

《统计分析与R语言》

基本统计分析



内容回顾

- ❧ 图形的创建和保存
- ❧ 自定义符号、线条、颜色和坐标轴
- ❧ 标注文本和标题
- ❧ 控制图形维度
- ❧ 组合多个图





内容回顾

- ❧ 条形图、箱线图和点图
- ❧ 饼图和扇形图
- ❧ 直方图与和密度图





内容

- 描述性统计分析
- 频数表和列联表
- 相关系数和协方差





描述性统计分析

```
1 > vars <- c("mpg", "hp", "wt")
2 > head(mtcars[vars])
3           mpg   hp  wt
4 Mazda RX4    21.0 110 2.62
5 Mazda RX4 Wag 21.0 110 2.88
6 Datsun 710    22.8  93 2.32
7 Hornet 4 Drive 21.4 110 3.21
8 Hornet Sportabout 18.7 175 3.44
9 Valiant      18.1 105 3.46
```




描述性统计分析

通过summary()计算描述性统计量

```
1 > summary(mtcars[vars])
2      mpg      hp      wt
3  Min.   :10.4   Min.   : 52.0   Min.   :1.51
4  1st Qu.:15.4   1st Qu.: 96.5   1st Qu.:2.58
5  Median :19.2   Median :123.0   Median :3.33
6  Mean   :20.1   Mean   :146.7   Mean   :3.22
7  3rd Qu.:22.8   3rd Qu.:180.0   3rd Qu.:3.61
8  Max.   :33.9   Max.   :335.0   Max.   :5.42
```



描述性统计分析

通过sapply()计算描述性统计量

```
1 > mystats <- function(x, na.omit=FALSE){
2     if (na.omit)
3         x <- x[!is.na(x)]
4     m <- mean(x)
5     n <- length(x)
6     s <- sd(x)
7     skew <- sum((x-m)^3/s^3)/n
8     kurt <- sum((x-m)^4/s^4)/n - 3
9     return(c(n=n, mean=m, stdev=s, skew=skew, kurtosis=kurt))
10
11 }
12
13 > sapply(mtcars[vars], mystats)
14      mpg      hp      wt
15 n      32.000  32.000 32.0000
16 mean   20.091 146.688  3.2172
17 stdev   6.027  68.563  0.9785
18 skew   0.611   0.726  0.4231
19 kurtosis -0.373 -0.136 -0.0227
```



描述性统计分析

扩展

通过Hmisc包中的describe()函数
计量

```
1 > library(Hmisc)
2 > describe(mtcars[vars])
3
4 3 Variables      32 Observations
5 -----
6 mpg
7 n missing  unique   Mean   .05   .10   .25   .50   .75   .90   .95
8 32      0      25  20.09 12.00 14.34 15.43 19.20 22.80 30.09 31.30
9
10 lowest : 10.4 13.3 14.3 14.7 15.0, highest: 26.0 27.3 30.4 32.4 33.9
11 -----
12 hp
13 n missing  unique   Mean   .05   .10   .2   .50   .75   .90   .95
14 32      0      22 146.7  63.65 66.00 96.50 123.00 180.00 243.50 253.55
15
16 lowest : 52  62  65  66  91, highest: 215 230 245 264 335
17 -----
18 wt
19 n missing  unique   Mean   .05   .10   .25   .50   .75   .90   .95
20 32      0      29  3.217 1.736 1.956 2.581 3.325 3.610 4.048 5.293
21
22 lowest : 1.513 1.615 1.835 1.935 2.140, highest: 3.845 4.070 5.250 5.345
23 5.424
24 -----
```




描述性统计分析

扩展

通过pastecs包中的stat.desc()函数计算统计量

```
1 > library(pastecs)
2 > stat.desc(mtcars[vars])
```

	mpg	hp	wt
4 nbr.val	32.00	32.000	32.000
5 nbr.null	0.00	0.000	0.000
6 nbr.na	0.00	0.000	0.000
7 min	10.40	52.000	1.513
8 max	33.90	335.000	5.424
9 range	23.50	283.000	3.911
10 sum	642.90	4694.000	102.952
11 median	19.20	123.000	3.325
12 mean	20.09	146.688	3.217
13 SE.mean	1.07	12.120	0.173
14 CI.mean.0.95	2.17	24.720	0.353
15 var	36.32	4700.867	0.957
16 std.dev	6.03	68.563	0.978
17 coef.var	0.30	0.467	0.304



描述性统计

```
1 library(psych)
2
3 Attaching package: 'psych'
4 The following object(s) are masked from package:Hmisc :
5 describe
6
7 > describe(mtcars[vars])
8      var  n  mean   sd median trimmed  mad   min   max
9 mpg    1 32  20.09  6.03  19.20   19.70  5.41 10.40 33.90
10 hp     2 32 146.69 68.56 123.00  141.19 77.10 52.00 335.00
11 wt     3 32   3.22  0.98   3.33    3.15  0.77  1.51   5.42
12      range skew kurtosis    se
13 mpg  23.50 0.61   -0.37  1.07
14 hp  283.00 0.73   -0.14 12.12
15 wt   3.91 0.42   -0.02  0.17
```

扩展

通过psych包中的describe()计算描述性统计量



描述性统计分析

- ❧ 分组计算描述性统计量
- ❧ 使用aggregate()分组获取描述性统计量

```
1 > aggregate(mtcars[vars], by=list(am=mtcars$am), mean)
2   am mpg  hp  wt
3 1  0 17.1 160 3.77
4 2  1 24.4 127 2.41
5 > aggregate(mtcars[vars], by=list(am=mtcars$am), sd)
6   am mpg  hp  wt
7 1  0 3.83 53.9 0.777
8 2  1 6.17 84.1 0.617
```



描述性统计分析

🌀 分组计算描述性统计量

```
by(data, INDICES, FUN)
```

🌀 使用by()分组计算描述性统计量

```
dstats <- function(x){sapply(x, mystats)}  
myvars <- c("mpg", "hp", "wt")  
by(mtcars[myvars], mtcars$am, dstats)
```




描述性统计分析

- 🌀 分组计算描述性统计量
- 🌀 使用describeBy()分组计算描述性统计量

```
#扩展 使用psych包中的describeBy()分组计算描述统计量  
library(psych)  
myvars <- c("mpg", "hp", "wt")  
describeBy(mtcars[myvars], list(am=mtcars$am))
```




频数表和列联表

```
> library(vcd)
```

载入需要的程辑包: **grid**

```
> head(Arthritis)
```

	ID	Treatment	Sex	Age	Improved
1	57	Treated	Male	27	Some
2	46	Treated	Male	29	None
3	77	Treated	Male	30	None
4	17	Treated	Male	32	Marked
5	36	Treated	Male	46	Marked
6	23	Treated	Male	58	Marked



频数表和列联表

生成频数表

表7-1 用于创建和处理列联表的函数

函 数	描 述
<code>table(var1, var2, ..., varN)</code>	使用 N 个类别型变量（因子）创建一个 N 维列联表
<code>xtabs(formula, data)</code>	根据一个公式和一个矩阵或数据框创建一个 N 维列联表
<code>prop.table(table, margins)</code>	依 <code>margins</code> 定义的边际列表将表中条目表示为分数形式
<code>margin.table(table, margins)</code>	依 <code>margins</code> 定义的边际列表计算表中条目的和
<code>addmargins(table, margins)</code>	将概述边 <code>margins</code> （默认是求和结果）放入表中
<code>ftable(table)</code>	创建一个紧凑的“平铺”式列联表



频数表和列联表

一维列联表

使用table()函数生成简单的频数统计表

```
> mytable <- with(Arthritis, table(Improved))  
> mytable  
Improved  
None      Some Marked  
42         14        28
```



频数表和列联表

一维列联表

使用`prop.table()`将频数转化为比例值

```
> prop.table(mytable)
Improved
  None   Some Marked
0.500  0.167  0.333
```

```
> prop.table(mytable)*100
Improved
  None   Some Marked
50.0   16.7   33.3
```



频数表和列联表

❧ 二维列联表 使用xtabs()函数


```
mytable <- xtabs(~ A + B, data=mydata)
```

```
> mytable <- xtabs(~ Treatment+Improved, data=Arthritis)
> mytable
```

	Improved		
Treatment	None	Some	Marked
Placebo	29	7	7
Treated	13	7	21



频数表和列联表




```
#使用prop.table()生成各单元所占比例  
prop.table(mytable)
```

❧ 使用margin.table()和prop.table()函数分别生成边际频数和比例

```
#使用margin.table()生成边际频数  
margin.table(mytable, 1)  
margin.table(mytable, 2)
```

```
#使用prop.table()生成边际比例  
prop.table(mytable, 1)  
prop.table(mytable, 2)
```





频数表和列联表

使用addmargins()函数为这些表格添加边际和

```
addmargins(mytable)
addmargins(prop.table(mytable))
addmargins(prop.table(mytable), 2)
addmargins(prop.table(mytable), 1)
```



频数表和列联表

```
library(vcd)
mytable <- xtabs(~Treatment+Improved, data=Arthritis)
chisq.test(mytable)
mytable <- xtabs(~Improved+Sex, data=Arthritis)
chisq.test(mytable)
```

❧ 独立性检验

❧ 使用chisq.test()函数对二维表的行变量和列变量进行卡方独立性检验

```
> library(vcd)
> mytable <- xtabs(~Treatment+Improved, data=Arthritis)
> chisq.test(mytable)
```

Pearson's Chi-squared test

```
data: mytable
X-squared = 13.1, df = 2, p-value = 0.001463
```

```
> mytable <- xtabs(~Improved+Sex, data=Arthritis)
> chisq.test(mytable)
```

Pearson's Chi-squared test

```
data: mytable
X-squared = 4.84, df = 2, p-value = 0.0889
```

Warning message:

In chisq.test(mytable) : Chi-squared approximation may be incorrect

① 治疗情况和改善情况不独立



② 性别和改善情况独立





相关

🦌 `cor()`函数可计算相关系数，`cov()`函数可用来计算协方差

```
cor(x, use= , method= )
```

```
states <- state.x77[1:6]
cov(states)
cor(states)
cor(states, method="spearman")
```

参 数	描 述
<code>x</code>	矩阵或数据框
<code>use</code>	指定缺失数据的处理方式。可选的方式为 <code>all.obs</code> （假设不存在缺失数据——遇到缺失数据时将报错）、 <code>everything</code> （遇到缺失数据时，相关系数的计算结果将被设为 <code>missing</code> ）、 <code>complete.obs</code> （行删除）以及 <code>pairwise.complete.obs</code> （成对删除， <code>pairwise deletion</code> ）
<code>method</code>	指定相关系数的类型。可选类型为 <code>pearson</code> 、 <code>spearman</code> 或 <code>kendall</code>



相关

❧ 非方形相关矩阵相关系数计算

```
x <- states[,c("Population", "Income", "Illiteracy", "HS Grad")]  
y <- states[,c("Life Exp", "Murder")]  
cor(x,y)
```

	Life Exp	Murder
Population	-0.068	0.344
Income	0.340	-0.230
Illiteracy	-0.588	0.703
HS Grad	0.582	-0.488



相关

- ❧ 偏相关 指在控制一个或多个定量变量时，另外两个定量变量之间的相互关系
- ❧ 使用pcor()函数计算偏相关系数

```
install.packages("igraph")  
install.packages("ggm")  
library(igraph)  
library(ggm)  
pcor(c(1,5,2,3,6), cov(states))
```

```
pcor(u, S)
```

```
> library(ggm)  
> # 在控制了收入、文盲率和高中毕业率时  
> # 人口和谋杀率的偏相关系数  
> pcov(c(1,5,2,3,6), cov(states))  
[1] 0.346
```



相关

- 相关性显著性检验 使用cor.test()函数对单个的Pearson、Spearman和Kendall相关系数进行检

```
cor.test(states[,3], states[,5])
```

```
cor.test(x, y, alternative = , method = )
```



相关

通过corr.test计算相关矩阵并进行显著性检验

```
library(psych)  
corr.test(states, use="complete")
```



t检验

使用MASS包中的UScrime数据集

```
> UScrime
```

	M	So	Ed	Po1	Po2	LF	M.F	Pop	NW	U1	U2	GDP	Ineq	Prob	Time	y
1	151	1	91	58	56	510	950	33	301	108	41	394	261	0.084602	26.2011	791
2	143	0	113	103	95	583	1012	13	102	96	36	557	194	0.029599	25.2999	1635
3	142	1	89	45	44	533	969	18	219	94	33	318	250	0.083401	24.3006	578
4	136	0	121	149	141	577	994	157	80	102	39	673	167	0.015801	29.9012	1969
5	141	0	121	109	101	591	985	18	30	91	20	578	174	0.041399	21.2998	1234
6	121	0	110	118	115	547	964	25	44	84	29	689	126	0.034201	20.9995	682



t检验

独立样本的t检验

```
library(MASS)  
t.test(Prob ~ So, data=UScrime)
```

```
t.test(y ~ x, data)
```





t检验

非独立样本的t检验

```
sapply(UScrime[c("U1", "U2")], function(x){c(mean=mean(x), sd=sd(x))})  
with(UScrime, t.test(U1, U2, paired=TRUE))
```

```
t.test(y1, y2, paired=TRUE)
```



t检验

❧ 多于两组以使用方差分析 (ANOVA)





非参数检验

🦋 两组数据独立 秩和检验

```
with(UScrime, by(Prob, So, median))  
wilcox.test(Prob ~ So, data=UScrime)
```

```
wilcox.test(y ~ x, data)
```



非参数检验

🌀 两组数据不独立 符号检验

```
sapply(UScrime[c("U1", "U2")], median)  
with(UScrime, wilcox.test(U1, U2, paired=TRUE))
```




非参数检验

❧ 多于两组的比较

```
states <- data.frame(cbind(state.region, state.x77))  
kruskal.test(literacy ~ state.region, data=states)
```

```
kruskal.test(y ~ A, data)
```



小结

- 描述性统计分析
- 频数表和列联表
- 相关系数和协方差
- t检验
- 非参数统计





Thankyou !

