数据分析与处理技术

时间序列预测

一、时间序列分析和预测的工具

加载工具包：forecast
加载案例工具包：fpp2
加载工具包：GGally

切换环境到fpp2，其中包含了关于时间序列教学用的大量案例数据

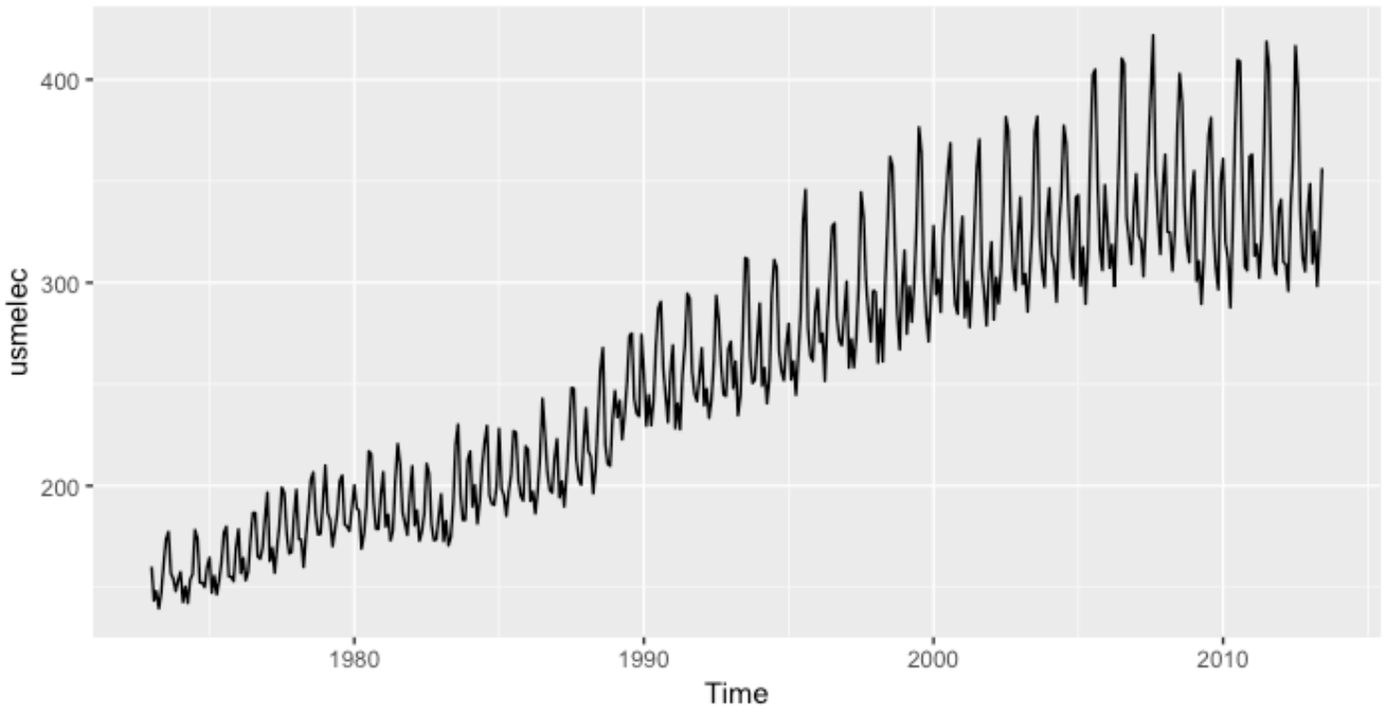
时间序列标记为 y_t 其中t为时间下标

package:fpp2	
Data	
elecdaily	Time-Series [1:365, 1:3]
prison	Time-Series [1:48, 1:32]
prisonLF	1536 obs. of 5 variables
uschange	Time-Series [1:187, 1:5]
Values	

autoplot自动识别变量类型，做时间序列散点图。

```
> autoplot(elec)

> autoplot(a10)

> autoplot(h02)
```

autoplot是ggplot2包中一个自动化绘图函数，与ggplot2语法一致



与ggplot()函数类似，一个图形中只有第一个图层用autoplot，之后图层添加序列使用autolayer代替。

一个时间序列建模预测和检验基本过程——指数平滑法

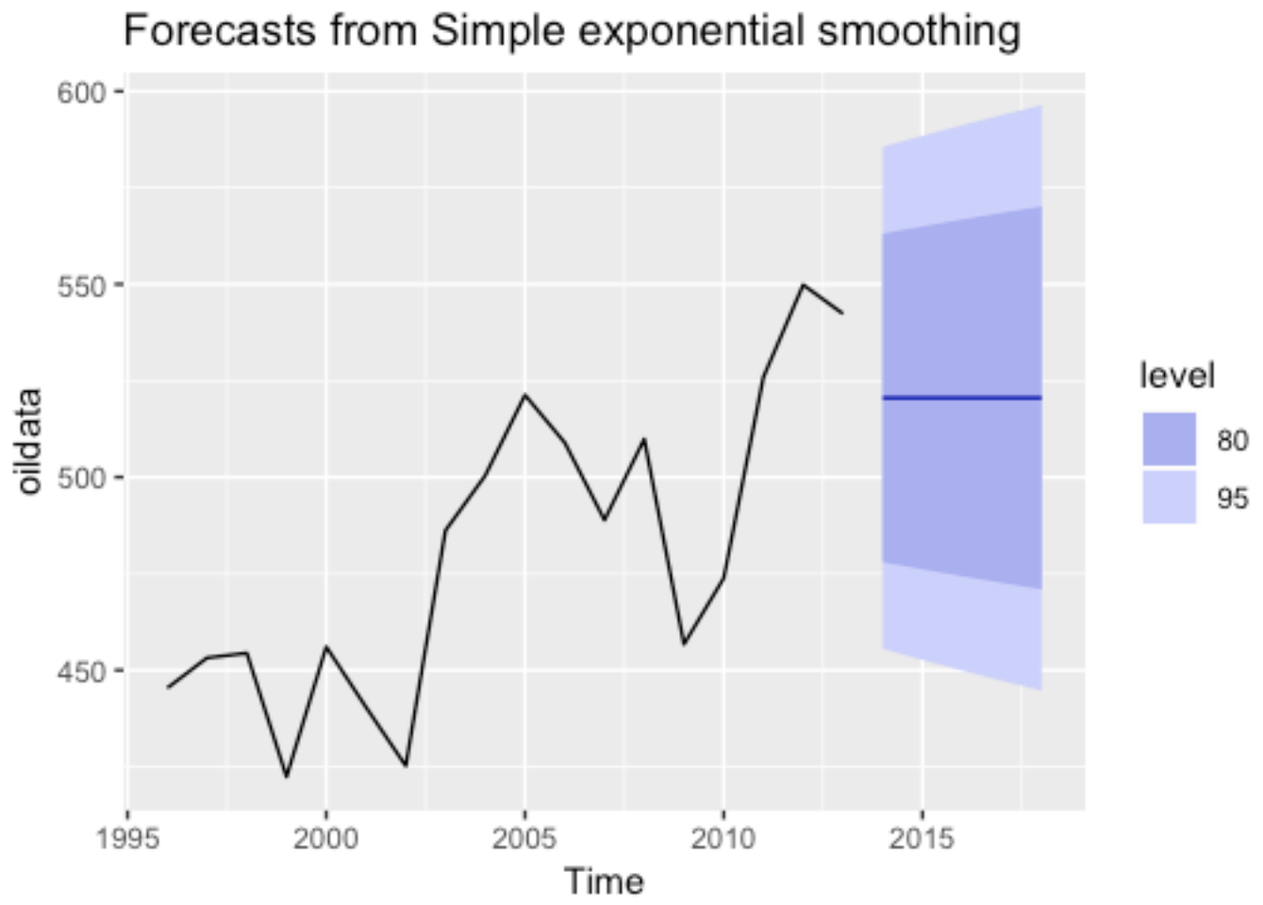
用下边案例说明通常步骤

```
oildata=window(oil,start=1996)

fc=ses(oildata,h=5,alpha = 0.3)

autoplot(fc)
```

截取1992年以后的石油价格数据，用指数平滑法做5期的预测，并可视化

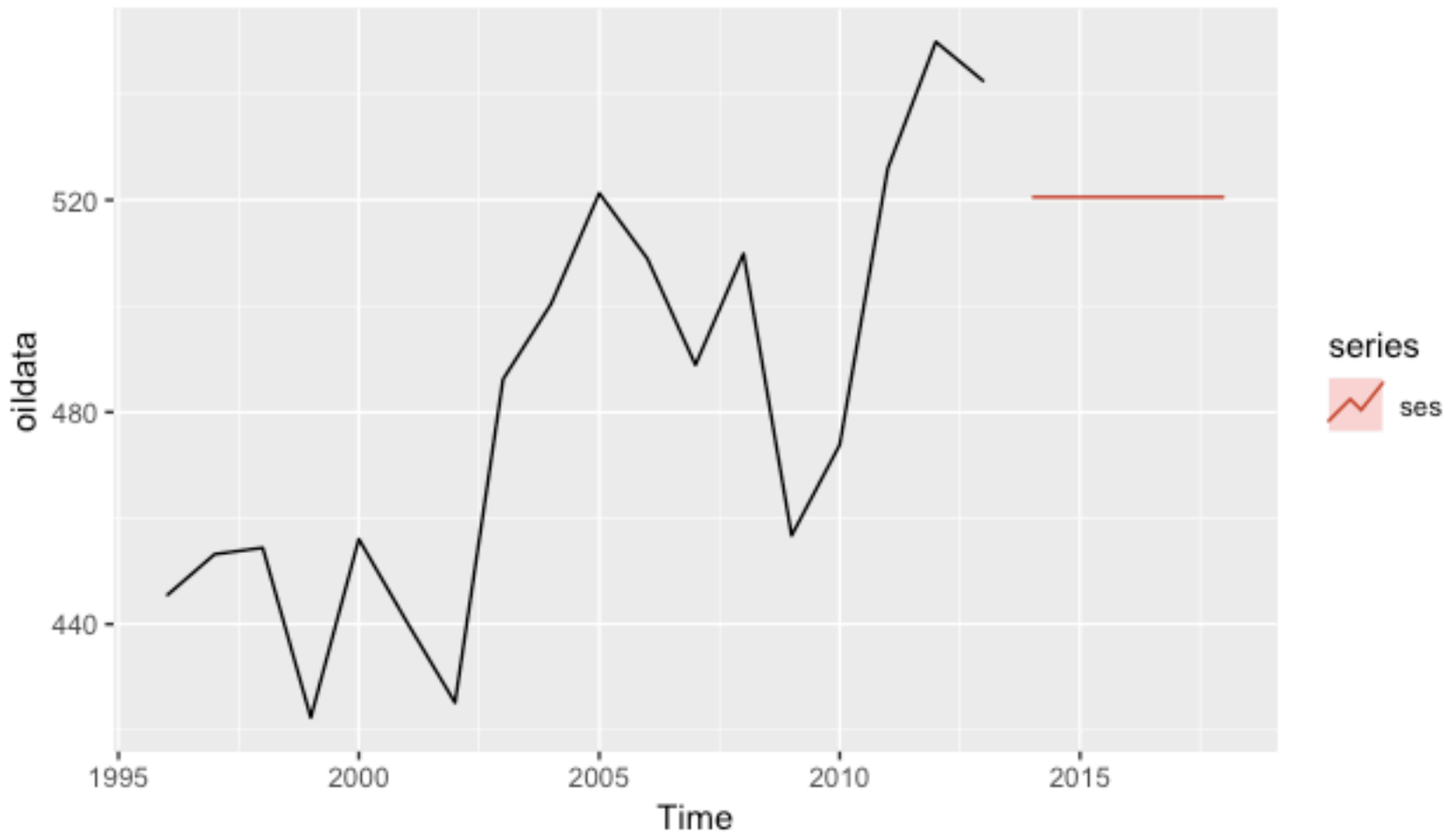
```
> round(accuracy(fc),2)
                ME   RMSE    MAE  MPE MAPE MASE ACF1
Training set 13.02 31.28 25.49 2.34 5.16 1.06 0.35
```

accuracy用来检测模型的拟合精度，有时为了读取方便用round限制四舍五入为2位小数



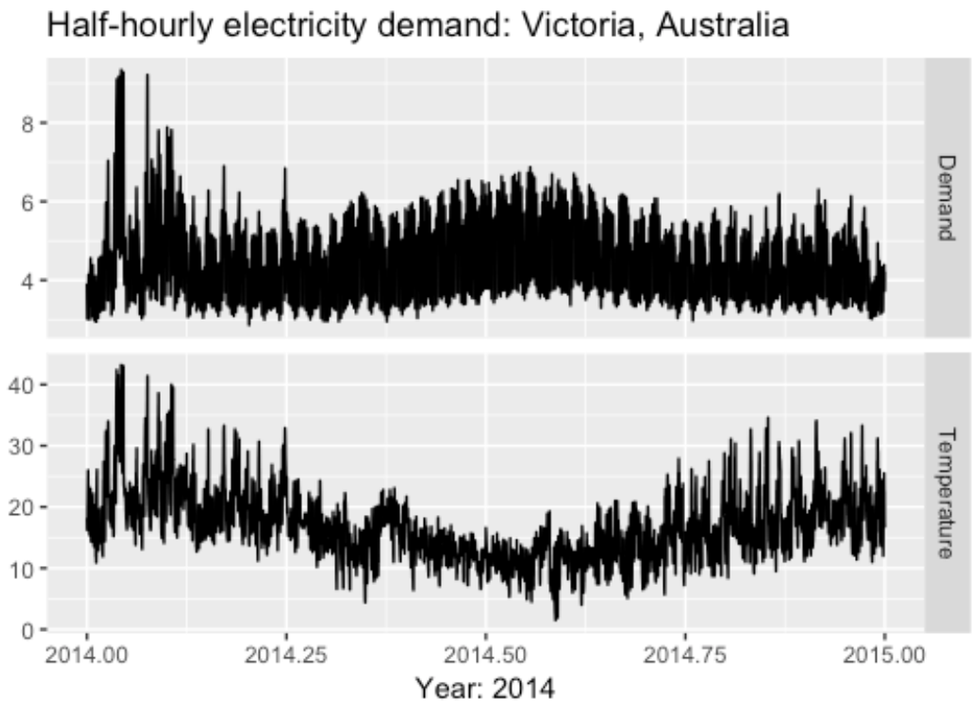
autolayer会为每一个自动图层添加图例，并且会自动识别时间序列和预测模型，按照规范将模型输出在同一个图片中

```
autoplot(oildata)+
    autolayer(fc,series='ses',PI=F)+
    guides(colour=guide_legend(title='series'))+
    ylab('Oil(millions of tonnes)')+xlab('Year')+
    ggtitle('Oil production in Saudi Arabia from 1996 to 2013')
```

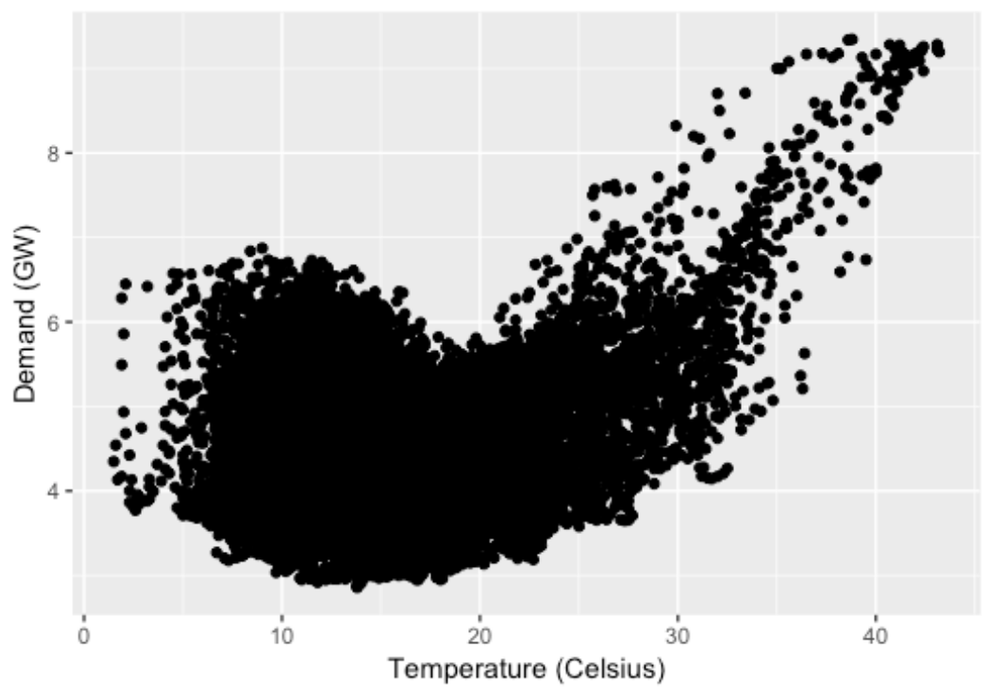


二、相关对比分析

```
autoplot(elecdemand[,c("Demand","Temperature")], facets=TRUE) +
    xlab("Year: 2014") + ylab("") +
    ggtitle("Half-hourly electricity demand: Victoria, Australia")
```

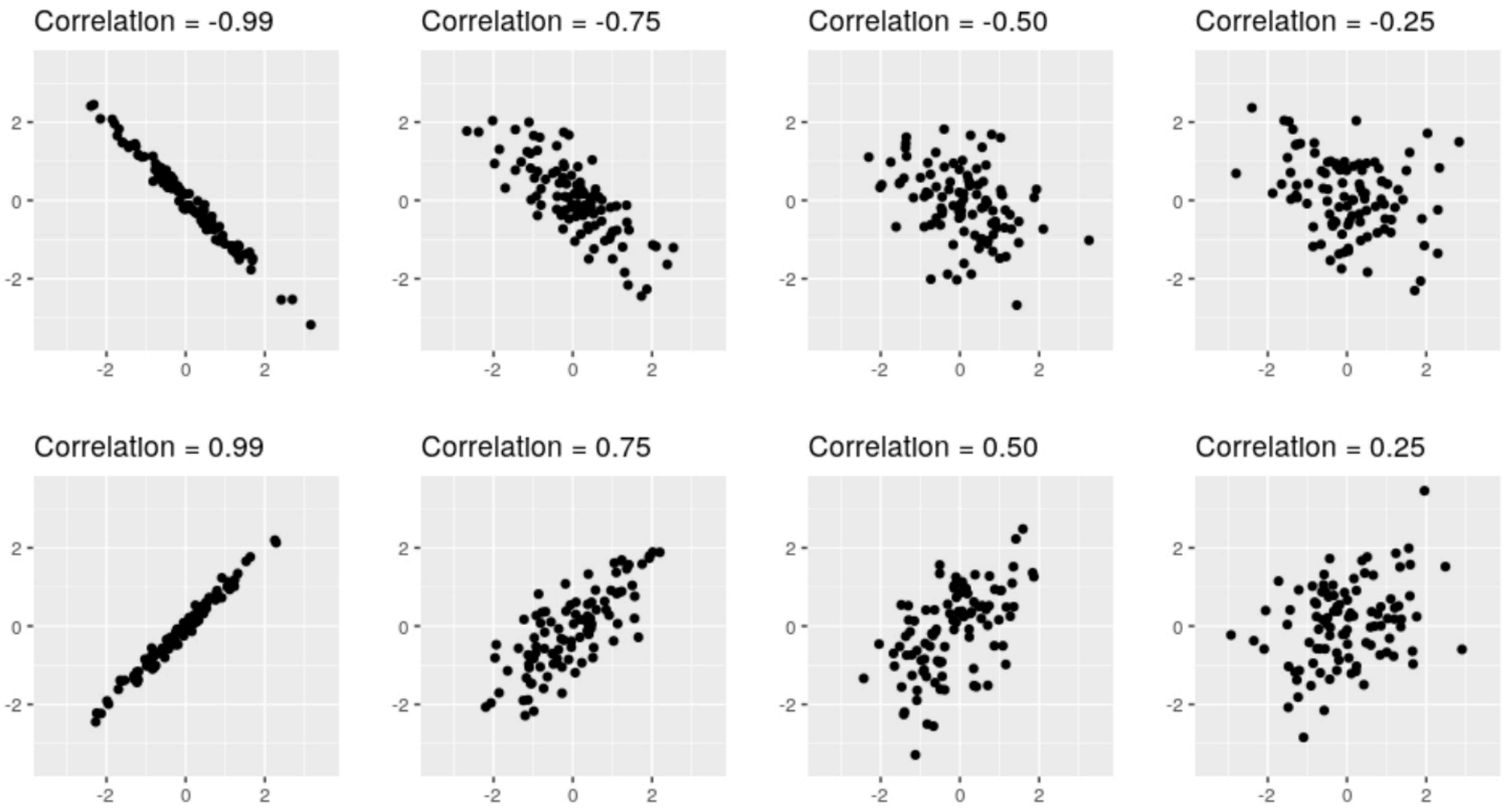



```
qplot(Temperature, Demand, data=as.data.frame(elecdemand)) +
    ylab("Demand (GW)") + xlab("Temperature (Celsius)")
```



从直观上感受不同水平的相关系数反应的序列关联性特征
下列用随机数做出了从弱相关到强相关(0.25到0.99)的四类正负相关序列散点图，观察不同系数水平代表的情景。

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}.$$



时间序列中一个重要的解释因子是自身滞后量，即 y_t y_{t-k} 之间的相关程度

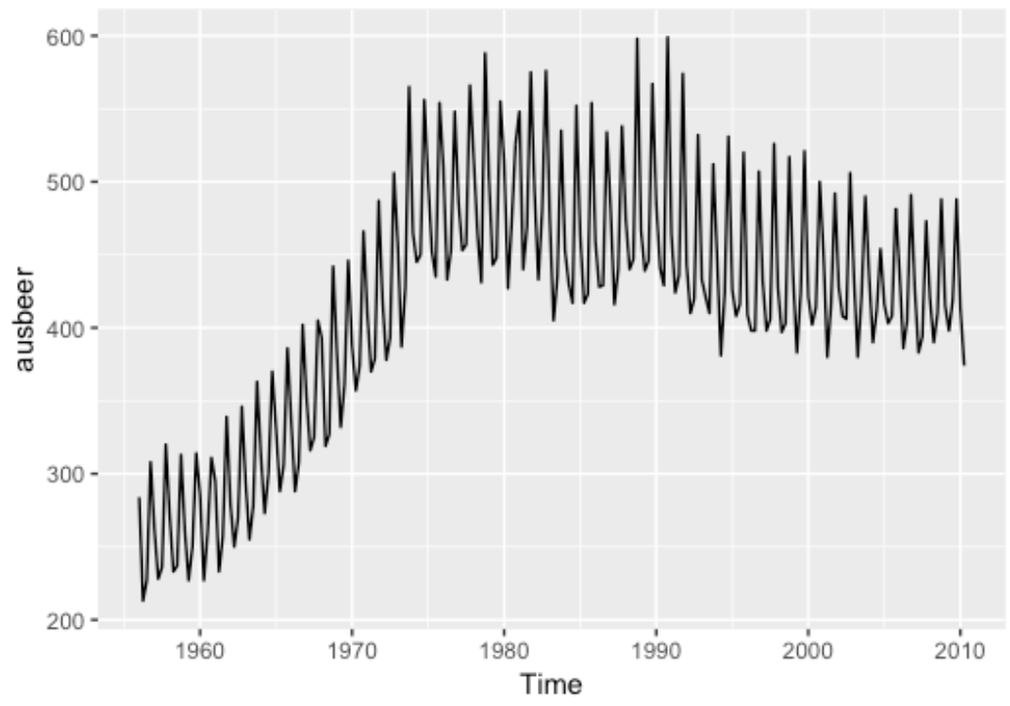
为了凸出季节周期带来的相关性，从ausbeer数据中截出1992年滞后的数据如下

```
beer2 <- window(ausbeer, start=1992)
```

数据集具体内容为：

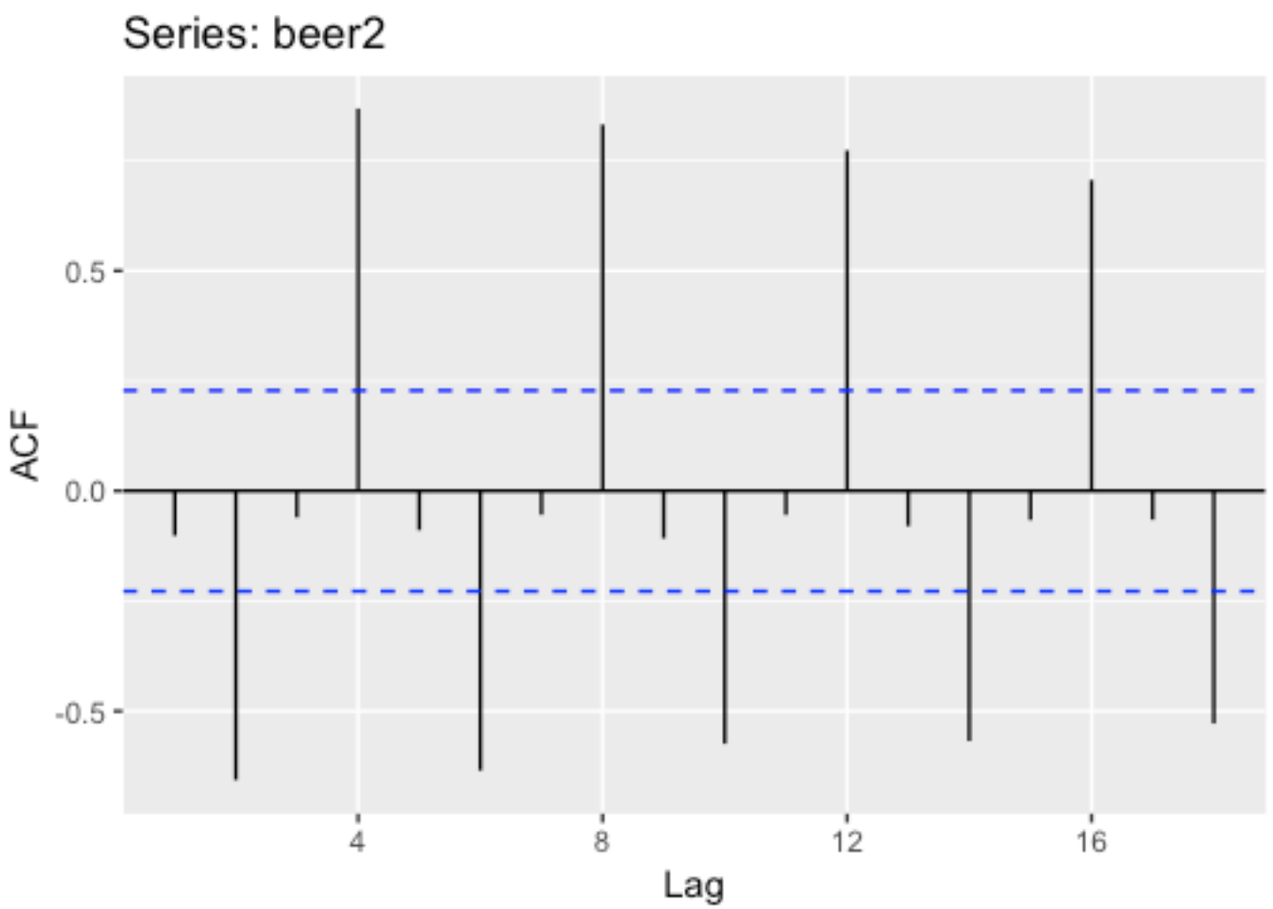
```
     Qtr1 Qtr2 Qtr3 Qtr4
1992  443  410  420  532
1993  433  421  410  512
1994  449  381  423  531
1995  426  408  416  520
1996  409  398  398  507
1997  432  398  406  526
1998  428  397  403  517
1999  435  383  424  521
2000  421  402  414  500
2001  451  380  416  492
2002  428  408  406  506
2003  435  380  421  490
2004  435  390  412  454
2005  416  403  408  482
2006  438  386  405  491
2007  427  383  394  473
2008  420  390  410  488
2009  415  398  419  488
2010  414  374
```

```
autoplot(ausbeer)
```



ACF图列出了各滞后期的具体相关系数数值，如下图

```
ggAcf(beer2)
```



白噪声与残差

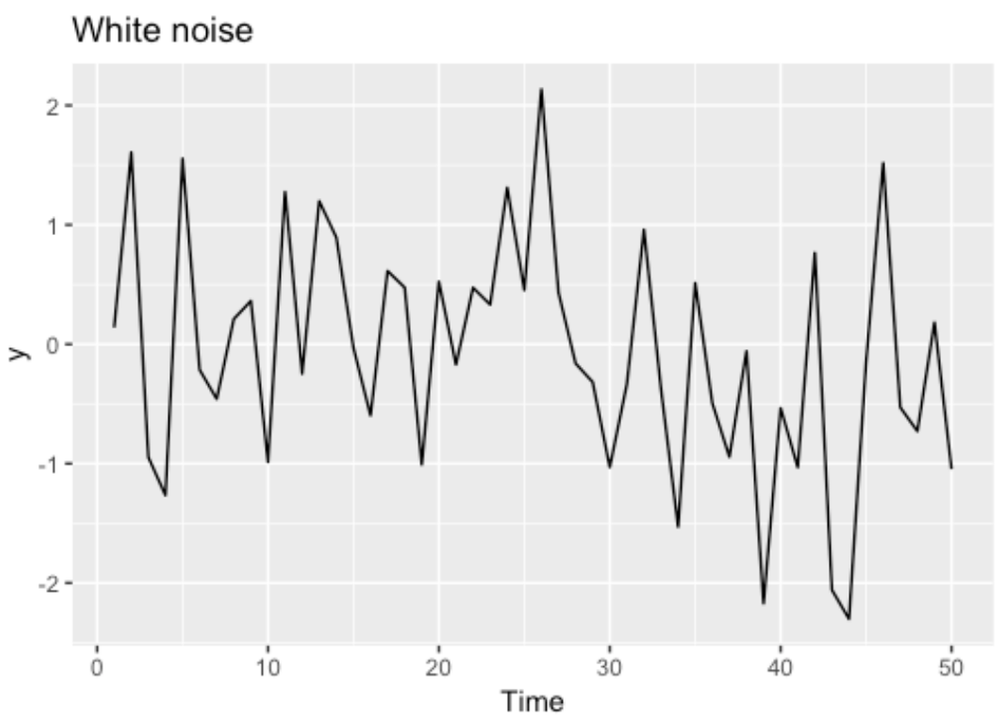
时间序列中没有显示出明显自相关特征的变动称为白噪声

当时间序列中具有明显规则的特征被分解出来之后，剩下的无法解释的部分成为噪声数据，即无法解释的随机事件。

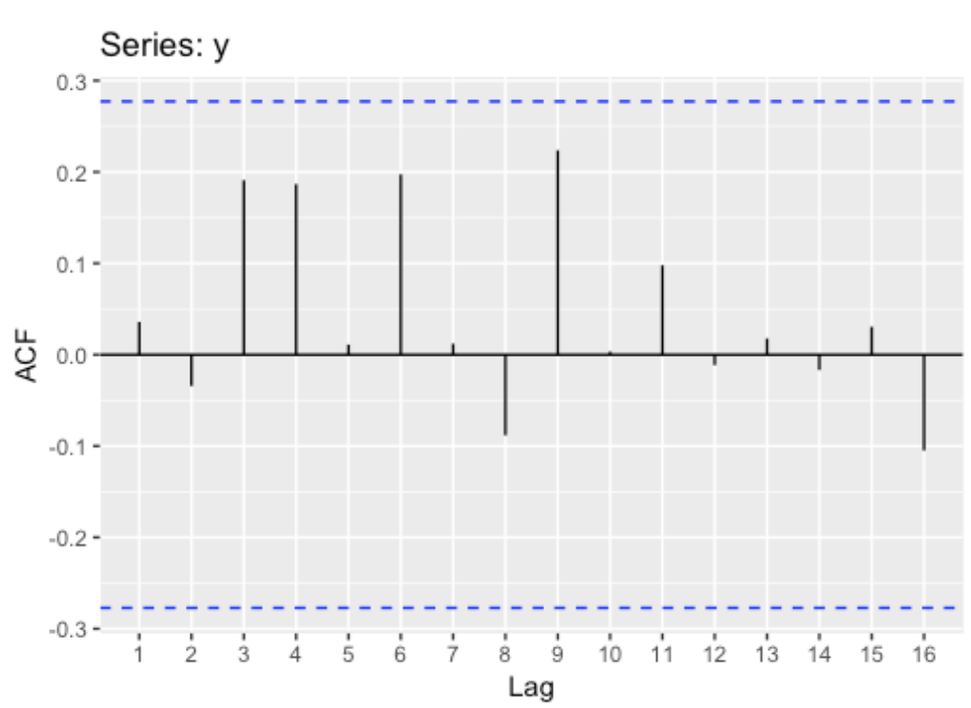
白噪声作为建模分析的剩余部分，最好的状态是不再含有明显的自相关，这可以通过残差的ACF图来检验

对于白噪声，我们希望序列的自相关能够降低到0，但这显然不现实，即使是真正的随机数也会产生一定的自相关性。
观察如下随机数生成的一个白噪声序列

```
y <- ts(rnorm(50))
autoplot(y) + ggtitle("White noise")
```

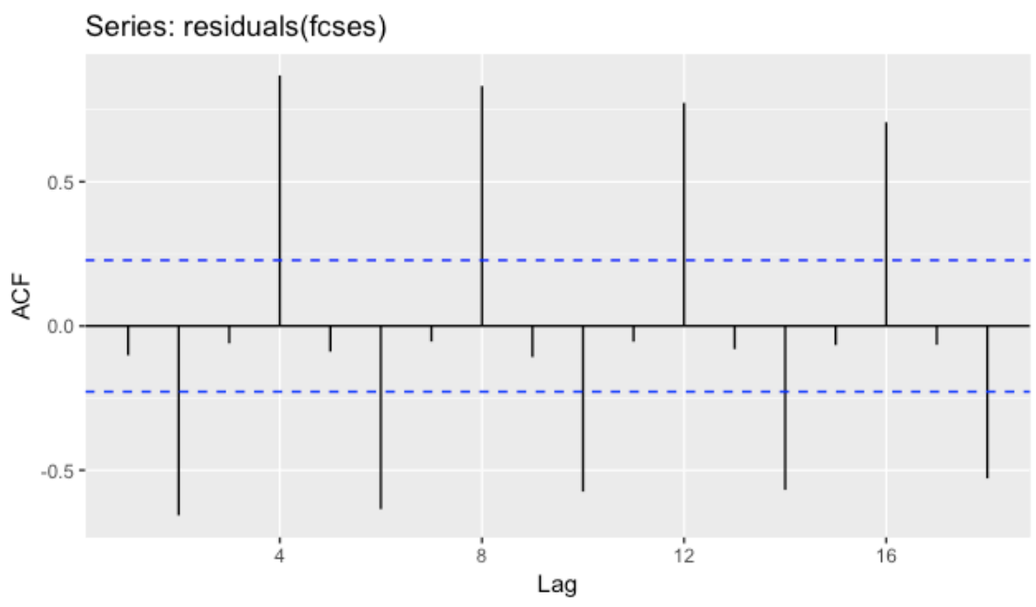



```
ggAcf(y)
```

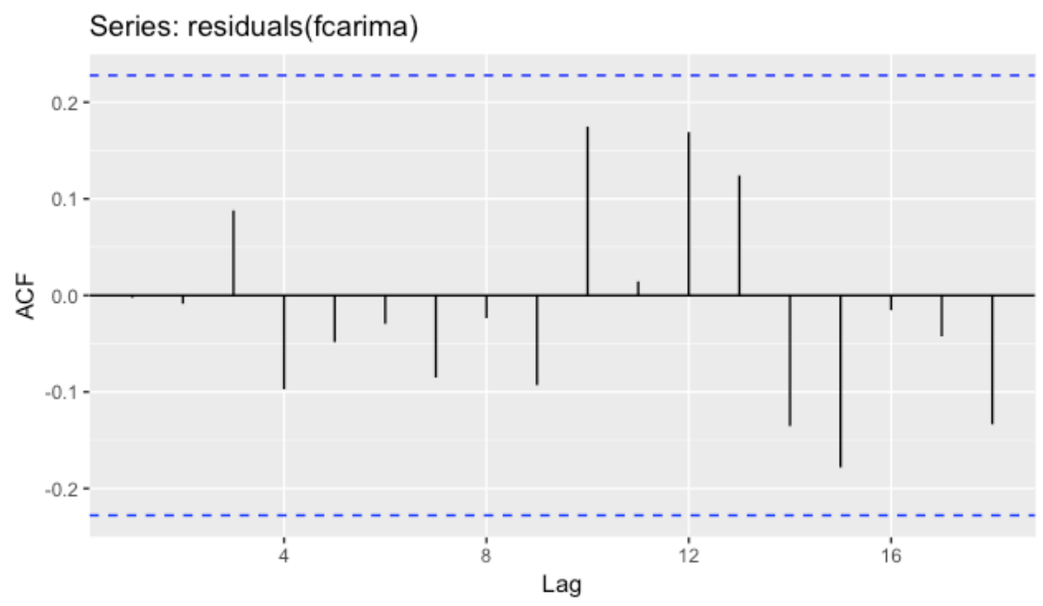


当残差ACF图中仍然有很高自相关性时，也就意味着还有更多规律没有被模型提取出来，如下是简单指数平滑法和优化的arima(滞后移动平均自回归)模型的残差检验

```
fcses=ses(beer2,15)
ggAcf(residuals(fcses))
```




```
fcarima=auto.arima(beer2)
ggAcf(residuals(fcarima))
```



forecast包也开发了完整的残差检验函数chekresiduals，直接将函数作用于训练模型之上

```
checkresiduals(fcses)

checkresiduals(fcarima)
```

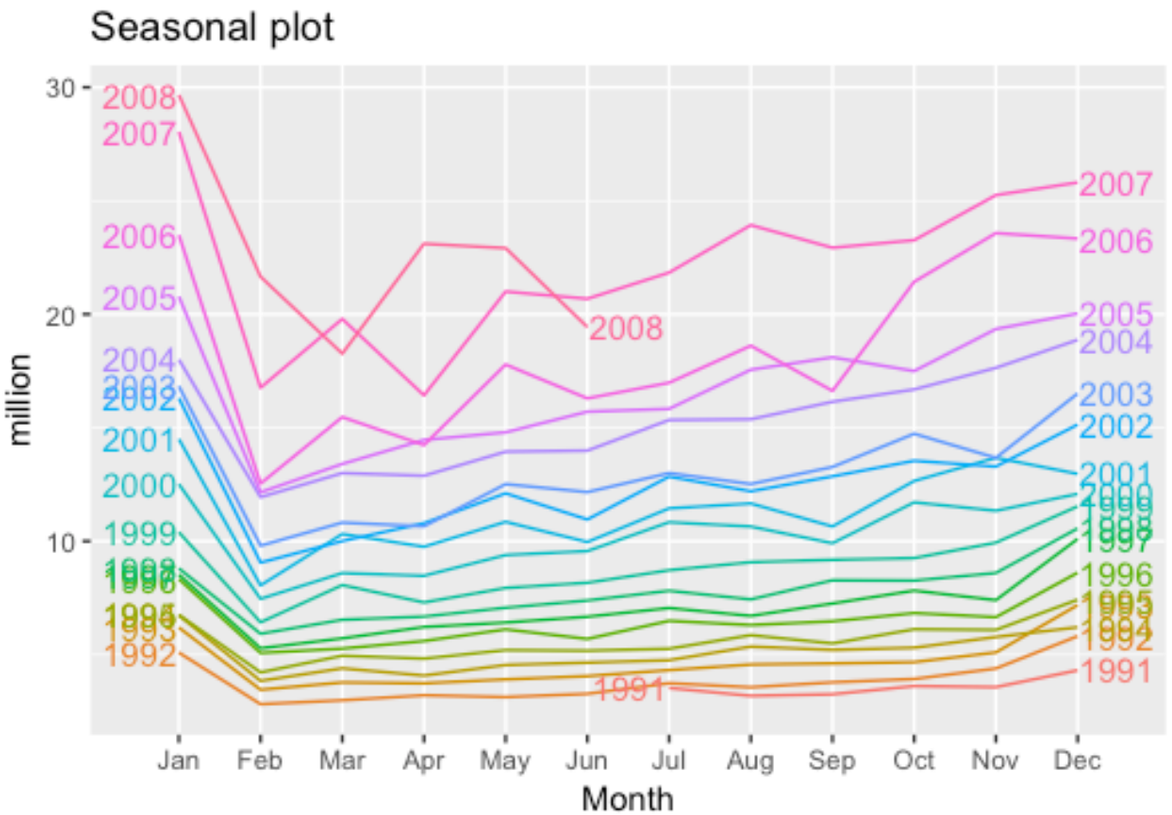
三、季节性特征分析

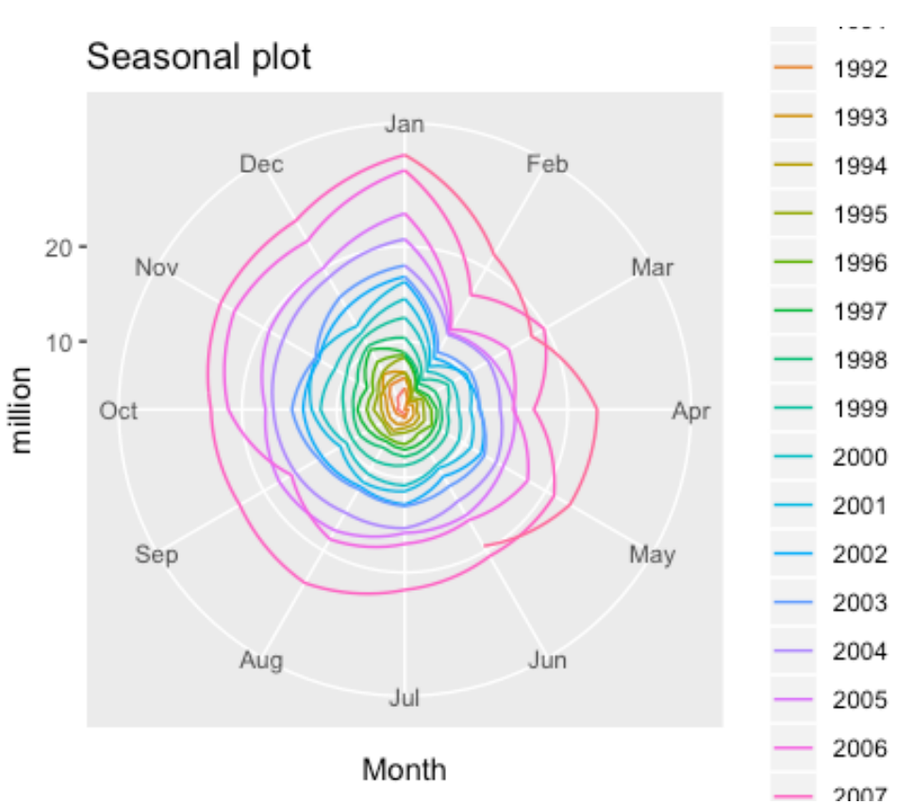
Seasonal（季节变动趋势）是时间序列的一个关键特征，数据在一年内随着月份、周、日发生看似不规则变化，但每年都会重复类似特征。

```
ggseasonplot(a10,year.labels = T,year.labels.left = T)+
ylab('million')+
ggtitle('Seasonal plot')
```

```
ggseasonplot(a10,polar = T)+
ylab('million')+
ggtitle('Seasonal plot')
```

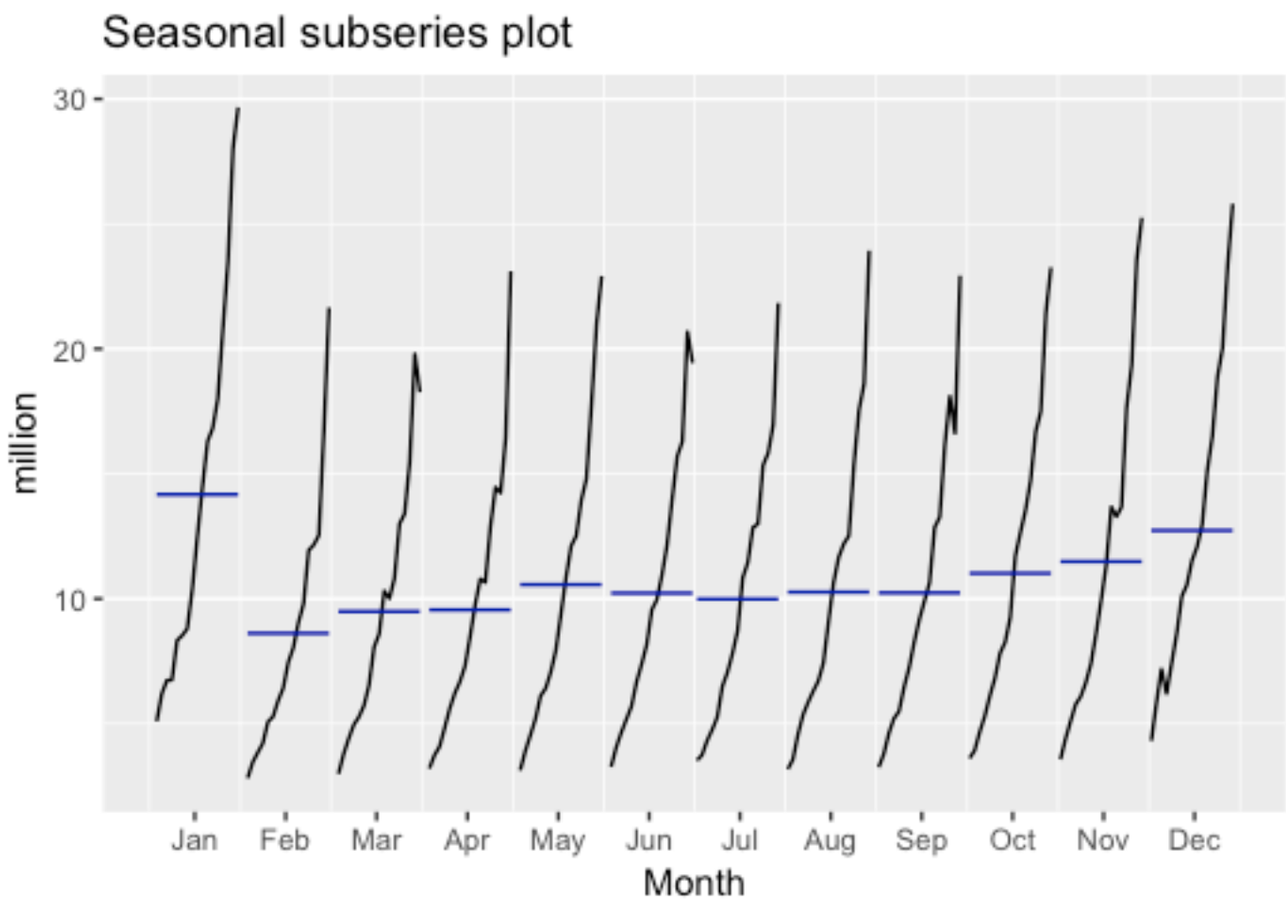
forecast包提供了一套对接ggplot2语法的时间序列专用可视化工具





季节子序列

```
ggsubseriesplot(a10)+
ylab('million')+
ggtitle('Seasonal subseries plot')
```



趋势-周期分解

时间序列的特征可以大致分成如下几类

Trend：长期趋势，记做T

Seasonal：季节变动S

Cyclic：周期趋势C

剩余的特征被作为剩余量记做Remainder，即R

由于C通常长于两年，与T特征可以合并为T-C特征，也简化记为T

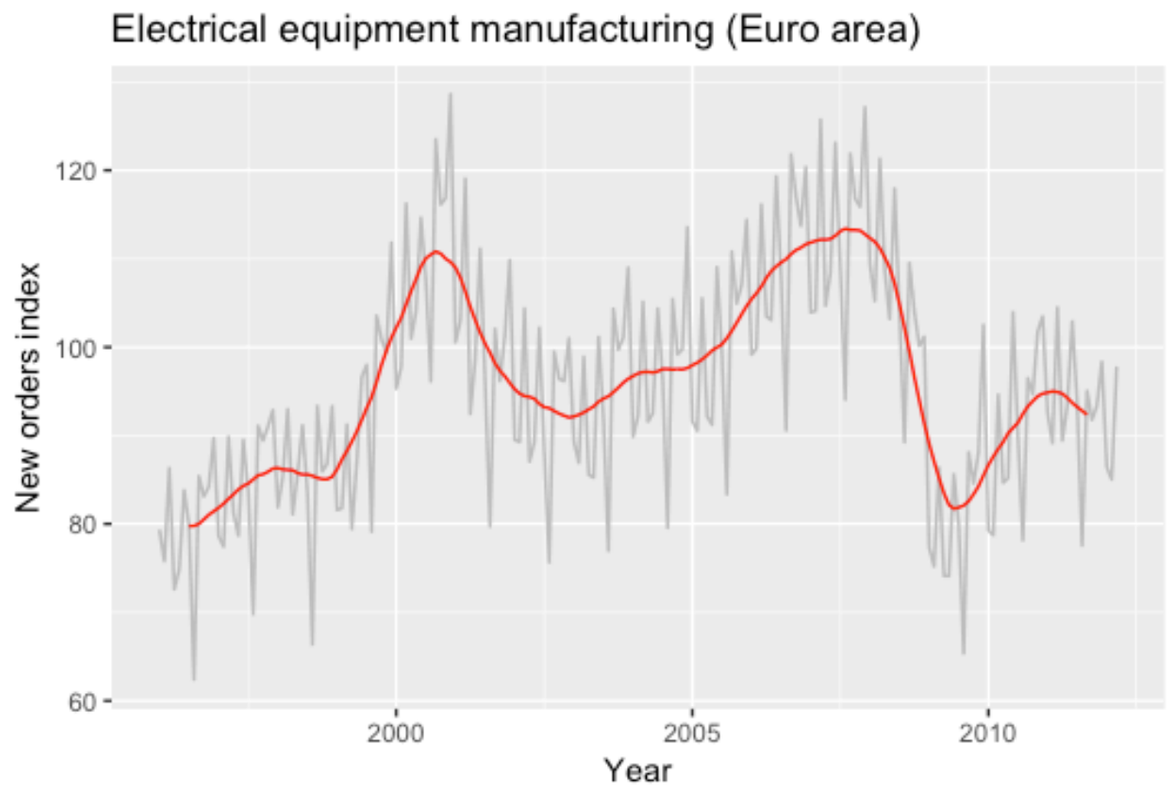
时间序列通常可以分为长期趋势、季节变动和周期趋势，但分解方法则有加法型 $y_t = S_t + T_t + R_t,$

和乘法型 $y_t = S_t \times T_t \times R_t.$

移动平均

移动平均方法能够抹平由于周期带来的数据波动，这中特性为提取趋势带来了方便

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t,$$




```
autoplot(elecequip, series="Data") +
    autolayer(ma(elecequip, 12), series="12-MA") +
    xlab("Year") + ylab("New orders index") +
    ggtitle("Electrical equipment manufacturing (Euro area)") +
    scale_colour_manual(values=c("Data"="grey","12-MA"="red"),
                        breaks=c("Data","12-MA"))
```

decompose函数能够依据加法规则或乘法规则分离提取数据的趋势

```
deseries=decompose(elecequip,type='multiplicative')
autoplot(deseries)
```