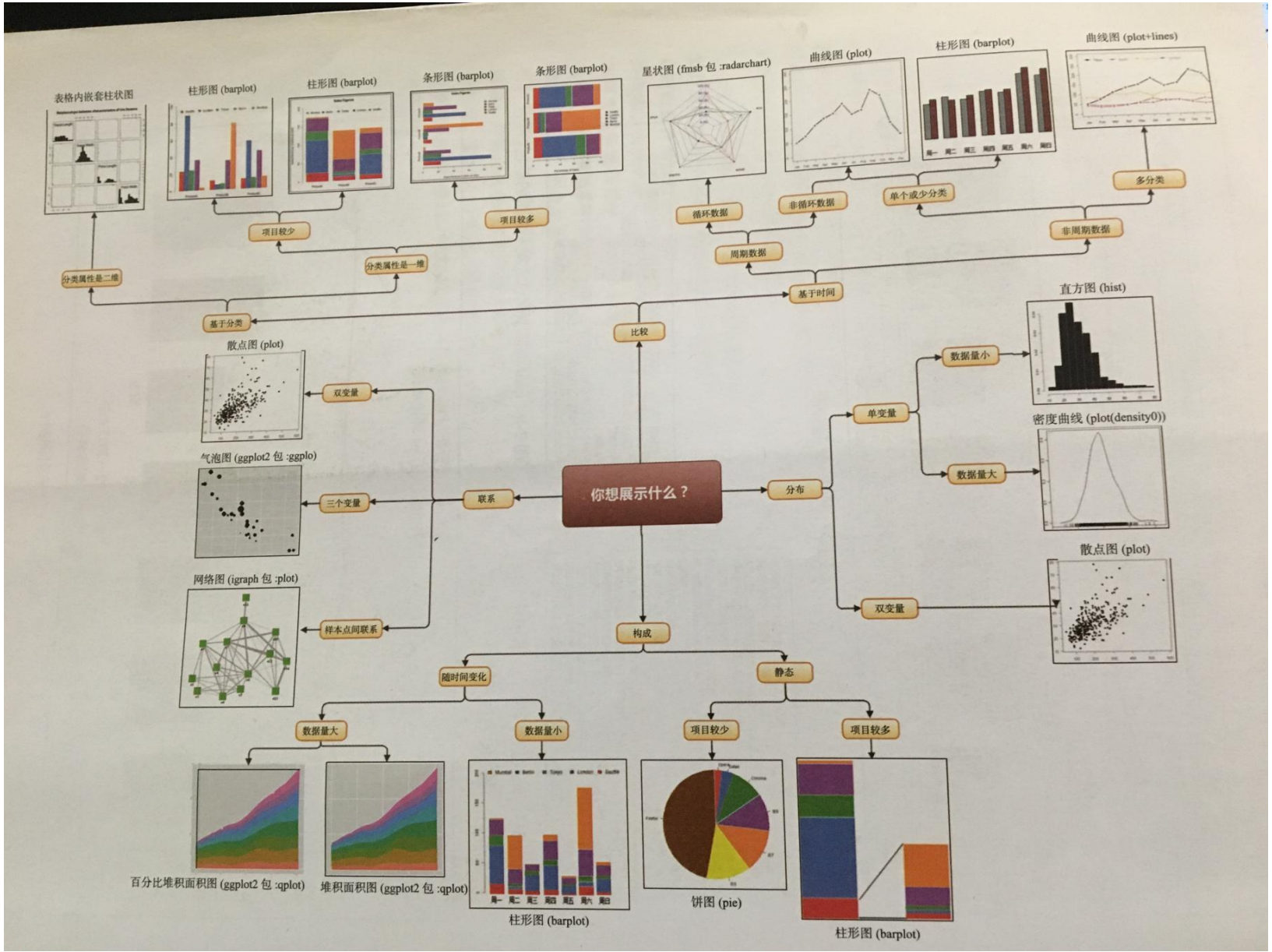
天善智能
TIANSHAN SOFT
专注于商业智能BI和大数据的垂直社区平台

数据分析常用方法

谢佳标（Daniel.xie）

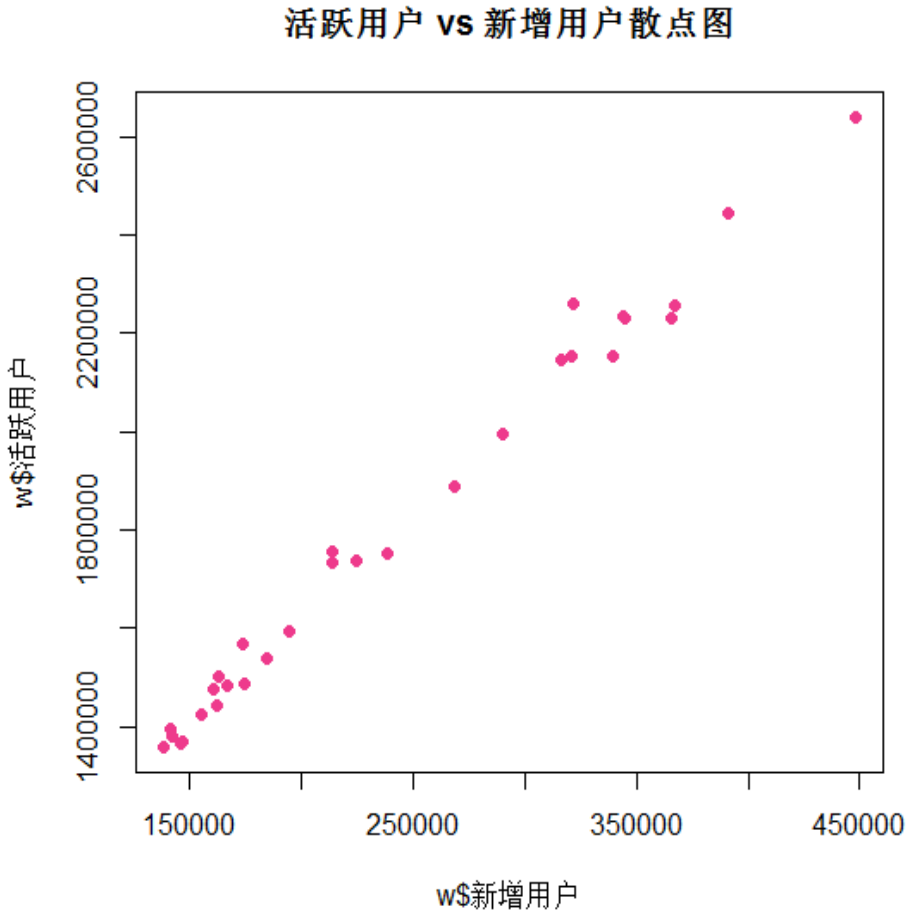
图表选择

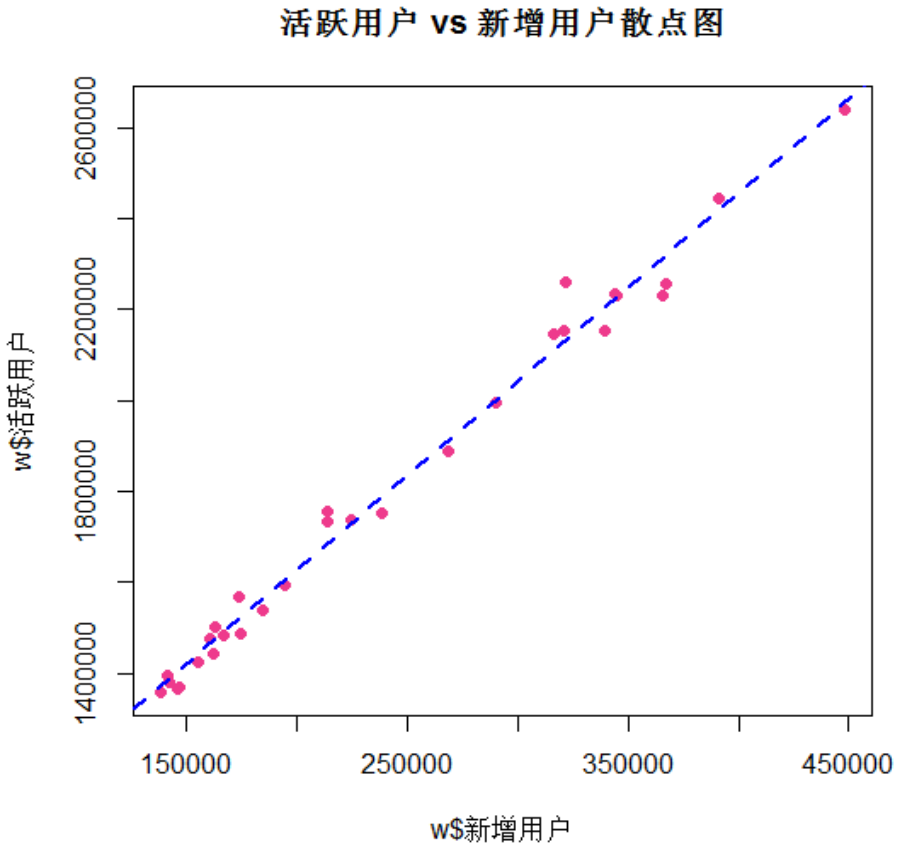


摘自李明《R语言与网站分析》

双指标数据可视化

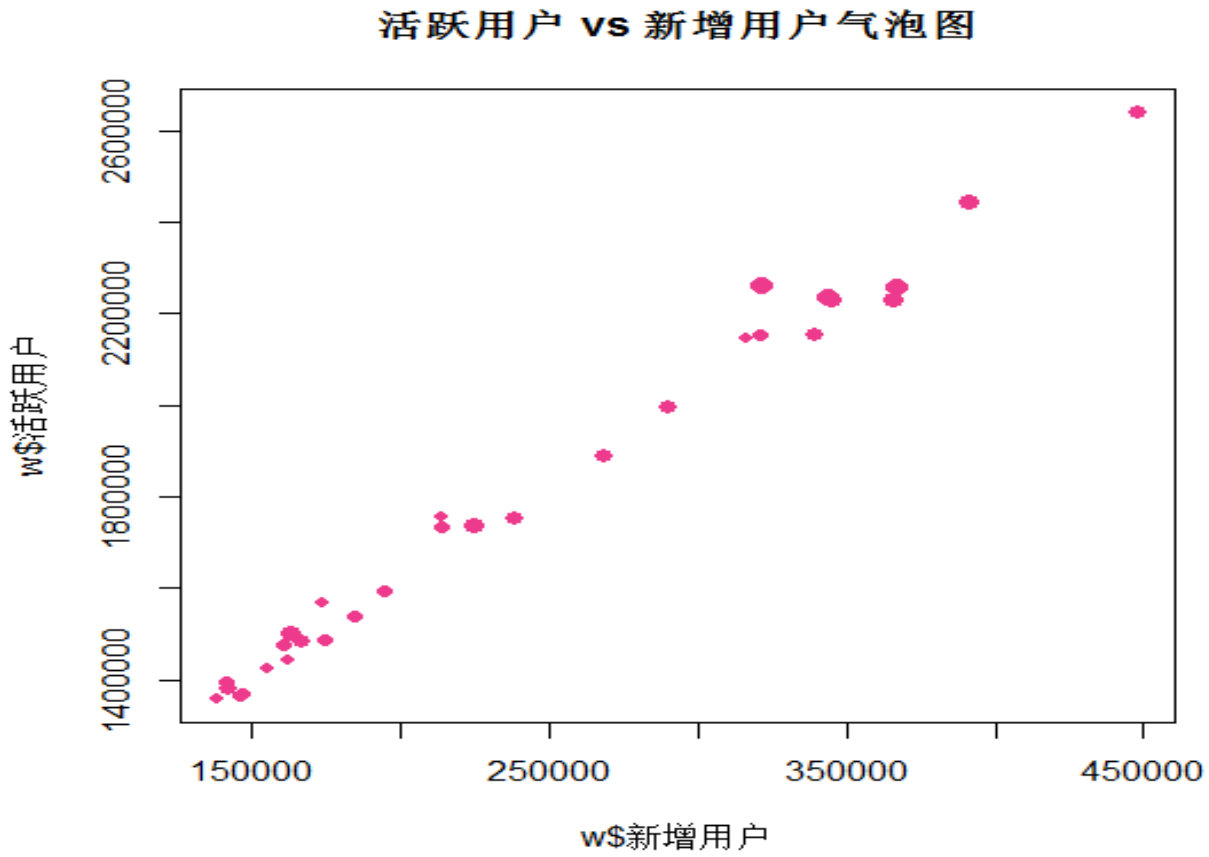
- 有时候，我们需要查看两个指标间的关系，可以通过绘制散点图的方式，用于查看其数据分布情况。





三指标数据可视化

- 有时候，可能需要同时查看三个指标的数据，此时我们可以在散点图的基础上，用第三个指标来控制散点的颜色或者大小，通过这样的方式来将三个维度的数据在二维平面展示出来。



案例：时间序列数据预测

- 时间序列分析是一种动态数据处理的统计方法，典型的假设是相邻观测值具有某种依赖性，从而基于随机过程理论和数理统计学方法，研究随机数据序列所遵从的统计规律。

- 在R语言中，ts(<向量对象>)可以把一个向量转化为一个时间序列对象。其表达形式为：

```
ts(data = NA, start = 1, end = numeric(), frequency = 1,
    deltat = 1, ts.eps = getOption("ts.eps"), class = , names = )
```

案例：时间序列数据预测

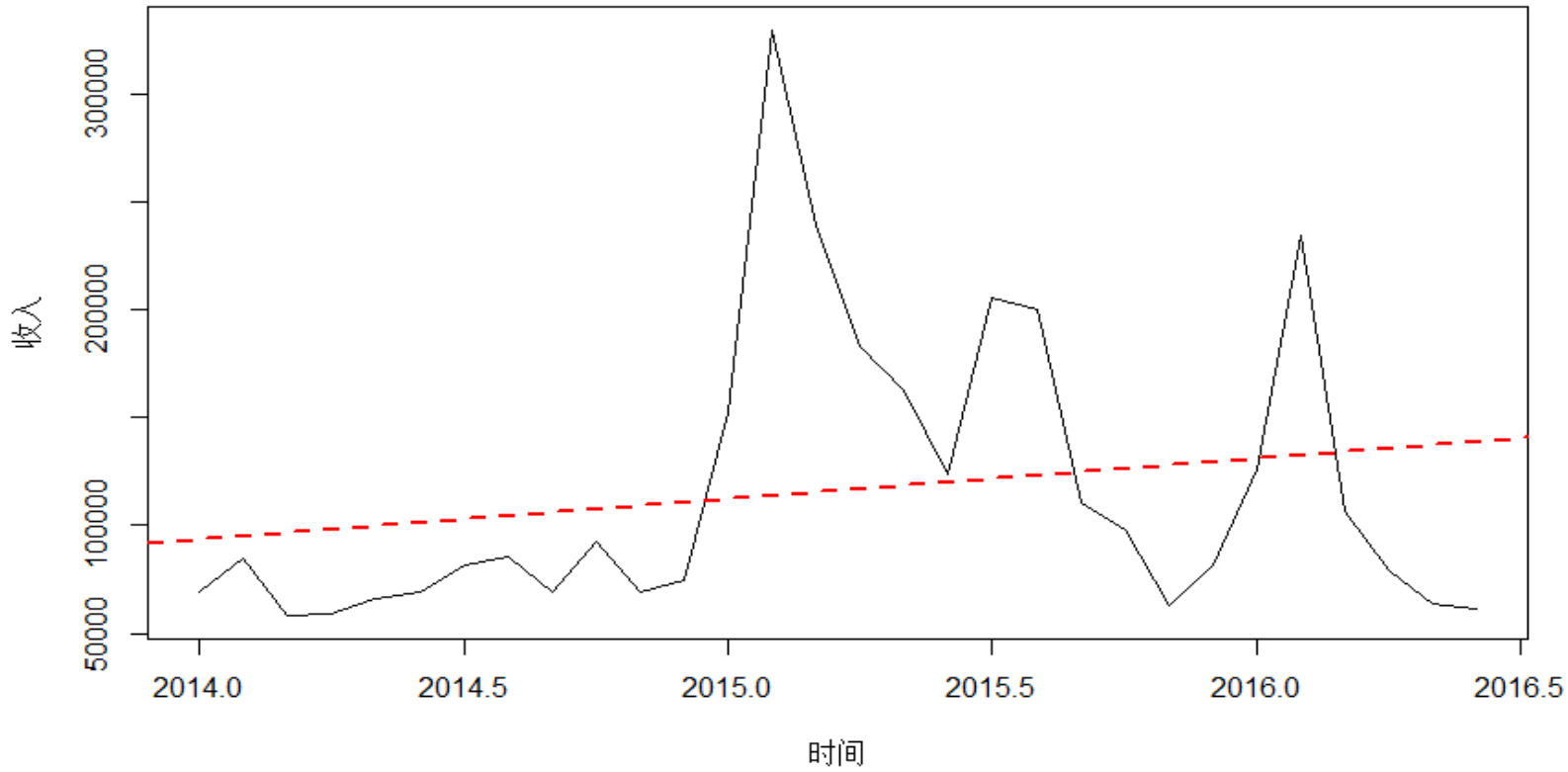
- 收集2014年1月到2016年6月的某游戏每月收入数据，我们将收入数据转换成时间序列对象，为建立时序模型做前期准备。

```
> revenue <- read.csv("收入数据.csv",T) #导入数据
> head(revenue)                    #查看前六行
日期收入
1 2014年1月 69359
2 2014年2月 84741
3 2014年3月 58485
4 2014年4月 59013
5 2014年5月 66076
6 2014年6月 68972
> revenue.ts <- ts(revenue[,2],frequency = 12,start = c(2014,1)) # 转换成时间序列对象
> is.ts(revenue.ts)                #判断是否为时间序列对象
[1] TRUE
> start(revenue.ts)                #查看开始日期
[1] 2014    1
> end(revenue.ts)                  #查看结束日期
[1] 2016    6
> frequency(revenue.ts)            #查看一个周期的频数
[1] 12
```

案例：时间序列数据预测

- 通过plot.ts函数可以绘制时间序列的时序图。执行以下代码绘制时序图，并添加线性拟合直线。

```
> plot.ts(revenue.ts,xlab="时间",ylab="收入")
> abline(lm(revenue.ts~time(revenue.ts)),col="red",lty=2,lwd=2)
```



案例：时间序列数据预测

- 对于时间序列的平稳性检验通常使用单位根检验的方法。在R中，我们可以使用fUnitRoots包中的unitrootTest函数实现。

unitrootTest(x, lags = 1, type = c("nc", "c", "ct"), title = NULL, description = NULL)

其中，输入参数x为观测值序列，lags为用于校正误差项的最大滞后项，type为单位根的回归类型，返回的参数p值，p值小于0.05表示满足单位根检验。

```
> library(fUnitRoots)
> unitrootTest(revenue.ts)
Title:
 Augmented Dickey-Fuller Test
Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    DF: -1.1738
  P VALUE:
    t: 0.2139
    n: 0.4314
Description:
 Tue Jul 26 20:55:45 2016 by user: Think
```

案例：时间序列数据预测

- 对于非平稳时间序列，首先需要对其进行差分得到一个平稳时间序列。在R软件中，可以使用diff()函数对时间序列进行差分运算，diff()函数的用法如下：

  diff(x, lag = 1, differences = 1, ...)

  其中，输入参数"x"代表观测值序列；"lag"代表差分运算的步数，缺省值代表一步差分；

  " differences"代表差分运算的阶数，缺省值代表一阶差分。

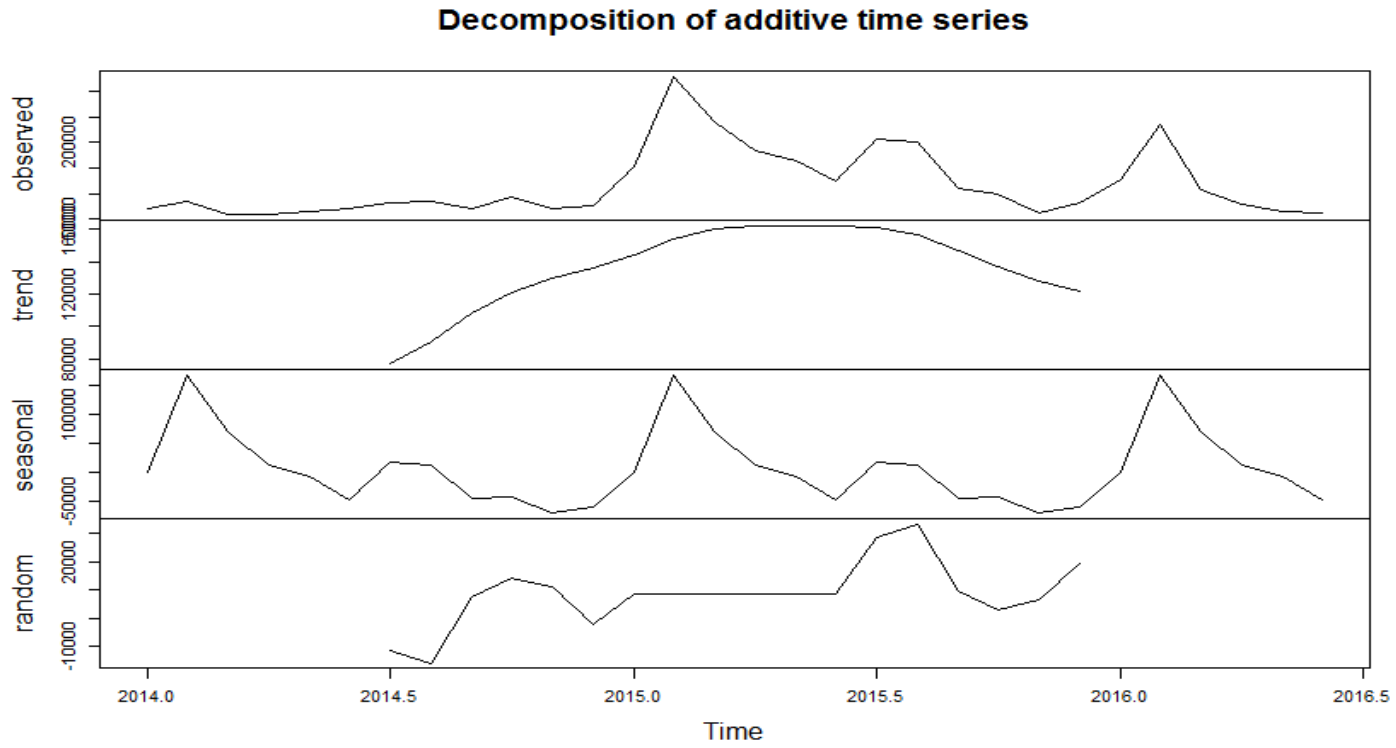
```
> revenue.ts.dif <- diff(revenue.ts)
> unitrootTest(revenue.ts.dif)
Title:
 Augmented Dickey-Fuller Test
Test Results:
  PARAMETER:
    Lag Order: 1
  STATISTIC:
    DF: -4.6456
  P VALUE:
    t: 4.186e-05
```

案例：时间序列数据预测

- 时间序列的变化主要受到长期趋势、季节变动、周期变动和噪声变动这四个因素的影响。根据序列的特点，可以构建加法模型和乘法模型。可以通过decompose（ ）函数实现。其基本表达形式为：

decompose(x, type = c("additive", "multiplicative"), filter = NULL)

其中，x为时间序列对象；type指定分解为加法模型还是乘法模型；filter是滤波系数。



案例：时间序列数据预测

- 在R中，arima()函数设置时序模式的建模参数，创建ARIMA时序模型或者把一个回归时序模型转换为ARIMA模型。其形式为：

  arima(x, order, seasonal, period,method, ...)

  其中，x为观测值序列，order为构建的ARIMA(p,d,q)模型的参数，seasonal为模型的季节性参数，period为观测值序列的周期，method为估计模型参数所使用的方法。

```
> library(forecast)
> fit <- auto.arima(revenue.ts)
> fit
Series: revenue.ts
ARIMA(0,1,0)(0,1,0)[12]

sigma^2 estimated as 3.395e+09:  log likelihood=-210.66
AIC=423.32   AICc=423.59   BIC=424.15
> fit.forecast <- forecast(fit,h=6)
> fit.forecast
         Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
Jul 2016       143605   68929.78 218280.2   29399.10 257810.9
Aug 2016       137927   32320.29 243533.7  -23584.53 299438.5
Sep 2016        48021   -81330.28 177362.2 -140780.42 236821.4
```

相关分析基本原理

- 相关分析就是分析连续变量之间线性的相关程度的强弱，并用适当的统计指标表示出来的过程。相关系数可以用来描述连续变量之间的关系，相关系数的符号（正负）表明关系的方向（正相关或负相关），数值大小表示关系的强弱（完全不相关时为0，完全相关时为1）。

- 在二元变量的相关分析过程中比较常用的如Pearson相关系数、Spearman秩相关系数和Kendall’s Tau相关系数。Pearson相关系数一般用于对定距变量的数据进行计算，即分析两个连续性变量之间的关系；Spearman秩相关系数用于描述分类或等级变量之间、分类或等级变量与连续变量之间的关系；Kendall’s Tau相关系数也是一种非参数的等级相关的度量。

相关关系可视化

- 相关系数矩阵可以很直观地展示各变量间的关系强弱，但是随着变量的增加，想从相关系数矩阵中快速发现不同变量间的关系和强弱变得困难，此时我们可以利用相关图对相关系数矩阵进行可视化展示。

- 在R中，绘制相关图的方式有好多，最常用的是corrgram扩展包中的corrgram函数，以及魏太云开发的corrplot包中的corrplot函数。

案例：活跃时间段相关分析

- logindata数据集收录了30000位玩家上个月在不同渠道、不同游戏的登录天数和登录次数的数据。将数据导入R中，并查看数据结构。

```
> logindata <- read.csv("data\\logindata.csv")
> str(logindata)
'data.frame':   30000 obs. of  7 variables:
 $ 渠道名称: Factor w/ 15 levels "渠道A","渠道B",..: 2 5 7 7 11 5 9 5 11 5 ...
 $ 游戏名称: Factor w/ 7 levels "游戏A","游戏B",..: 5 3 1 3 3 3 4 3 5 3 ...
 $ 是否付费: Factor w/ 2 levels "否","是": 1 1 1 1 1 1 1 2 2 2 ...
 $ 性别    : Factor w/ 2 levels "男","女": 1 1 1 1 2 2 2 1 2 1 ...
 $ 年龄    : int  39 50 38 53 28 37 49 52 31 42 ...
 $ 登录天数: Factor w/ 5 levels "1天","2天","3天",..: 5 5 5 3 3 5 3 5 5 5 ...
 $ 登录次数: Factor w/ 9 levels "100次以上","11至20次",..: 9 8 6 6 6 6 6 8 6 6 ...
> dim(logindata)
[1] 30000     7
```

案例：活跃时间段相关分析

- 可见，除了年龄变量是数值型变量，其他变量均为因子型变量。由于cor函数要求是数值型变量，所以我们需要先将这些变量都先进行哑变量处理，再求出新变量间的相关系数值。利用第二节课上提到的caret包中的dummyVars函数进行哑变量处理。

```
> library(caret)
> dmy<-dummyVars(~.,data=logindata)
> dmyTsrf<-data.frame(predict(dmy,newdata=logindata))
> dim(dmyTsrf)
[1] 30000    41
> str(dmyTsrf)
'data.frame':   30000 obs. of  41 variables:
 $ 渠道名称.渠道A    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 渠道名称.渠道B    : num  1 0 0 0 0 0 0 0 0 0 ...
 $ 渠道名称.渠道C    : num  0 0 0 0 0 0 0 0 0 0 ...
……
$ 登录次数.41至50次 : num  0 0 0 0 0 0 0 0 0 0 ...
 $ 登录次数.51至60次 : num  0 1 0 0 0 0 0 1 0 0 ...
 $ 登录次数.61.100次 : num  1 0 0 0 0 0 0 0 0 0 ...
```

案例：活跃时间段相关分析

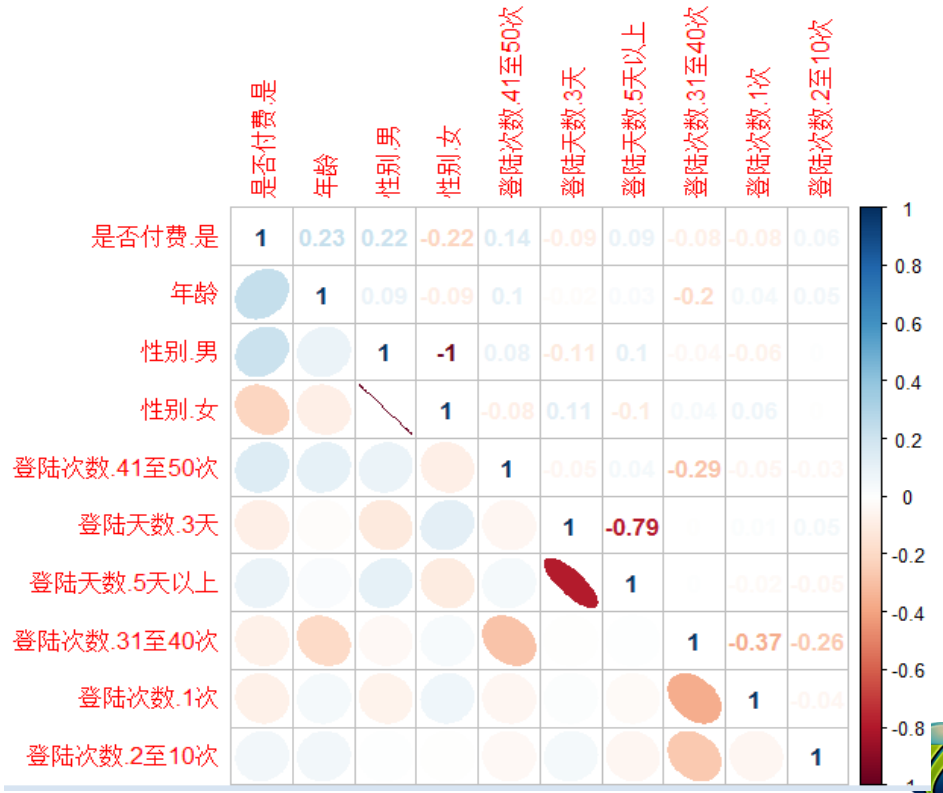
- 可见，经过哑变量处理后，所有变量都已经转换成数值型变量。我们利用自定义函数求出变量间的相关系数值及其显著性检验的P值，并对结果按照相关系数绝对值进行降序排序。

```
> # 导入自定义的求相关系数函数
> source("code//CorrelationFunction.R")
> corMasterList<-flattenSquareMatrix(cor.prob(dmyTsrf))
> # 按照相关系数的绝对值进行降序排序
> corList<-corMasterList[order(-abs(corMasterList$cor)),]
> print(head(corList,10))
                  i                 j     cor p
276      是否付费.否是否付费.是 -1.0000000 0
325        性别.男性别.女 -1.0000000 0
562   渠道名称.渠道A     登录次数.1次  0.9984244 0
495    登录天数.3天登录天数.5天以上 -0.7887687 0
189   游戏名称.游戏C    游戏名称.游戏E -0.6439172 0
345   游戏名称.游戏E           年龄 -0.5335248 0
249   游戏名称.游戏C      是否付费.否 -0.4442365 0
271   游戏名称.游戏C      是否付费.是  0.4442365 0
779 登录次数.31至40次登录次数.51至60次 -0.4431728 0
```

案例：活跃时间段相关分析

- 最后，我们提取与“是否付费.是”的相关系数大于0.04的记录，并绘制相关系数图。

```
> # 提取与“是否付费.是”的相关系数大于0.04的记录
> selectedSub <- subset(corList,(abs(cor)>0.04 & i %in%  c('是否付费.是')))
> bestsub <- as.character(selectedSub$j)
> # 绘制相关系数图
> library(corrplot)
> corrplot.mixed(cor(dmyTsrf[,c('是否付费.是',bestsub)]),
+         lower = "ellipse", upper = "number",
+         tl.pos="lt",diag="u")
```



游戏中的降维技术

- 游戏指标存在着某些复杂的关系，有时候我们需要利用降维技术消除这些指标间的共线性因素，再进行下一步的研究，或利用对应分析对R-Q型关系进行分析。

- 主成分分析及因子分析的目的都是降维，就是把相关的变量数目减少，用较少的变量来取代原始变量，而这些新变量为原始变量的线性组合。这种转变的目的，可以降低原始数据的维度，同时也在此过程中发现原始数据属性之间的关系。

- 因子分析是主成分的推广和延伸，是寻找公共因子的模型分析方法，它是在主成分的基础上构筑若干意义较为明确的公因子，以它们为框架分解原始变量，以此考察原始变量间的联系与区别。

对应分析基本原理

- 对应分析是在因子分析的基础上发展起来的。对应分析把R型因子分析和Q型因子分析统一起来，通过R型因子分析直接得到Q型因子分析的结果，同时把变量和样本反映到相同的坐标轴（因子轴）的一张图上，以此来说明变量与样本之间的对应关系。对应分析基本原理主要应用在市场细分、产品定位、竞争分析、用户偏好分析和广告研究等领域。

- 在R中，实现对应分析有3个包：ca、MASS、FactoMineR。其中ca包专门用于计算并可视化简单对应分析、多重及联合对应分析。ca包中的主要函数有：ca函数用于计算简单对应分析；mjca函数用于计算多重对应分析和联合对应分析；print和summary函数用于打印和汇总；plot和plot3d.ca函数用于绘图。

游戏玩家偏好分析

案例：游戏玩家偏好分析研究