# 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

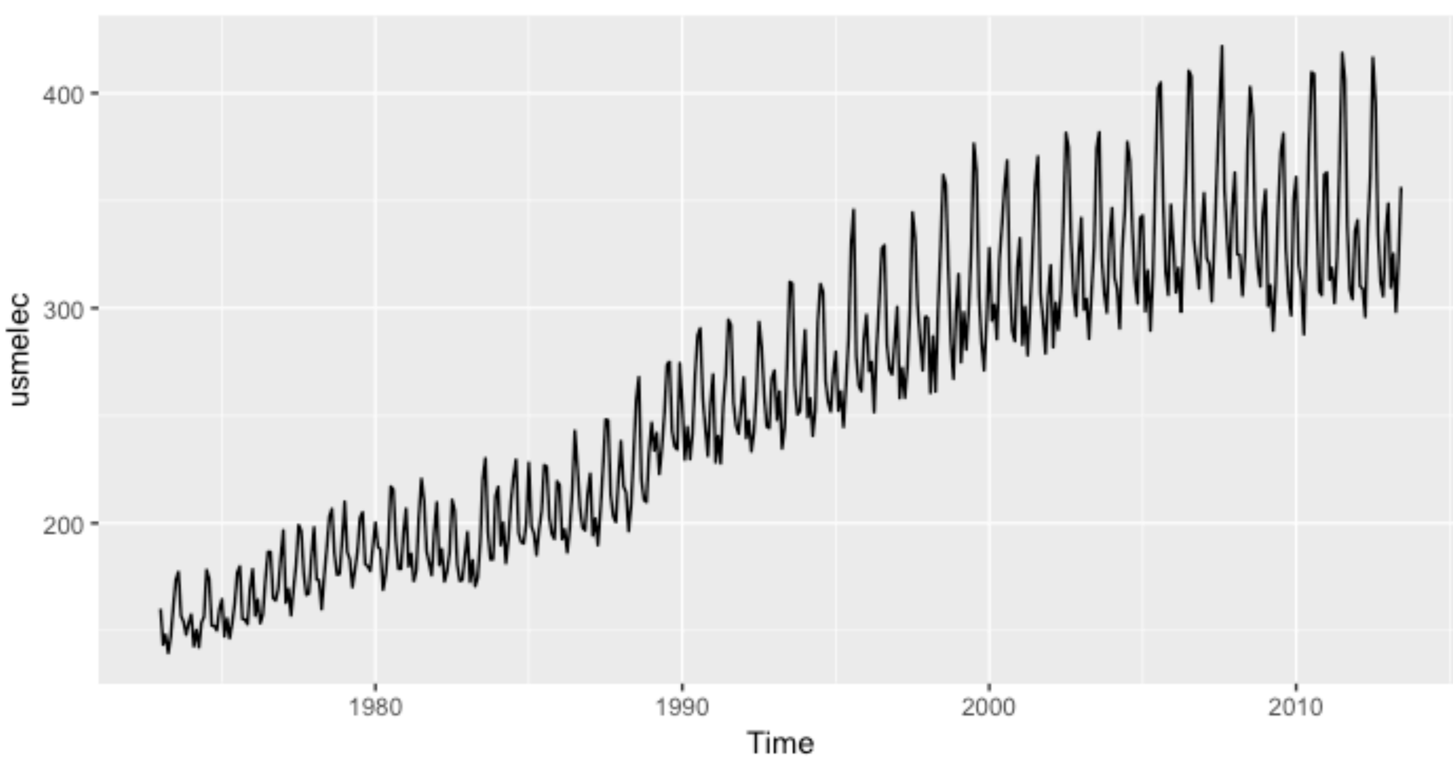
| Data | |
|---|---|
| elecdaily | Time-Series [1:365, 1:3] |
| prison | Time-Series [1:48, 1:32] |
| prisonLF | 1536 obs. of 5 variables |
| uschange | Time-Series [1:187, 1:5] |
| Values | |

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

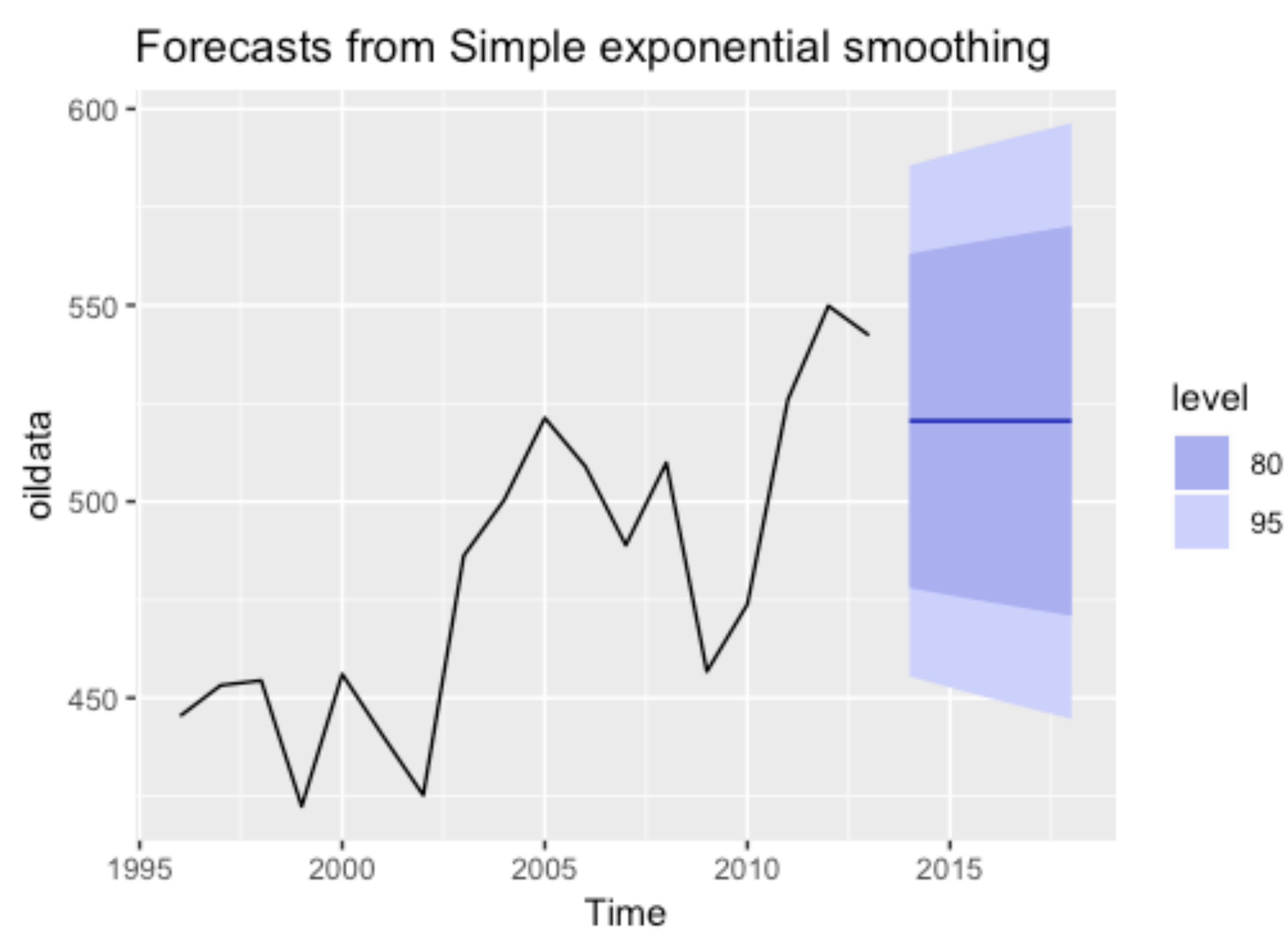
用下边案例说明通常步骤

```
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

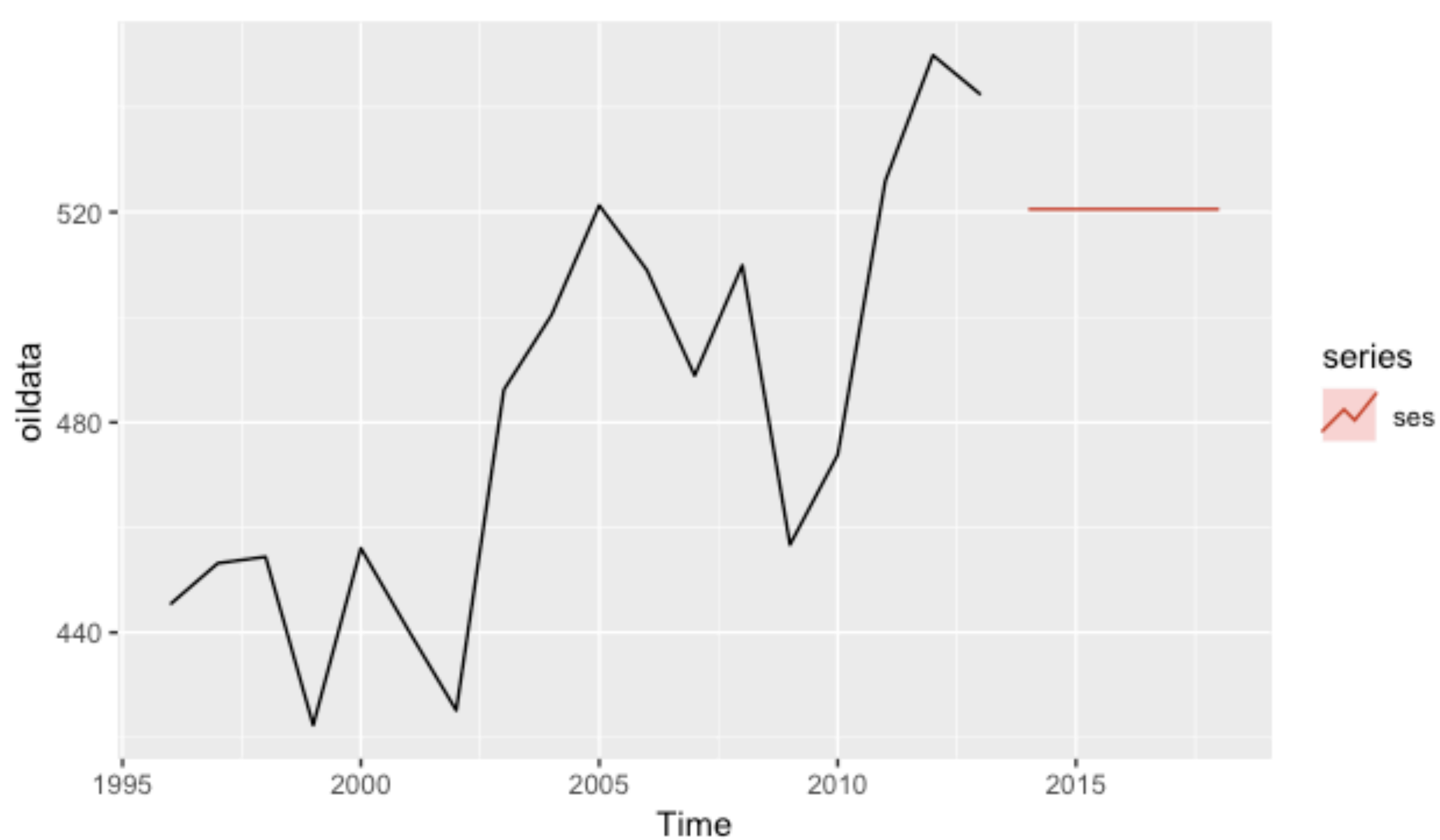
```
> round(accuracy(fc),2)
                ME  RMSE   MAE  MPE MAPE MASE ACF1
Training set 13.02 31.28 25.49 2.34 5.16 1.06 0.35
```

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



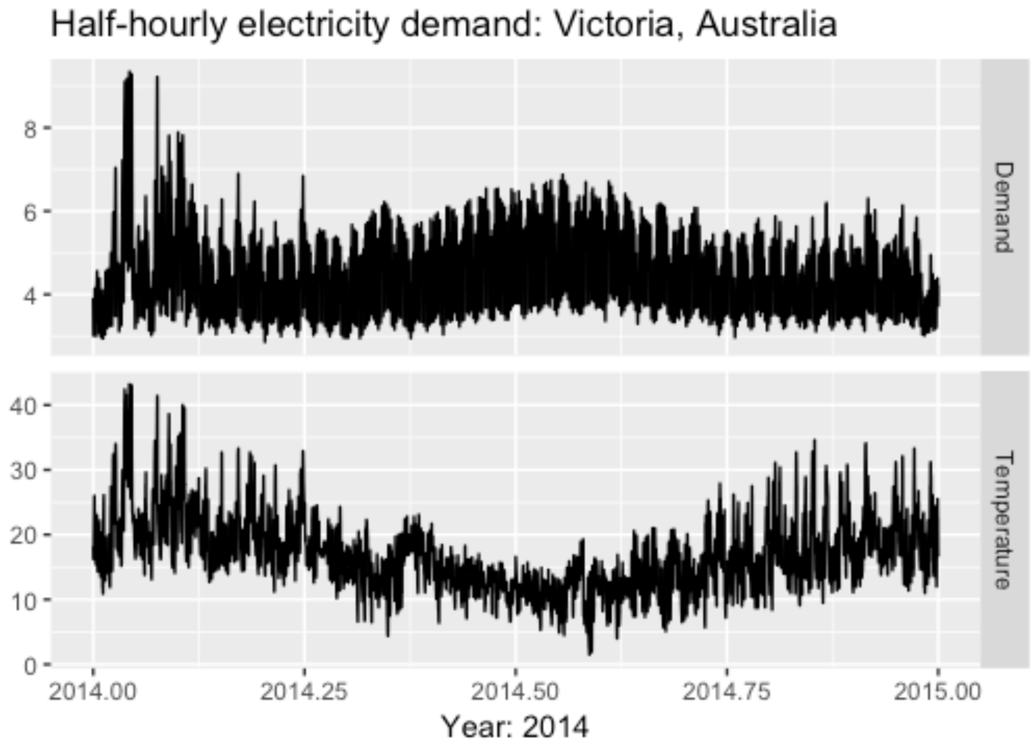
autolayer会为每一个自动图层添加图例，并且会自动识别时间序列和预测模型，按照规范将模型输出在同一个图片中

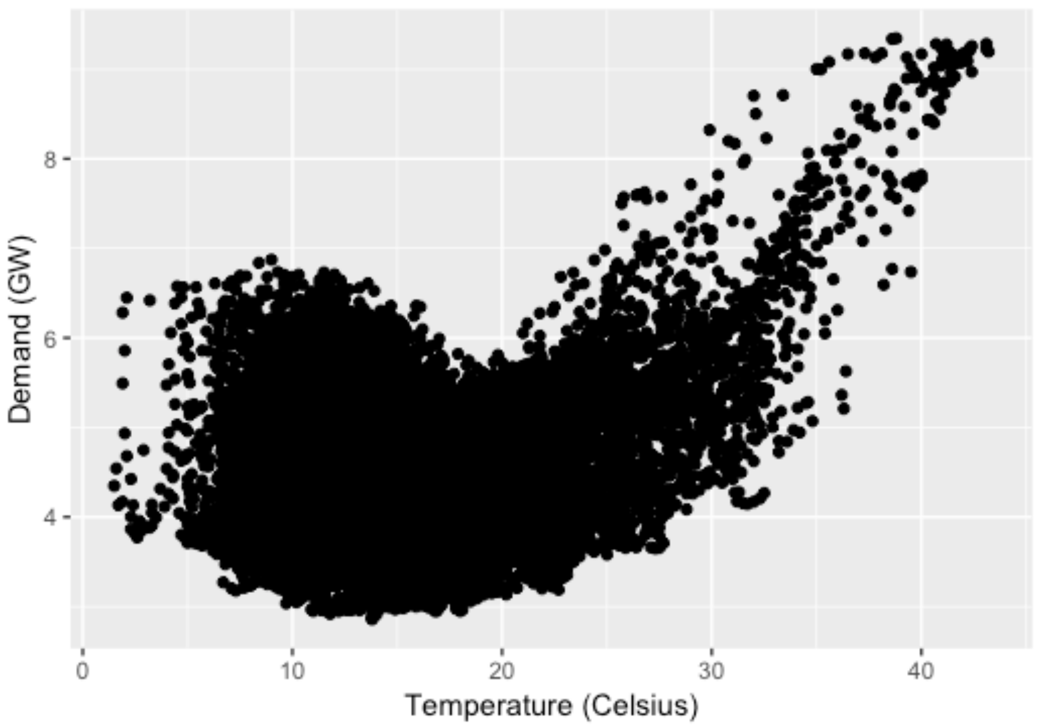
```
autoplot(oildata)+
    autolayer(fc,series='ses',PI=F)+
    guides(colour=guide_legend(title='series'))+
    ylab('Oil(millions of tonnes)')+xlab('Year')+
    ggtitle('Oil production in Saudi Arabia from 1996 to 2013')
```



# 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")
```



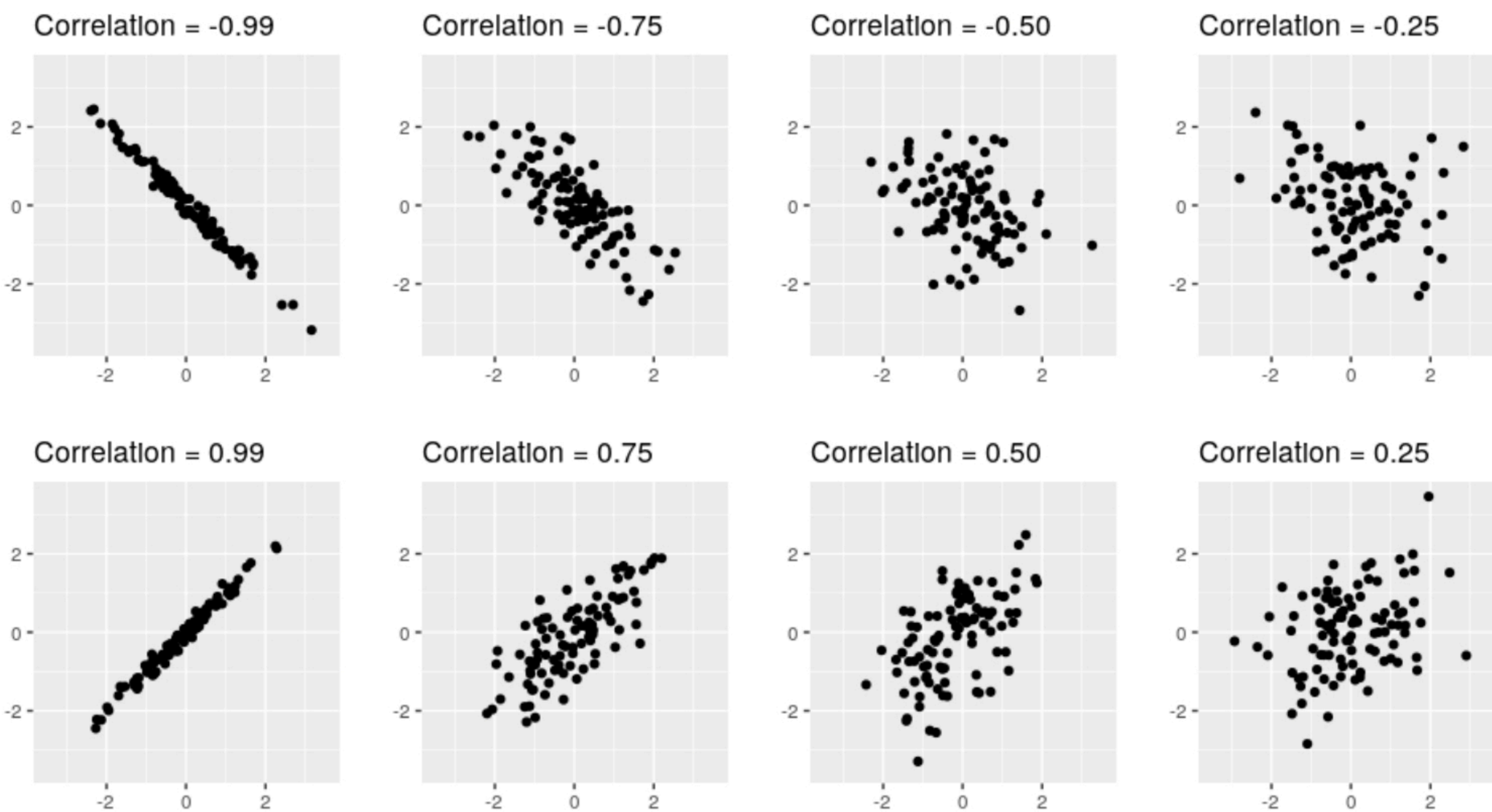




```
qplot(Temperature, Demand, data=as.data.frame(elecdemand)) +
    ylab("Demand (GW)") + xlab("Temperature (Celsius)")
```

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

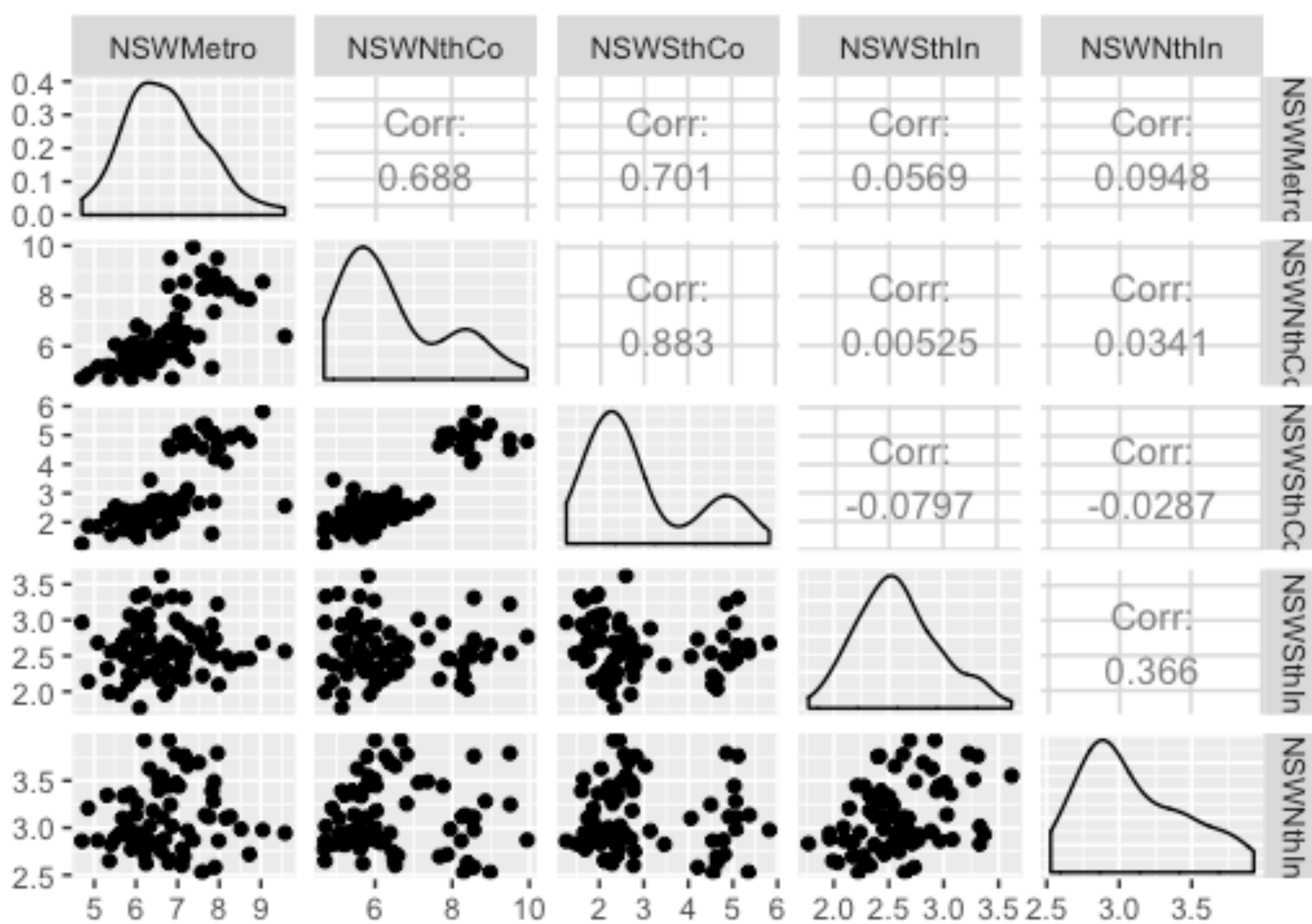
下列用随机数做出了从弱相关到强相关(0.25到0.99)的四类正负相关序列散点图，观察不同系数水平代表的情景。

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



```
autoplot(visnights[,1:5], facets=TRUE) +
    ylab("Number of visitor nights each quarter (millions)")
```

```
library(GGally)

ggpairs(as.data.frame(visnights[,1:5]))
```



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

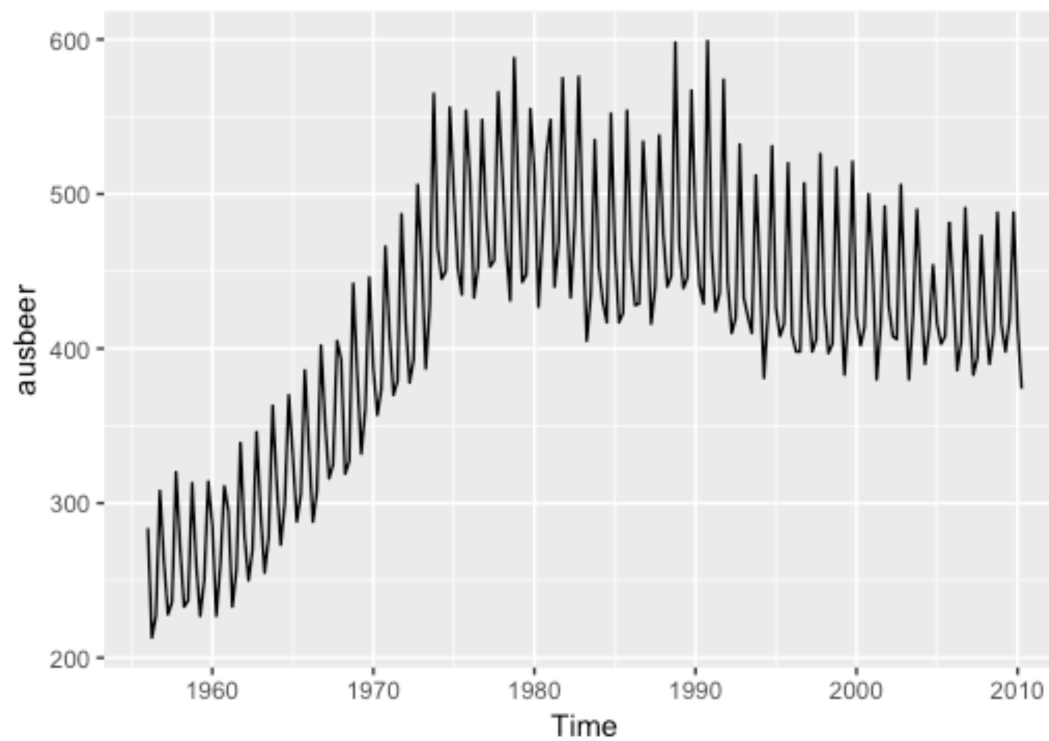
为了凸出季节周期带来的相关性，从ausbeer数据中截出1992年滞后的数据如下

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

数据集具体内容为：

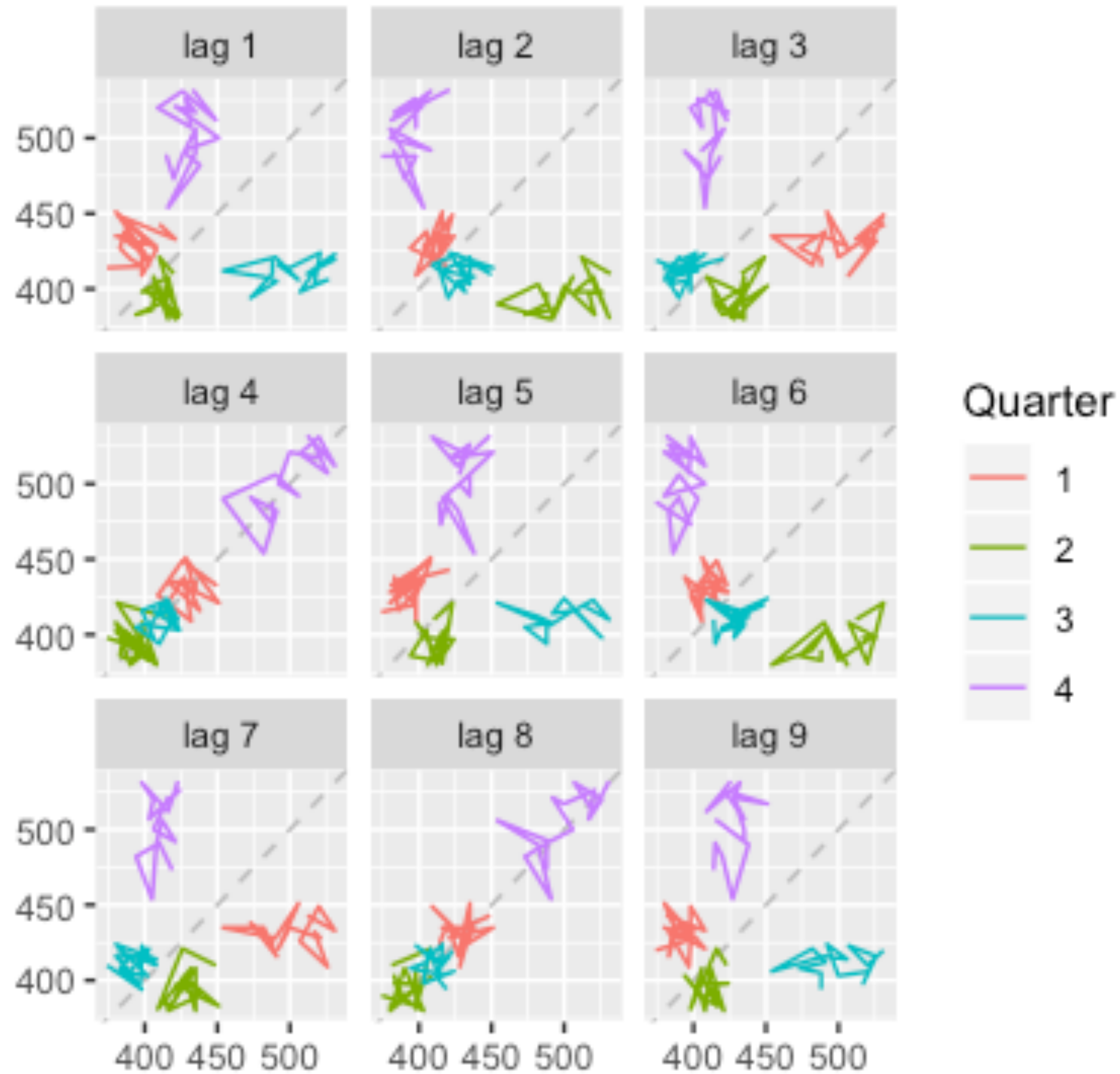
| | Qtr1 | Qtr2 | Qtr3 | Qtr4 |
|---|---|---|---|---|
| 1992 | 443 | 410 | 420 | 532 |
| 1993 | 433 | 421 | 410 | 512 |
| 1994 | 449 | 381 | 423 | 531 |
| 1995 | 426 | 408 | 416 | 520 |
| 1996 | 409 | 398 | 398 | 507 |
| 1997 | 432 | 398 | 406 | 526 |
| 1998 | 428 | 397 | 403 | 517 |
| 1999 | 435 | 383 | 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)
```



```
gglagplot(beer2)
```

每隔4个时间单位就呈现出规则的相关特征




相差4个时间单位数据到底有多大相关性?

ACF图列出了更长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

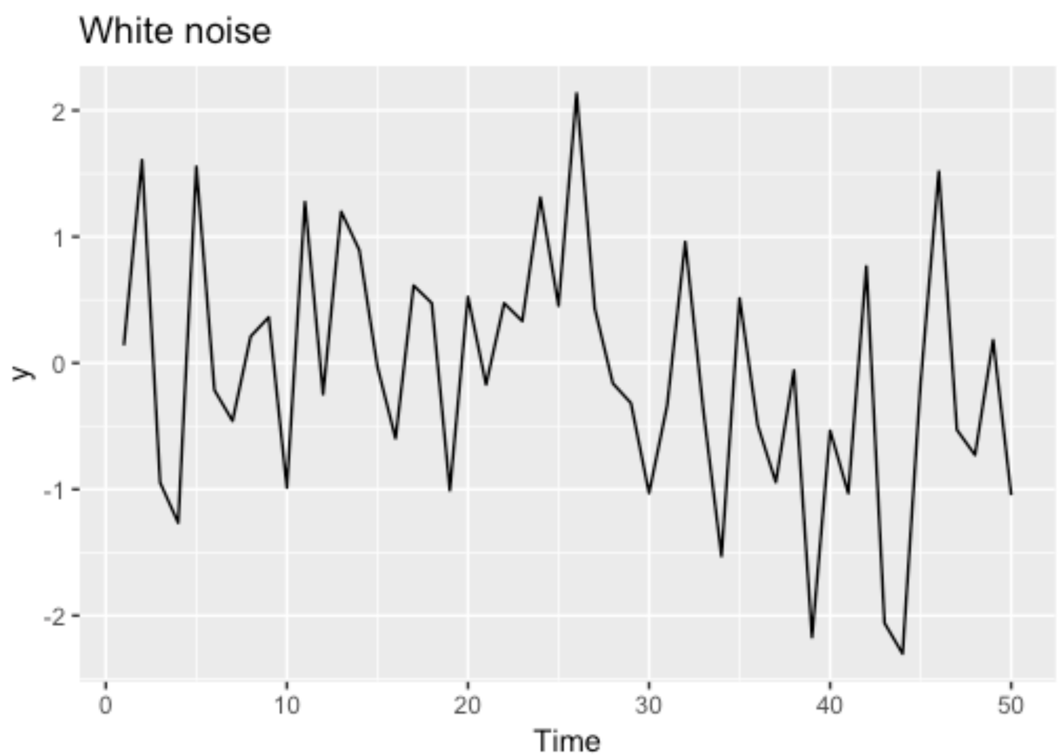
当时间序列中具有明显规则的特征被分解出来之后，剩下的无法解释的部分成为噪声数据，即无法解释的随机事件。

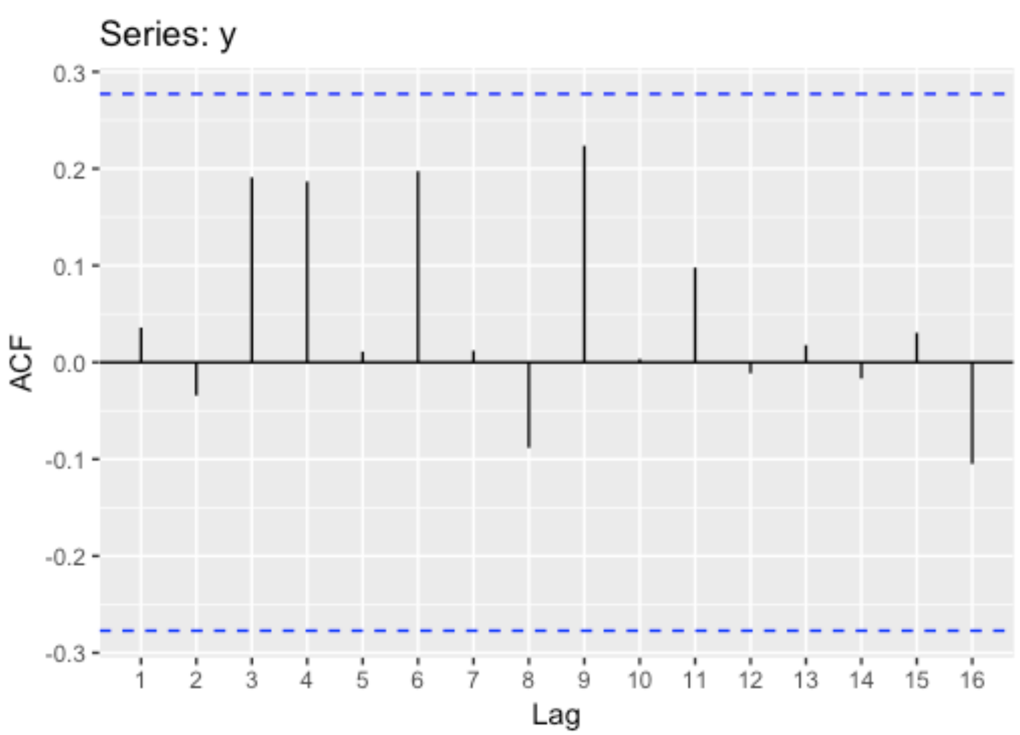
白噪声作为建模分析的剩余部分，最好的状态是不再含有明显的自相关，这可以通过残差的ACF图来检验

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")
```

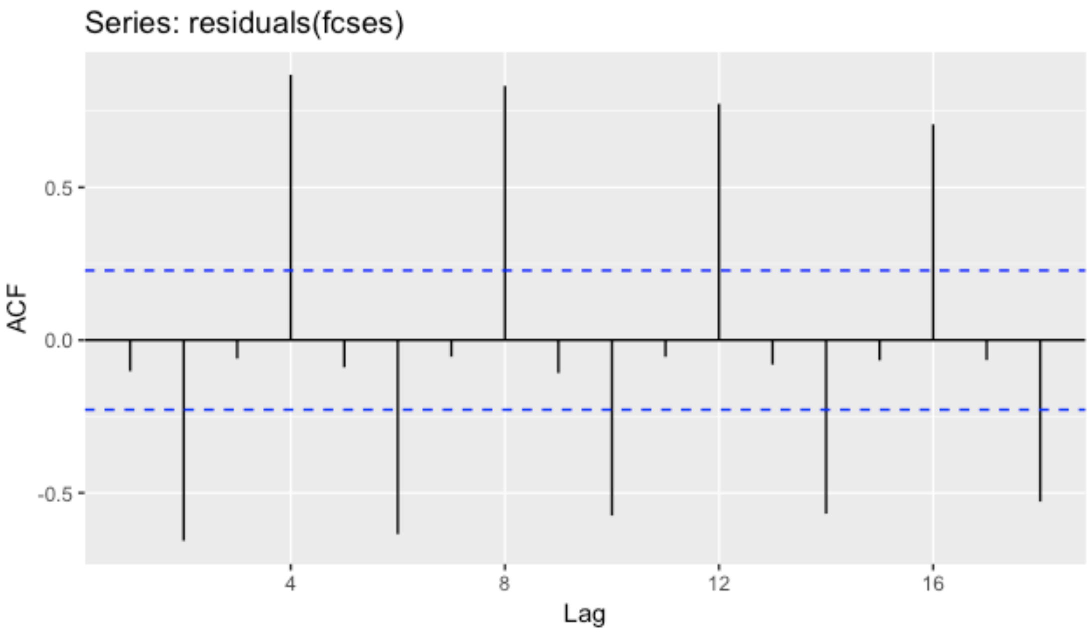
```
ggAcf(y)
```



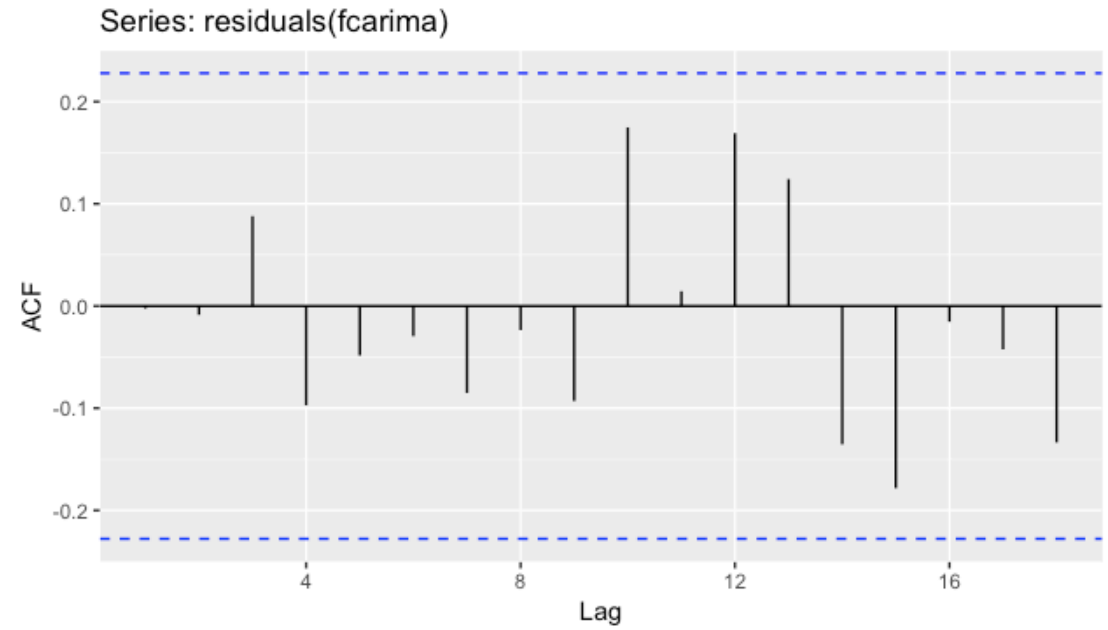



当残差ACF图中仍然有很高自相关性时，也就意味着还有更多规律没有被模型提取出来，如下是简单指数平滑法和优化的arima(滞后移动平均自回归)模型的残差检验

```
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)
```

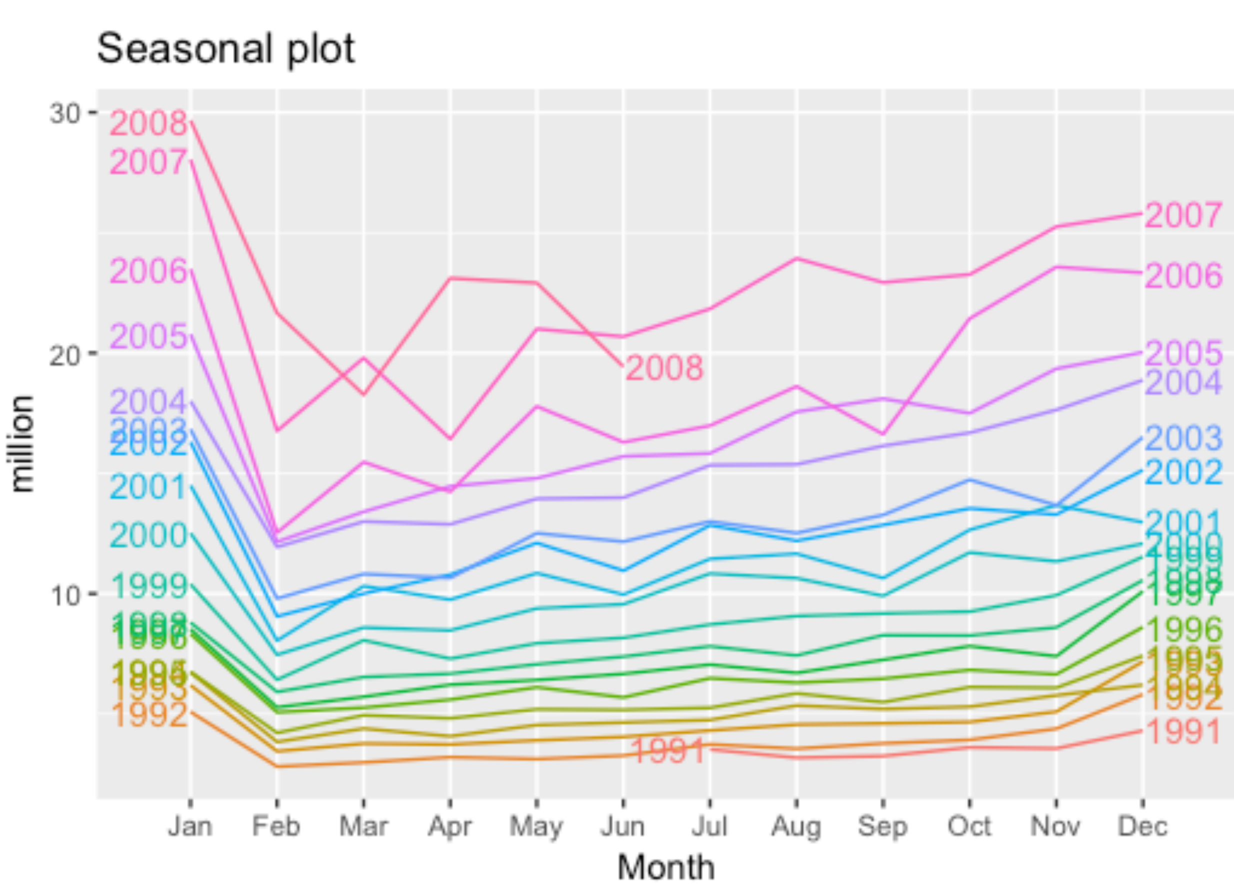
# 三、季节性特征分析

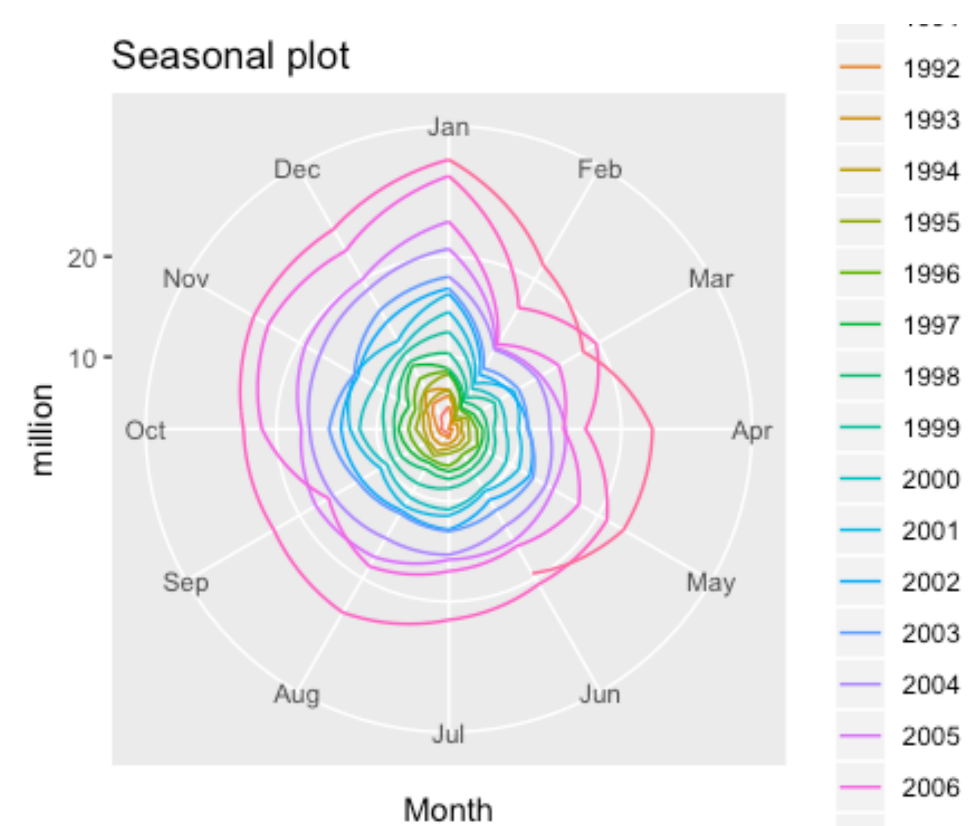
Seasonal（季节变动趋势）是时间序列的一个关键特征，数据在一年内随着月份、周、日发生看似不规则变化，但每年都会重复类似特征。

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

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

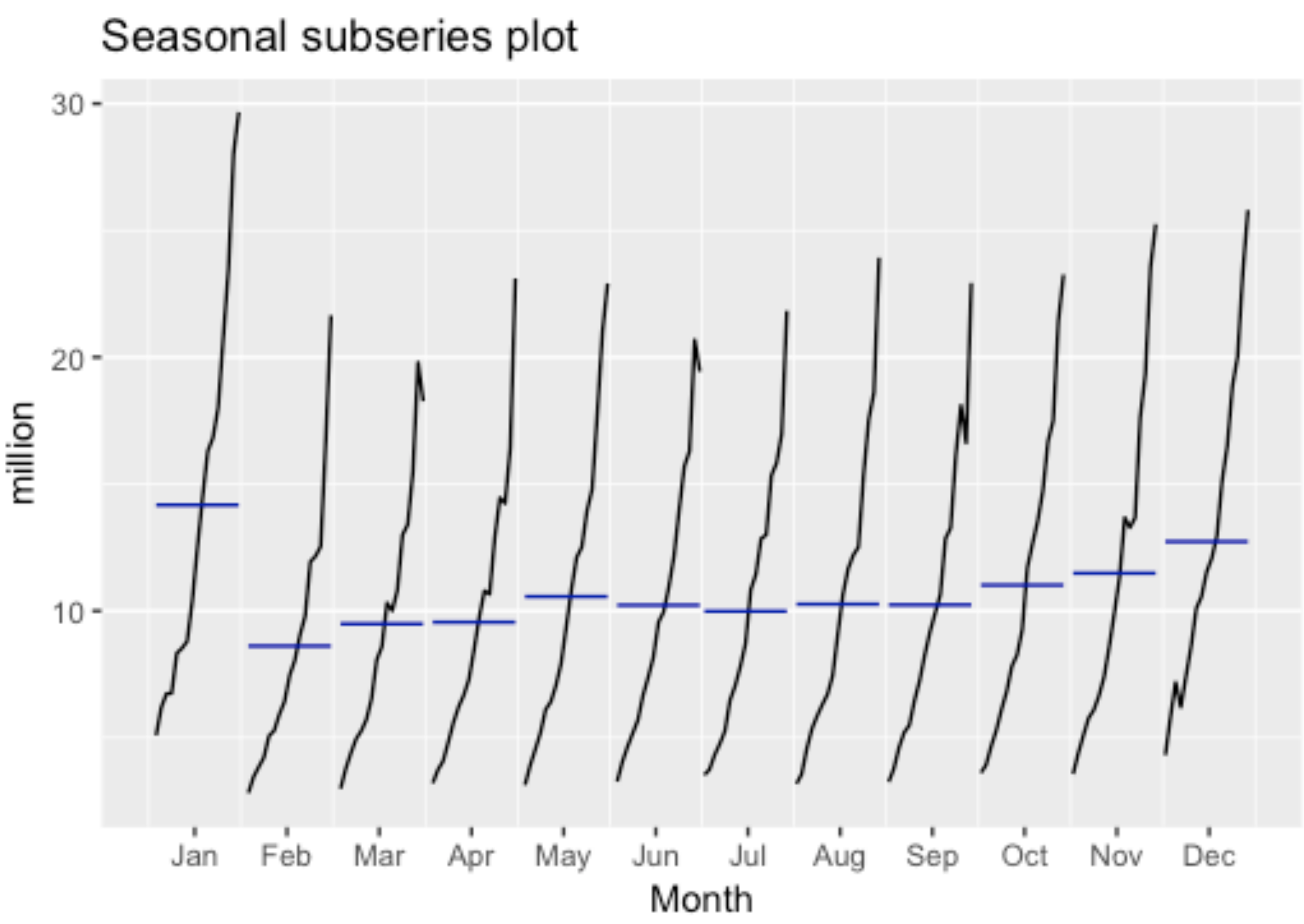
forecast包提供了一套对接ggplot2语法的时间序列专用可视化工具



# 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

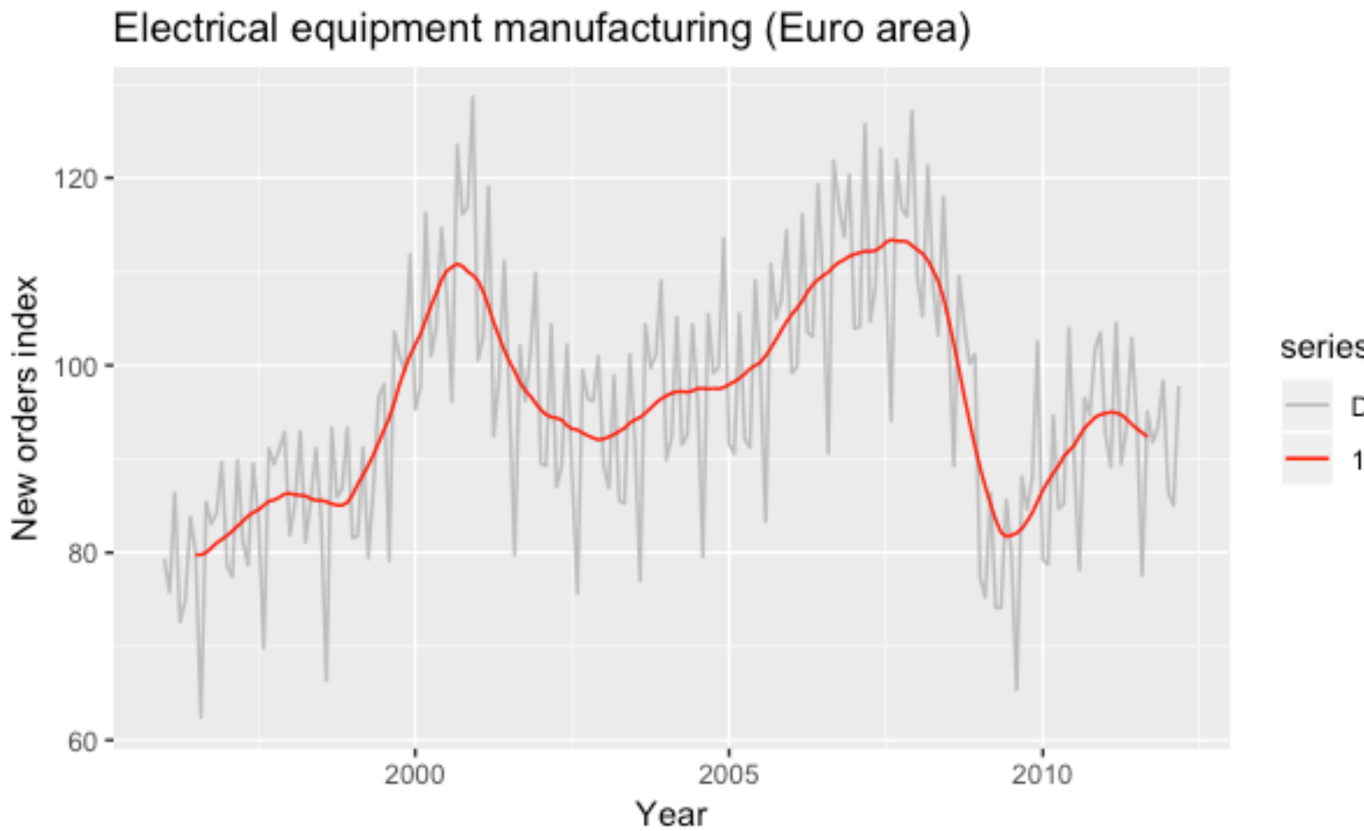
时间序列通常可以分为长期趋势、季节变动和周期趋势，但分解方法则有加法型 $y_t = S_t + T_t + R_t,$

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

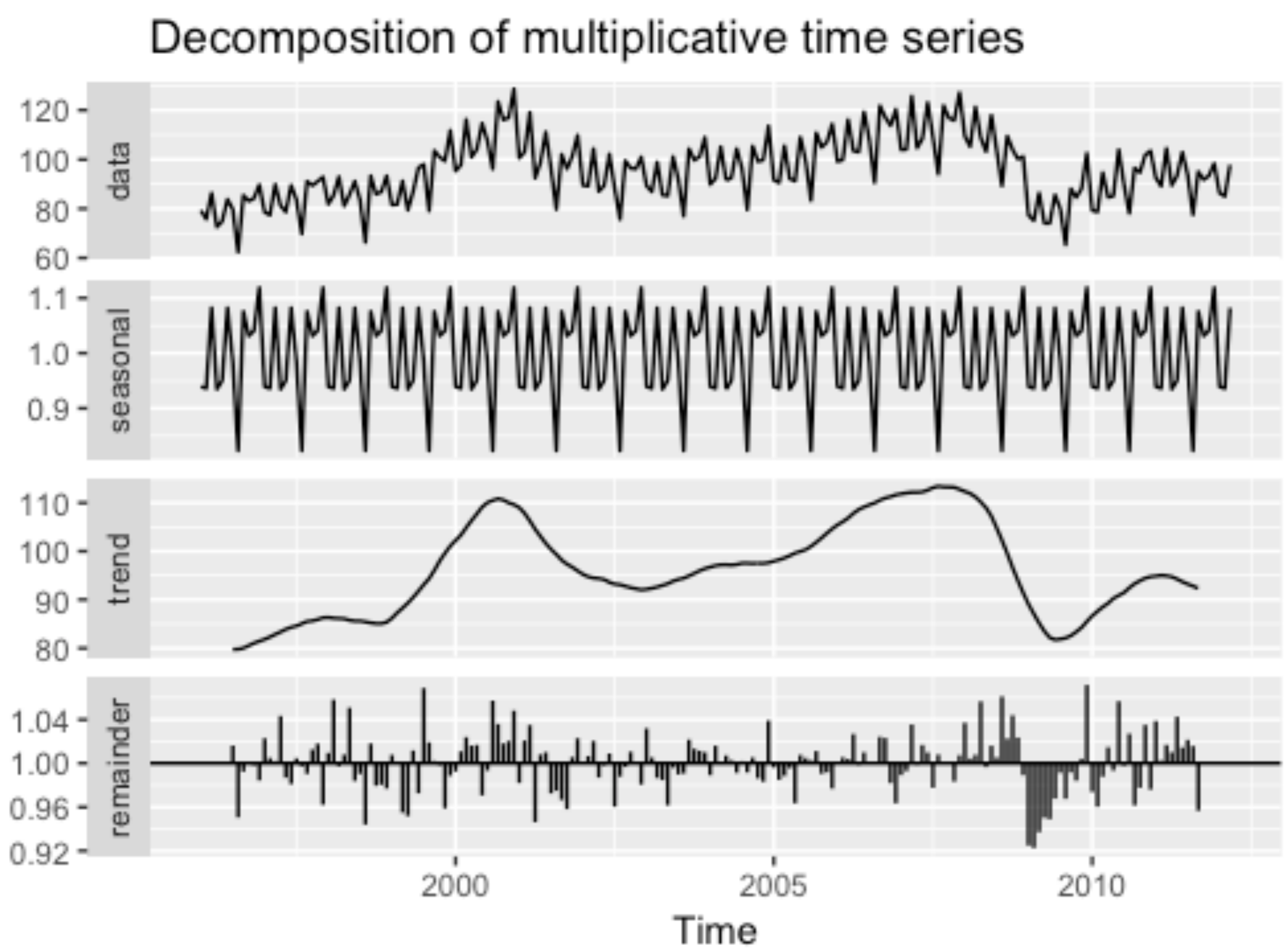
$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t,$$




```
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)
```



# 四、简单预测模型

综合过去所有数据，利用均值做平均是常用的一种预测方法，但也有它明显的局限性

```
> mean(beer2)
[1] 433.5135
> meanf(beer2,1)
        Point Forecast    Lo 80    Hi 80   Lo 95    Hi 95
2010 Q3       433.5135 377.2457 489.7813 346.801 520.2261
```

naive方法则简单用最末值做预测

```
> naive(beer2,1)
```

```
> rwf(beer2,1)
```

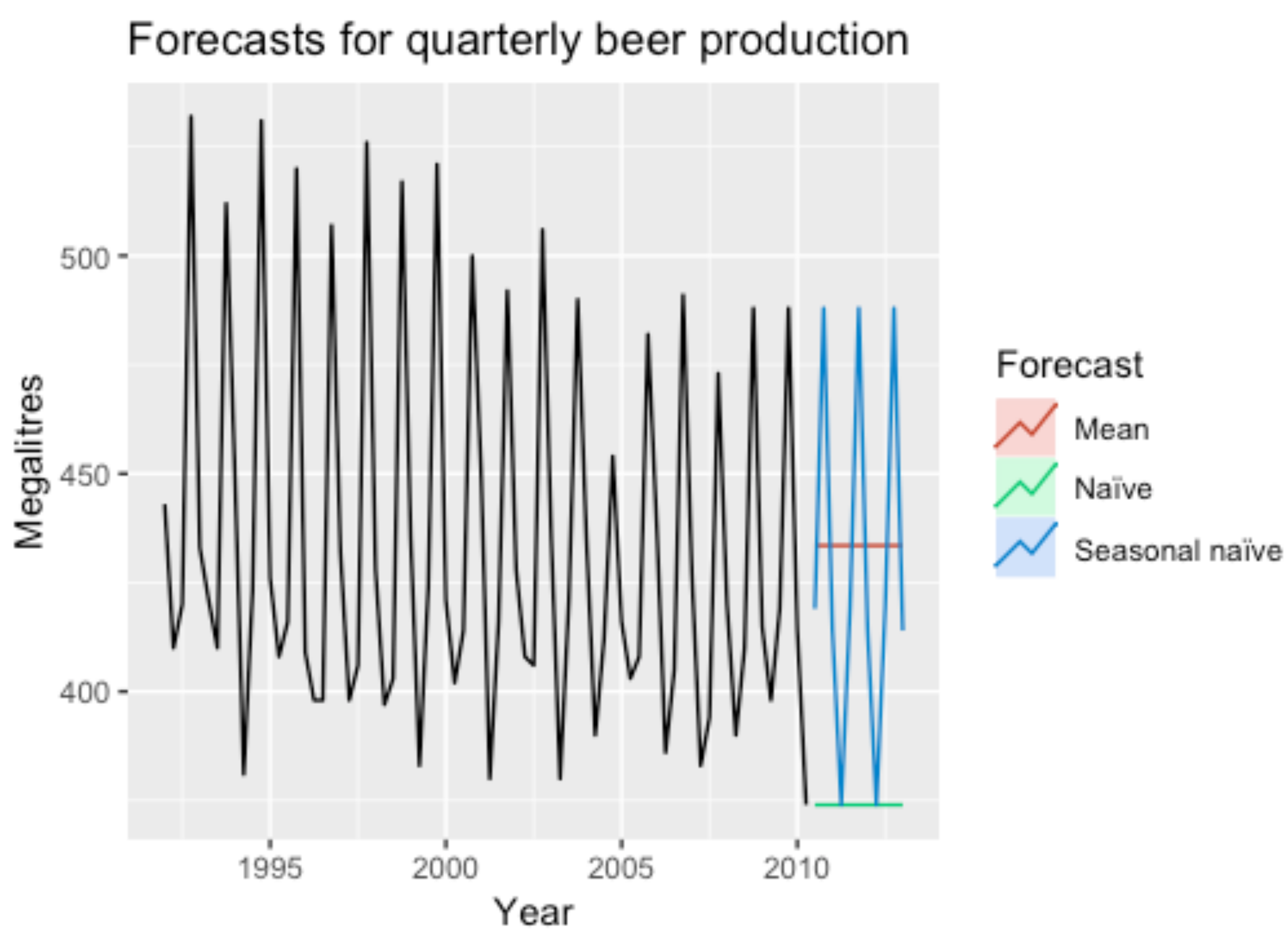
naive方法也叫做random walk forecast

以季节变动为基础的naive方法

```
> snaive(y,1)
```

加入趋势漂移的naive预测

```
> rwf(y,1,drift = T)
```




练习：尝试对fpp2中goog200序列用meanf,naive和趋势漂移的naive方法做预测并做图

```
autoplot(beer2) +
    autolayer(meanf(beer2, h=11),
              series="Mean", PI=FALSE) +
    autolayer(naive(beer2, h=11),
              series="Naïve", PI=FALSE) +
    autolayer(snaive(beer2, h=11),
              series="Seasonal naïve", PI=FALSE) +
    ggtitle("Forecasts for quarterly beer production") +
    xlab("Year") + ylab("Megalitres") +
    guides(colour=guide_legend(title="Forecast"))
```

autoplot只能有一个，下一个自适应图层需要变成autolayer

# 时间项回归

趋势与季节性是时间序列要考虑的首要特征，线性回归可以时间项作为自变量做回归预测，如 $y_t = \beta_0 + \beta_1 t + \varepsilon_t,$

在趋势项基础上加入按日期周期型出现的季节调整项如下

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t,$$

在周期非常明显时，人为做出一系列周期变量数据进入回归建模

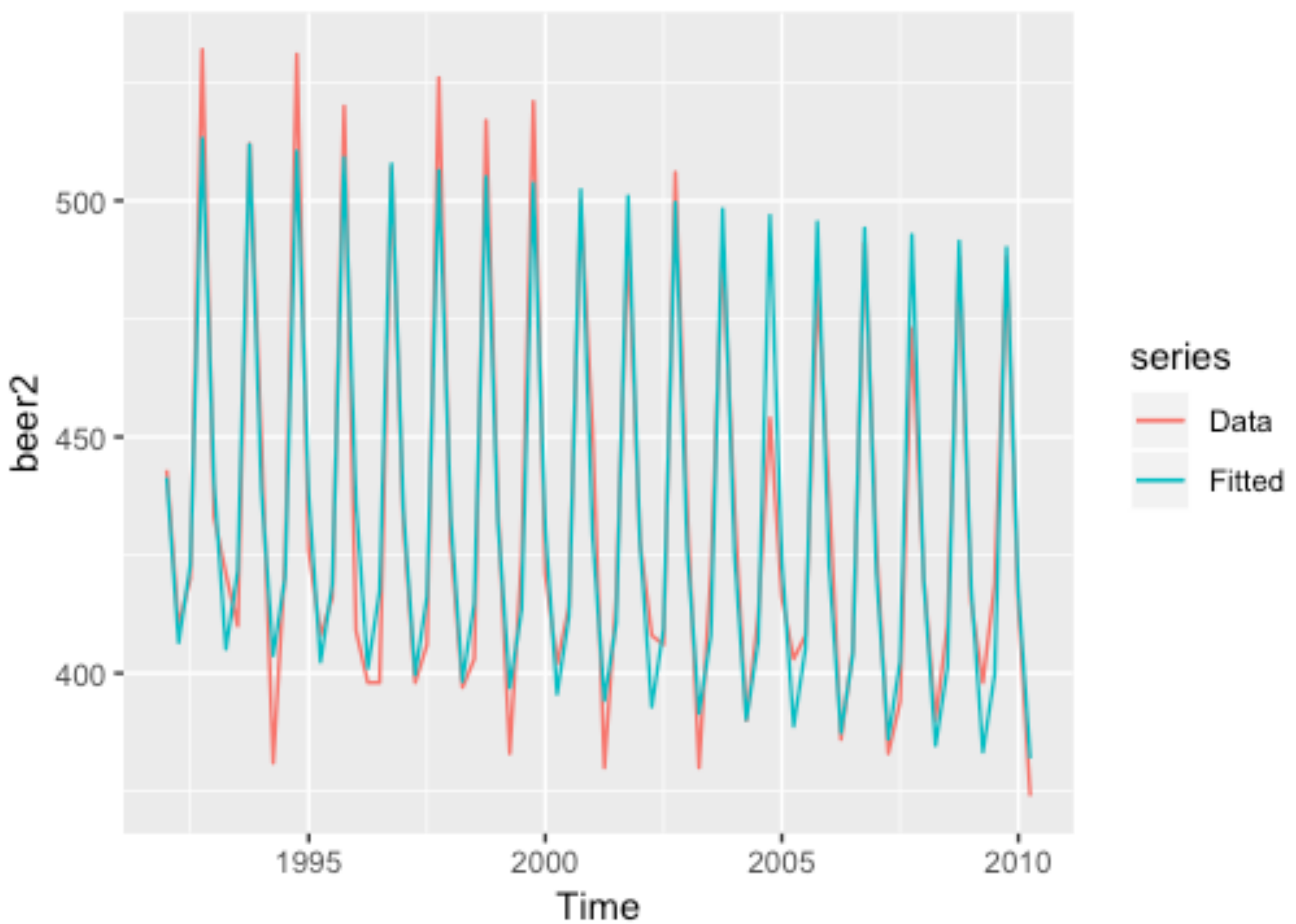
| | $d_{1,t}$ | $d_{2,t}$ | $d_{3,t}$ | $d_{4,t}$ | $d_{5,t}$ | $d_{6,t}$ |
|---|---|---|---|---|---|---|
| Monday | 1 | 0 | 0 | 0 | 0 | 0 |
| Tuesday | 0 | 1 | 0 | 0 | 0 | 0 |
| Wednesday | 0 | 0 | 1 | 0 | 0 | 0 |
| Thursday | 0 | 0 | 0 | 1 | 0 | 0 |
| Friday | 0 | 0 | 0 | 0 | 1 | 0 |
| Saturday | 0 | 0 | 0 | 0 | 0 | 1 |
| Sunday | 0 | 0 | 0 | 0 | 0 | 0 |
| Monday | 1 | 0 | 0 | 0 | 0 | 0 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

线性回归+周期项的方法在forecast包中有对应工具，省去了手动设置周期数据的麻烦

```
fit=tslm(beer2~trend+season)
```

做出图形对比

```
autoplot(beer2, series="Data") +
    autolayer(fit$fitted.values, series="Fitted")
```



# 五、时间序列模型

## 指数平滑法

移动平均实际将所有参与平滑的数据当作相等作用看待，而naive方法则认为最新的数据会最接近未来预测值，结合两者想法另最末的数据权重高，越远的数据权重越低，做出一种变权平均的效果。

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

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

截取oil数据集(年度石油产量数据)1996年后的部分

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

autoplot(fc)
```

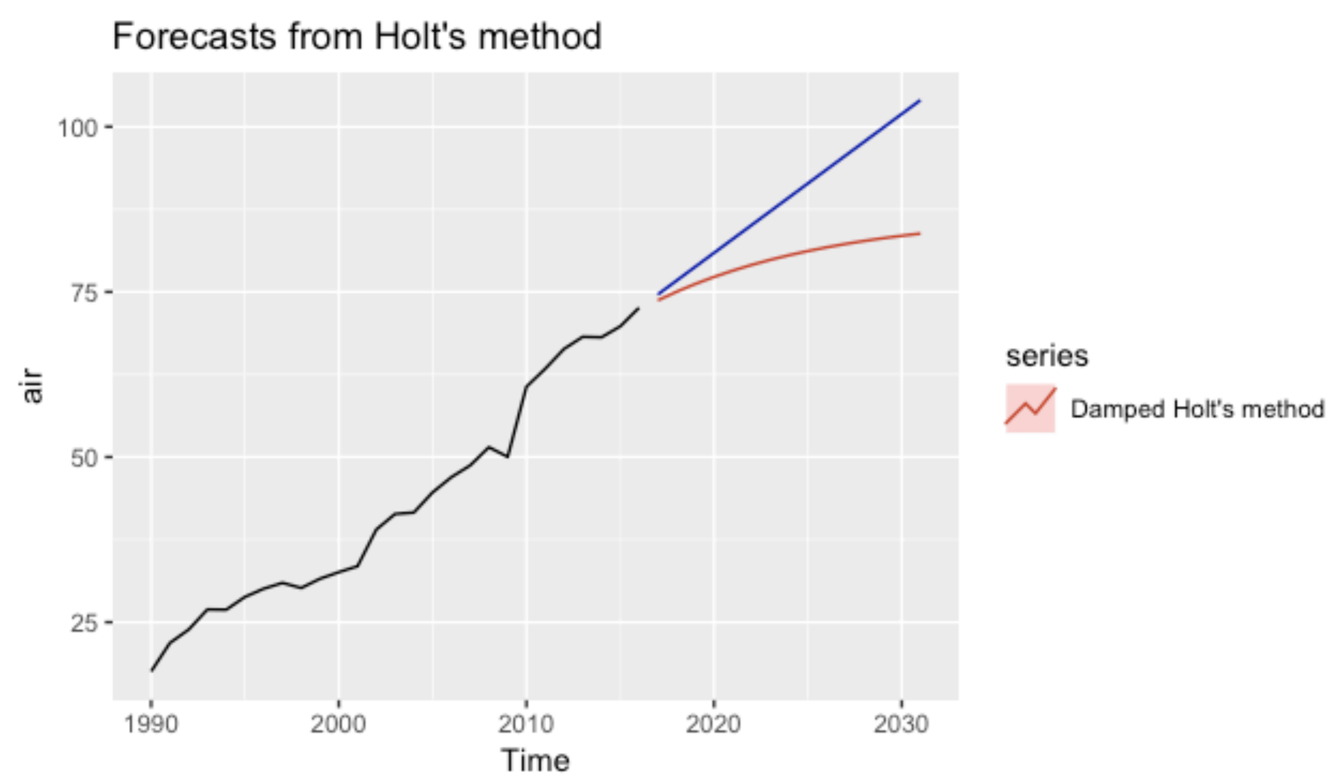
指数平滑法适用于趋势并不太明显的

# 带趋势的指数平滑法——Holt’s 线性趋势法

指数平滑基础上改进的线性趋势法解决带有明显增长趋势的问题，公式如下

$$\hat{y}_{t+h|t} = \ell_t + hb_t$$
$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$

```
air=window(ausair,start=1990)
fc=holt(air,h=5)
```




但holt线性趋势会无限制增长，这不符合常识，任何增长都会遇到瓶颈，然后逐步放缓。阻滞线性趋势模型在holt模型基础上对预测加入了放缓增长因素

```
fc1<- holt(air, h=15)
fc2<- holt(air,damped = T,phi=0.9, h=15)
```

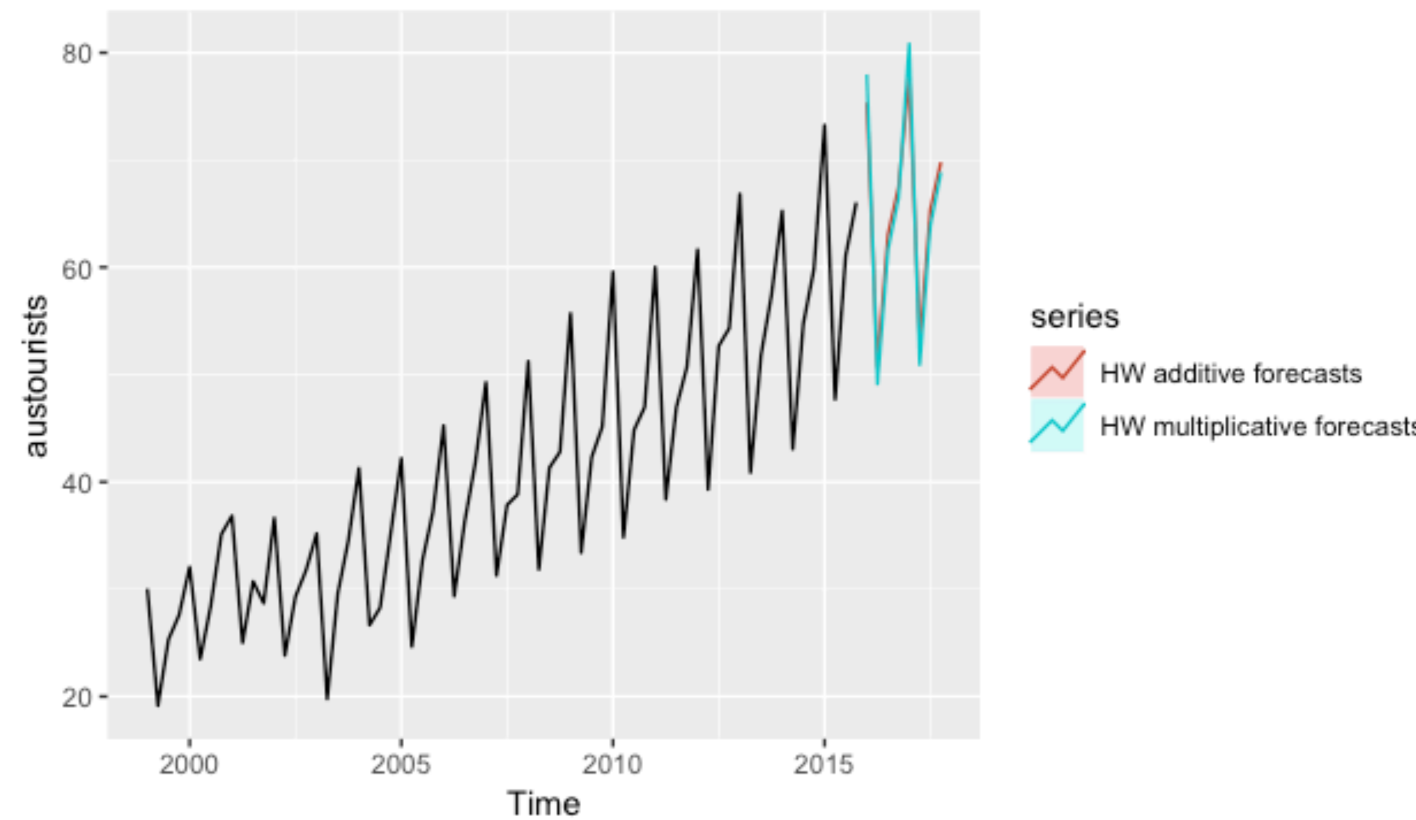
```
autoplot(fc1,series="Holt's method", PI=FALSE) +
    autolayer(fc2, series="Damped Holt's method", PI=FALSE)
```

## Holt-Winter季节模型

Holt-Winter模型在Holt模型基础上解决了数据既带有明显季节性又复合了强烈趋势的问题

Holt-Winter模型需要通过将数据T-S特征分解后建模，从而出现加法型'additive'和乘法型'multiplicative'两种模型

```
fit1=hw(austourists,seasonal = 'additive')
fit2=hw(austourists,seasonal = 'multiplicative')
```




```
autoplot(austourists)+
    autolayer(fit1, series="HW additive forecasts", PI=FALSE) +
    autolayer(fit2, series="HW multiplicative forecasts",PI=FALSE)
```

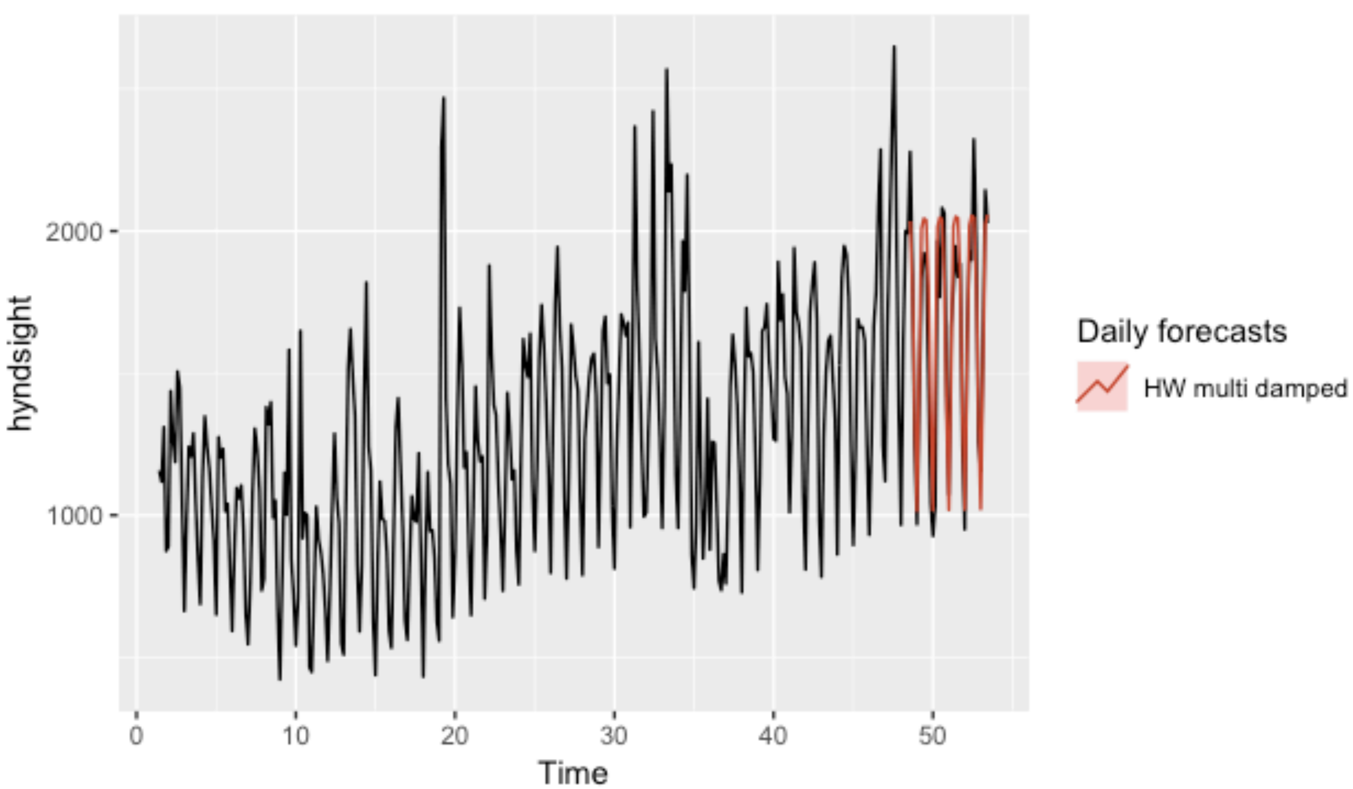
## 带阻滞的Holt-Winter季节模型

Holt-Winter模型同样集成了Holt模型的阻滞增长特点，在hw函数中存在与holt模型同样的阻滞逻辑参数

```
fc <- hw(subset(hyndsight,end=length(hyndsight)-35),
         damped = TRUE, seasonal="multiplicative", h=35)
```

为了检验预测效果，我们空出35个数据，其余数据用于训练模型

```
autoplot(hyndsight) +
    autolayer(fc, series="HW multi damped", PI=FALSE)+
    guides(colour=guide_legend(title="Daily forecasts"))
```



# 差分移动平均自回归模型-Arima

前述模型都是建立在趋势较为明显的基础上，当趋势越来越复杂，直接在原始序列上做任何模型都失去了意义，并且T-C特征也无法再混合在一起

为了能预测复杂趋势特征，我们需要更多的观察角度去找到可描述的趋势特征。为此，转向在数据的差分上做分析，即前后数据之差，也叫做随机游走(random walk)

原序列

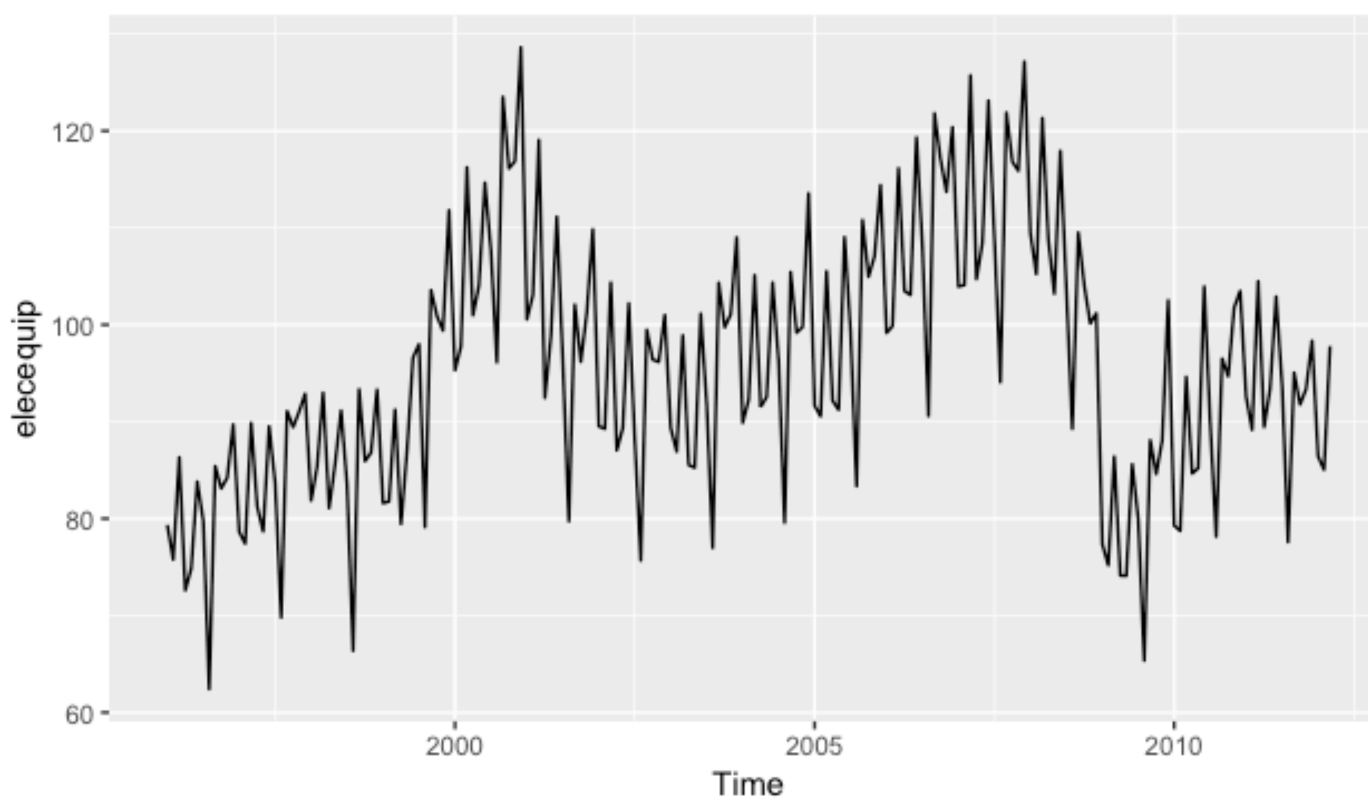
$y_t$

一阶差分量

$y'_t = y_t - y_{t-1}$

二阶差分量

$y''_t = y'_t - y'_{t-1}$




思考差分和导数的关系

随机漫步random walk

一阶平稳　一阶平稳对应了前变用到的所有模型

一阶差分量 $y'_t = y_t - y_{t-1}$ 为了方便记做 $y_t - y_{t-1} = \varepsilon_t$

如果一阶差分序列是平稳的，即 $y_t = y_{t-1} + \varepsilon_t$ 呈现出高度相关或者加入常数后 $y_t - y_{t-1} = c + \varepsilon_t$ or $y_t = c + y_{t-1} + \varepsilon_t$ 残差也是高度相关，则意味着序列趋势是增长或者下降。如果在一阶差分找不到规律，则需要高阶或者季节性差分中寻找。

## 二阶平稳

当我们在一阶序列中找不到平稳状态，则进入二阶差分寻找平稳性。

$$\begin{aligned} y_t'' &= y_t' - y_{t-1}' \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2}. \end{aligned}$$

## 季节平稳

除连续的差分外，季节性差分平稳性也是考虑的角度之一。如滞后m期做差分。$y_t' = y_t - y_{t-m}$. 或形式变为 $y_t = y_{t-m} + \varepsilon_t$

# 移动平均模型

Moving average model(简称MA)不同于AR用滞后变量做回归，MA用白噪声作为自变量做回归，阶数q指模型中的滞后变量个数。

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

# 自回归模型

Autoregression model(简称AR)利用序列自身的滞后期作为自变量做回归，它的阶数p指模型中的自回归变量个数，记做AR(p)

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

# 差分移动平均自回归模型

ARIMA(p,d,q)模型则是综合了AR和MA模型，其中p为自回归项数、q是移动平均项数，d则是差分阶数

确定各参数最合适的取值是一个不太容易的事情，forecast包中给出一个自动定参数的auto.arima函数，按照数据特征进行优化个参数。

```
fc=auto.arima(elecequip)
autoplot(forecast(fc,15))
```

arima模型相对于前变的简单模型要复杂的多，auto.arima或arima做出的结果仅是对模型的训练，而非直接给出预测结果，需要用forecast函数再做一次预测，类似于线性回归里的prediction函数