，用测试分析确定模型参数

机理分析没有统一的方法，主要通过实例研究(Case Studies)来学习。以下建模主要指机理分析。

建模步骤

数学建模的一般步骤



模型准备

了解实际背景　　明确建模目的

搜集有关信息　　掌握对象特征

形成一个比较清晰的‘问题’

建模步骤

数学建模的一般步骤

模型假设

针对问题特点和建模目的

作出合理的、简化的假设

在合理与简化之间作出折中

模型构建

用数学的语言、符号描述问题

发挥想像力　　使用类比法

尽量采用简单的数学工具

建模步骤

数学建模的一般步骤

模型求解　各种数学方法、软件和计算机技术

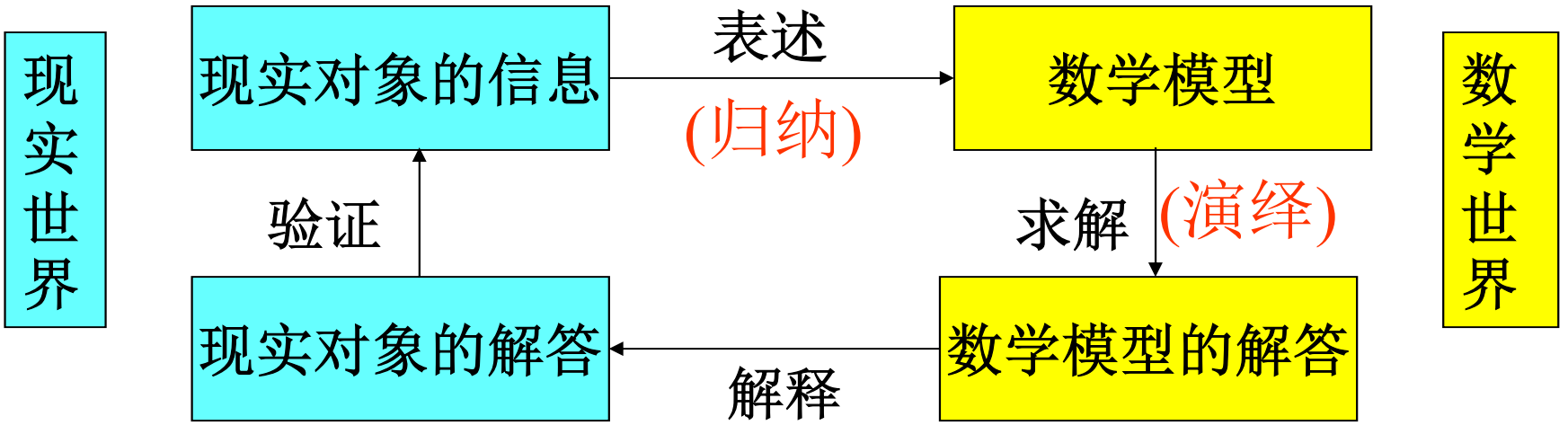
模型分析　如结果的误差分析、统计分析、模型对数据的稳定性分析

模型检验　与实际现象、数据比较，检验模型的合理性、适用性

模型应用

建模步骤

数学建模的全过程



表述	根据建模目的和信息将实际问题“翻译”成数学问题
求解	选择适当的数学方法求得数学模型的解答
解释	将数学语言表述的解答“翻译”回实际对象
验证	用现实对象的信息检验得到的解答

实践 ⇨ 理论 ⇨ 实践

4. 模型分类

模型分类

数学模型的特点

模型的逼真性和可行性　　模型的非预制性

模型的渐进性　　模型的条理性

模型的强健性　　模型的技艺性

模型的可转移性　　模型的局限性

模型分类

数学模型的分类

应用领域	人口、交通、经济、生态 ……
数学方法	初等数学、微分方程、规划、统计 ……
表现特性	确定和随机　　静态和动态 离散和连续　　线性和非线性
建模目的	描述、优化、预报、决策 ……
了解程度	白箱　　灰箱　　黑箱

模型分类

怎样学习数学建模

数学建模与其说是一门技术，不如说是一门艺术

技术大致有章可循　　艺术无法归纳成普遍适用的准则

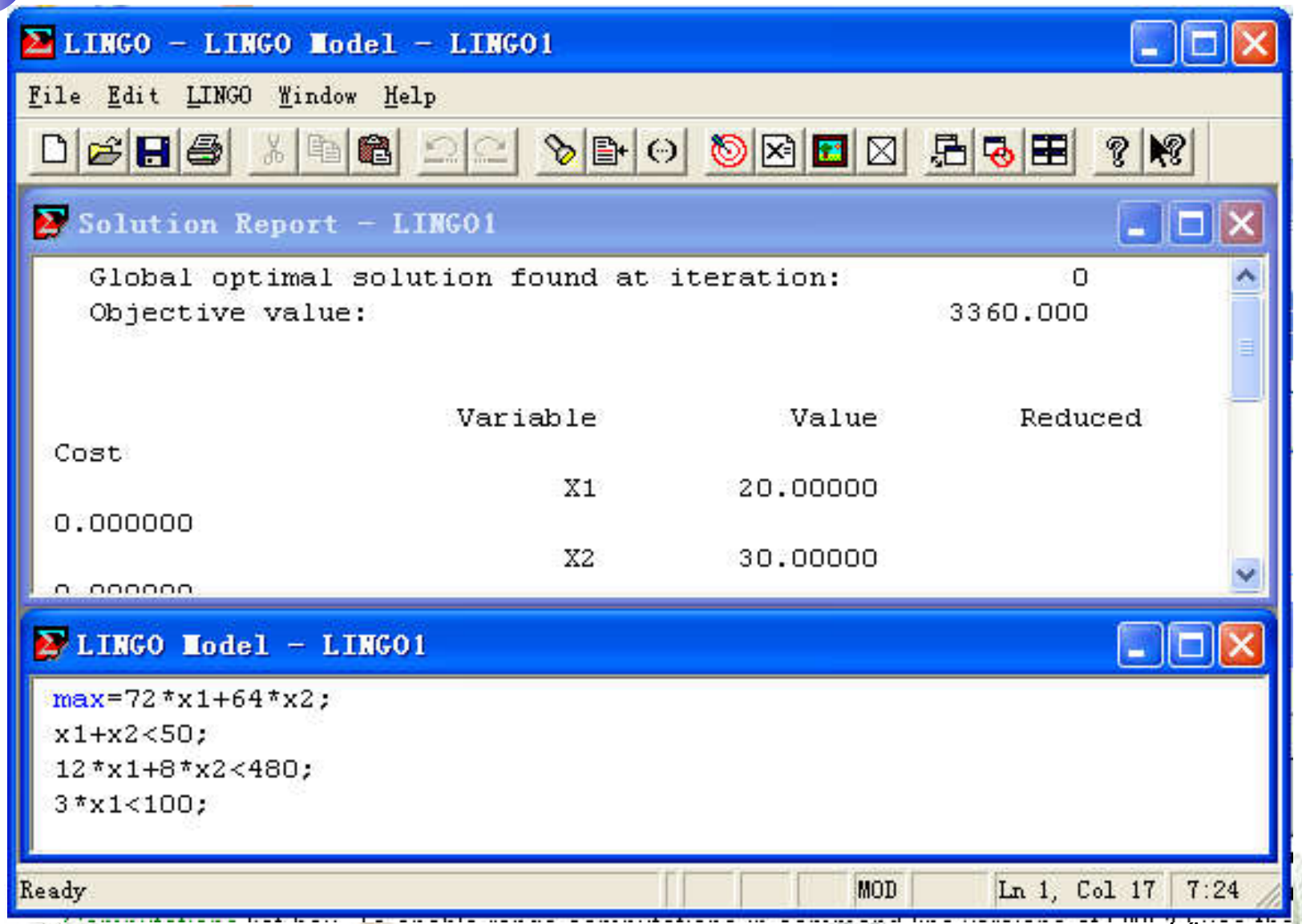
想像力　　洞察力　　判断力

- 学习、分析、评价、改进别人作过的模型
- 亲自动手，认真作几个实际题目

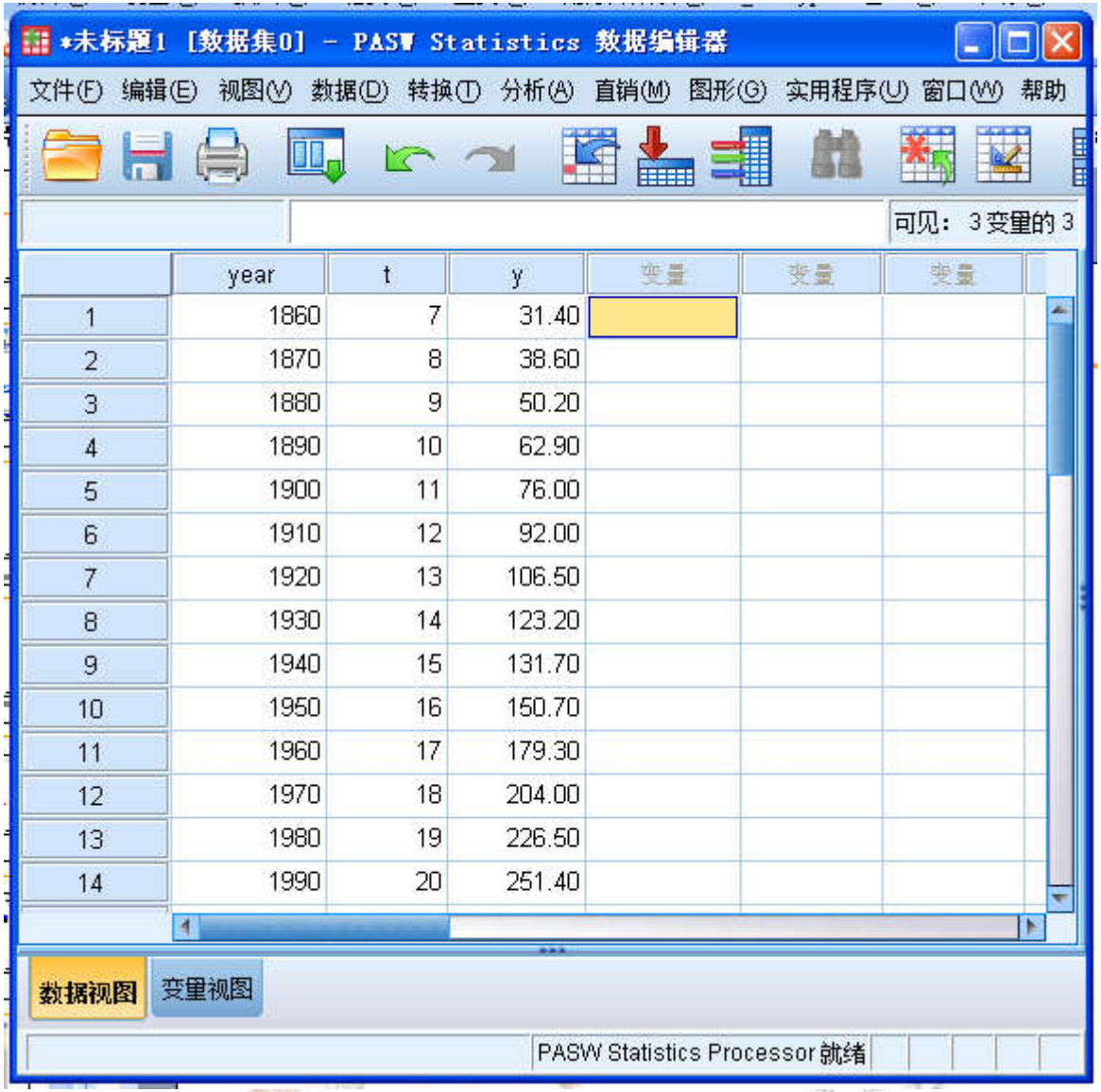



附录：工具

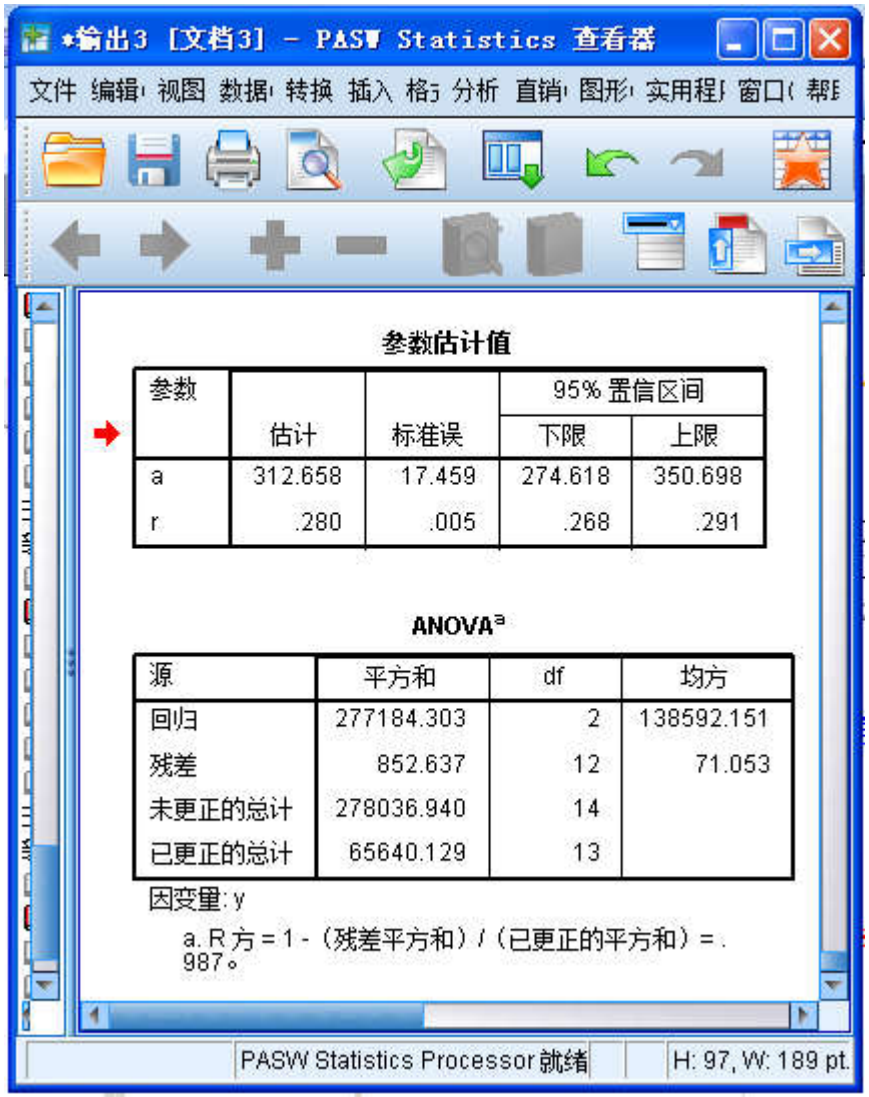
工具：LINGO


LINGO - LINGO Model - LINGO1
File Edit LINGO Window Help
Solution Report - LINGO1
Global optimal solution found at iteration: 0
Objective value: 3360.000
Variable Value Reduced
Cost
X1 20.00000
0.000000
X2 30.00000
LINGO Model - LINGO1
max=72*x1+64*x2;
x1+x2<50;
12*x1+8*x2<480;
3*x1<100;
Ready
MOD
Ln 1, Col 17
7:24

工具：SPSS (PASW)


*未标题1 [数据集0] - PASW Statistics 数据编辑器
文件(F) 编辑(E) 视图(V) 数据(D) 转换(T) 分析(A) 直销(M) 图形(G) 实用程序(U) 窗口(W) 帮助
可见：3 变量的 3
year t y 变量 变量 变量
1 1860 7 31.40
2 1870 8 38.60
3 1880 9 50.20
4 1890 10 62.90
5 1900 11 76.00
6 1910 12 92.00
7 1920 13 106.50
8 1930 14 123.20
9 1940 15 131.70
10 1950 16 150.70
11 1960 17 179.30
12 1970 18 204.00
13 1980 19 226.50
14 1990 20 251.40
数据视图 变量视图
PASW Statistics Processor 就绪

工具：SPSS



$$x(t)=\frac{x_m}{1+(\frac{x_m}{x_0}-1)e^{-rt}}$$

非线性回归求解

$x_m = 312.658$

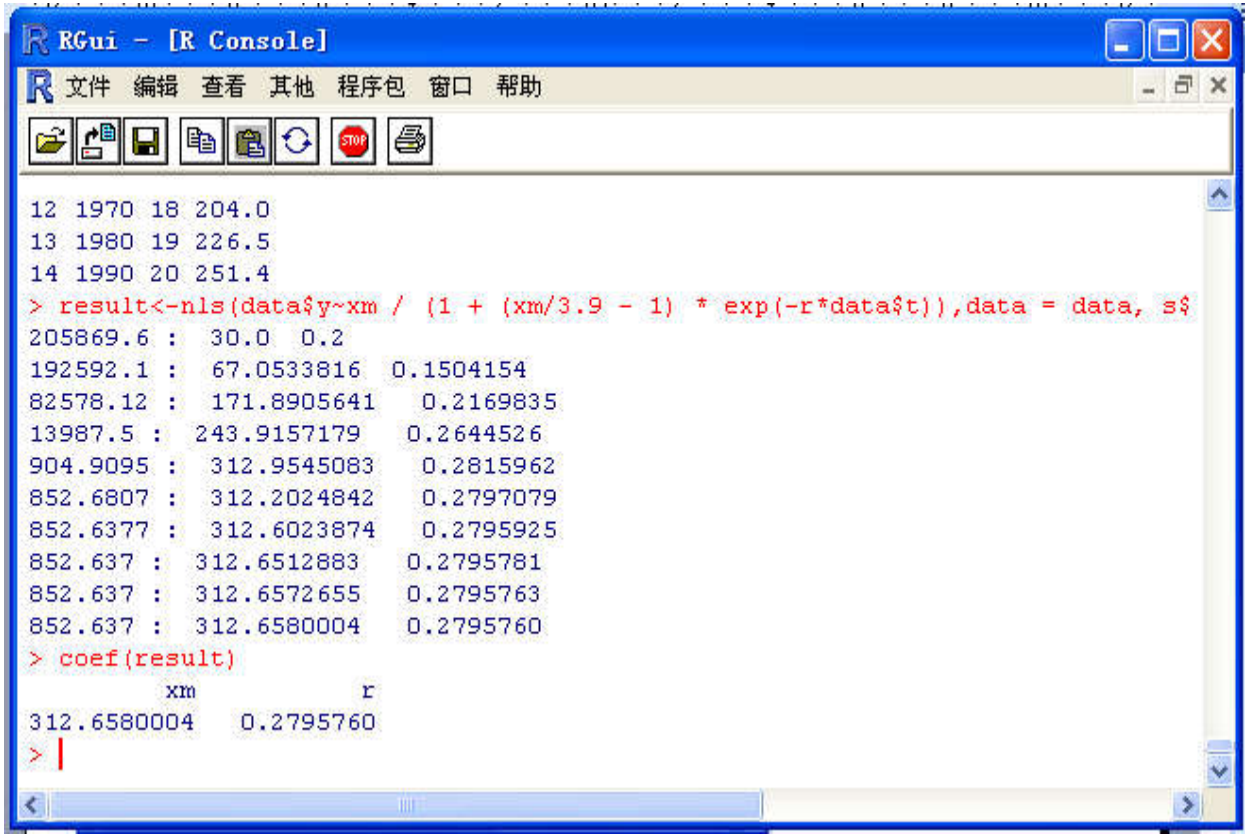
$r = 0.280$

线性回归方法求解

x_m=392.0886

r=0.2557

工具：R



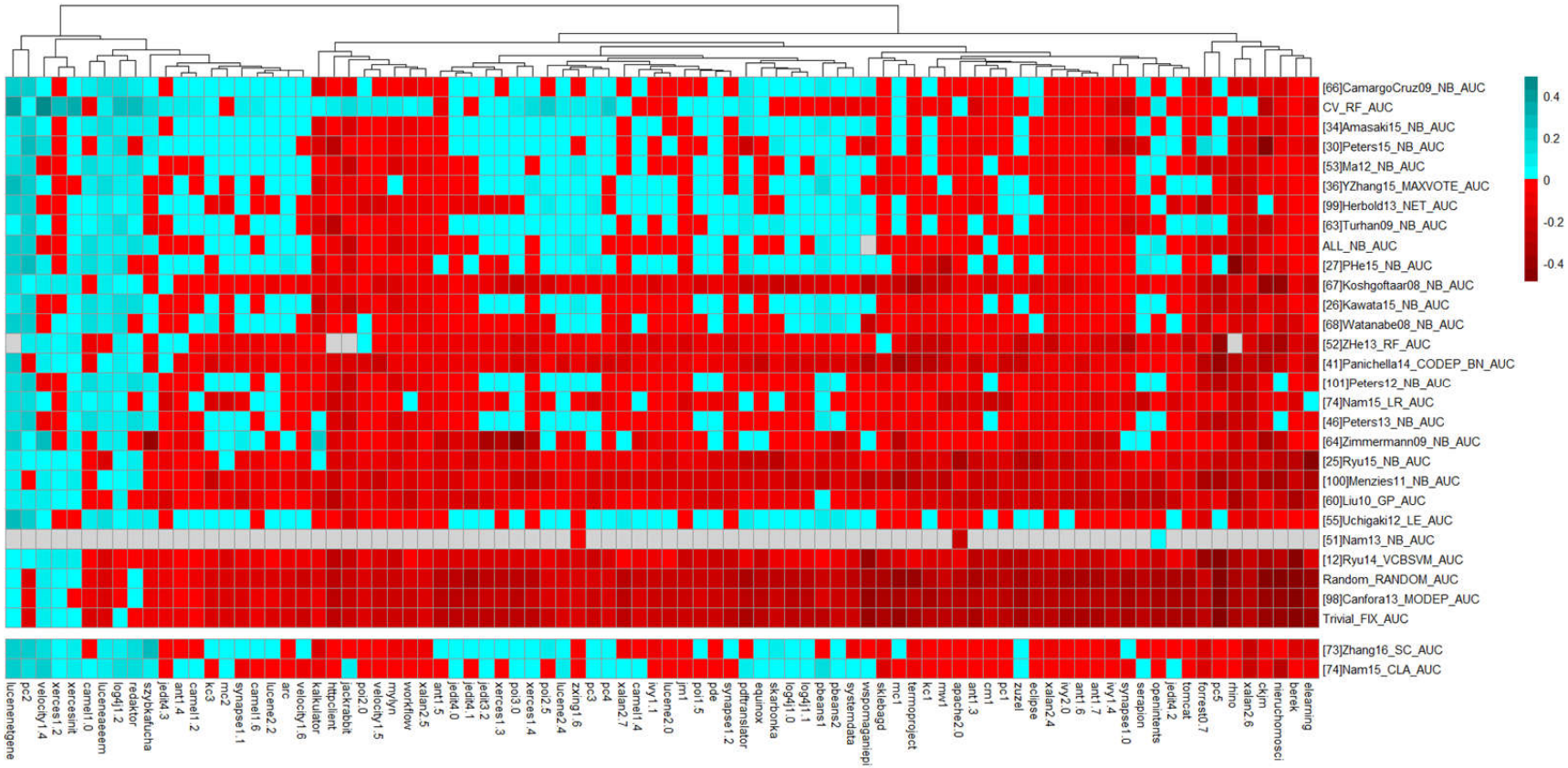
$$x(t) = \frac{x_m}{1 + (\frac{x_m}{x_0} - 1)e^{-rt}}$$

非线性回归求解

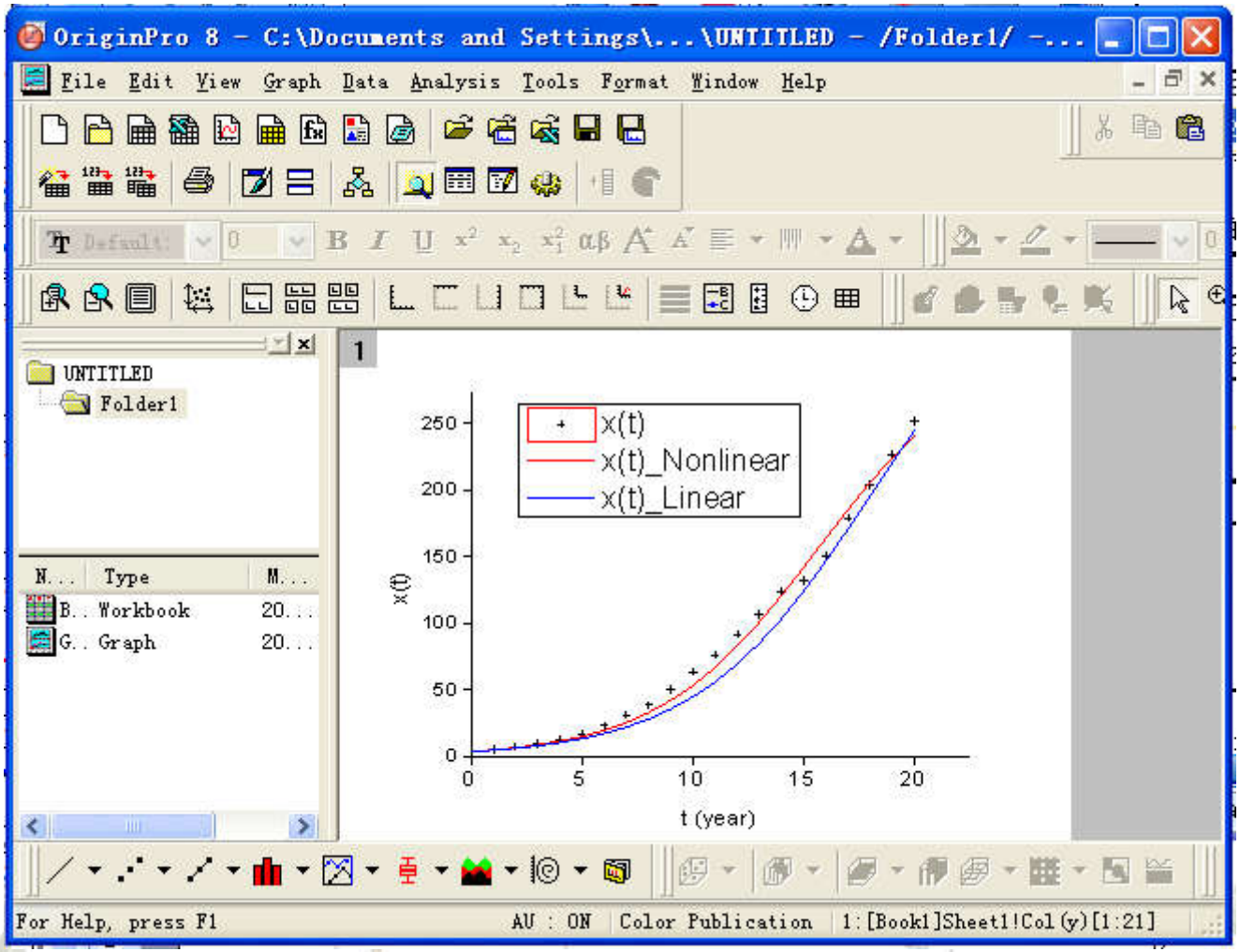
$x_m = 312.658$

$r = 0.280$

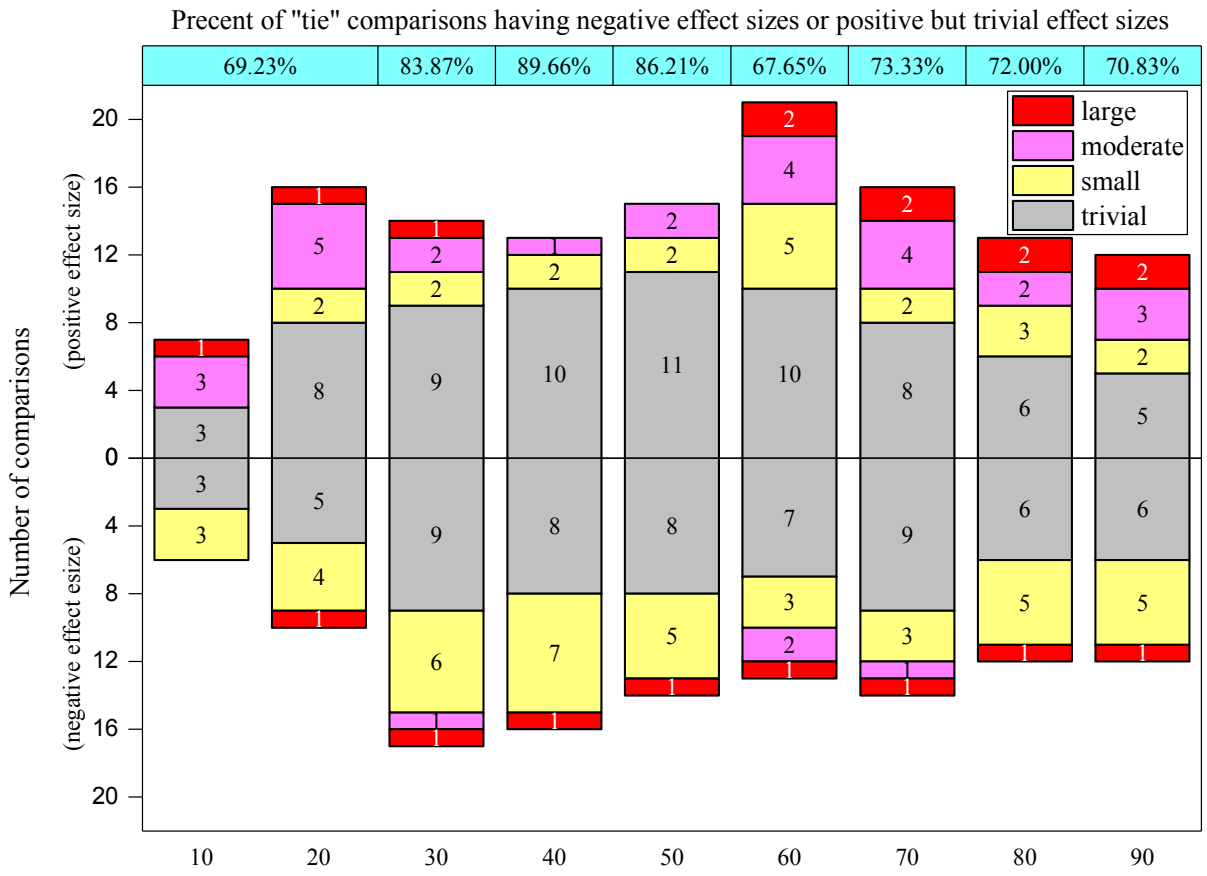
工具：R


[66]CamargoCruz09_NB_AUC
CV_RF_AUC
[34]Amasaki15_NB_AUC
[30]Peters15_NB_AUC
[53]Ma12_NB_AUC
[36]YZhang15_MAXVOTE_AUC
[99]Herbold13_NET_AUC
[63]Turhan09_NB_AUC
ALL_NB_AUC
[27]PHe15_NB_AUC
[67]Koshgoftaar08_NB_AUC
[26]Kawata15_NB_AUC
[68]Watanabe08_NB_AUC
[52]ZHe13_RF_AUC
[41]Panichella14_CODEP_BN_AUC
[101]Peters12_NB_AUC
[74]Nam15_LR_AUC
[46]Peters13_NB_AUC
[64]Zimmermann09_NB_AUC
[25]Ryu15_NB_AUC
[100]Menzies11_NB_AUC
[60]Liu10_GP_AUC
[55]Uchigaki12_LE_AUC
[51]Nam13_NB_AUC
[12]Ryu14_VCBSVM_AUC
Random_RANDOM_AUC
[98]Canfora13_MODEP_AUC
Trivial_FIX_AUC
[73]Zhang16_SC_AUC
[74]Nam15_CLA_AUC
0.4
0.2
0
-0.2
-0.4
lucenenetgene
pc2
velocity1.4
xerces1.2
xercesint
camel1.0
luceneaeeem
log4j1.2
redaktor
szybkafucha
jedit4.3
ant1.4
camel1.2
kc3
mc2
synapse1.1
camel1.6
lucene2.2
arc
velocity1.6
kalkulator
httpclient
jackrabbit
poi2.0
velocity1.5
rmylyn
workflow
xalan2.5
ant1.5
jedit4.0
jedit4.1
jedit3.2
xerces1.3
poi3.0
xerces1.4
poi2.5
lucene2.4
zxing1.6
pc3
pc4
xalan2.7
camel1.4
ivy1.1
lucene2.0
jm1
poi1.5
pde
synapse1.2
pdftranslator
equinox
skarbonka
log4j1.0
log4j1.1
pbeans2
pbeans1
systemdata
wspomaganiepi
sklebagd
mc1
termoproject
kc1
mwv1
apache2.0
ant1.3
cm1
pc1
zuzel
eclipse
xalan2.4
ivy2.0
ant1.6
ant1.7
ivy1.4
synapse1.0
serapion
openintents
jedit4.2
tomcat
forrest0.7
pc5
rhino
xalan2.6
ckjm
mieruchomosci
berek
elearning

工具：Origin


OriginPro 8 - C:\Documents and Settings\...\UNTITLED - /Folder1/ -...
File Edit View Graph Data Analysis Tools Format Window Help
UNTITLED
Folder1
N... Type M...
B.. Workbook 20...
G.. Graph 20...
1
x(t)
x(t)_Nonlinear
x(t)_Linear
250
200
150
100
50
0
x(t)
0
5
10
15
20
t (year)
For Help, press F1
AU : ON
Color Publication
1:[Book1]Sheet1!Col(y)[1:21]

工具：Origin


Precent of "tie" comparisons having negative effect sizes or positive but trivial effect sizes
69.23%
83.87%
89.66%
86.21%
67.65%
73.33%
72.00%
70.83%
large
moderate
small
trivial
Number of comparisons
(positive effect size)
(negative effect esize)
20
16
12
8
4
0
4
8
12
16
20
10
20
30
40
50
60
70
80
90

小结

- 什么是数学模型？
- 什么是数学建模?
- 怎样进行数学建模?
- 数学模型的类型？

思考题

1. Google是怎样实现准确的信息搜索的？
2. 如何知道软件系统中还有多少个bug?
3. 开源项目的存活期是多长？

Thanks for your time and attention!