**时间序列预测讲义**

**1.什么是时间序列？**

时间序列简单的说就是各时间点上形成的数值序列，时间序列分析就是通过观察历史数据预测未来的值。在这里需要强调一点的是，时间序列分析并不是关于时间的回归，它主要是研究自身的变化规律的。

**2.为什么要进行稳定性检验？**

大部分时间序列模型是在假设它是稳定的前提下建立的。直观地说，我们可以这样认为，如果一个时间序列随着时间产生特定的行为，就有很高的可能性认为它在未来的行为是一样的。同时，根据稳定序列得出的理论是更加成熟的， 也是更容易实现与非稳定序列的比较。

在大数定理和中心定理中要求样本同分布（这里同分布等价于时间序列中的平稳性），而我们的建模过程中有很多都是建立在大数定理和中心极限定理的前提条件下的，如果它不满足，得到的许多结论都是不可靠的。

**3.主要考虑的因素？**

**长期趋势(Long-term trend)**

时间序列可能相当稳定或随时间呈现某种趋势。

时间序列趋势一般为线性的(linear)，二次方程式的 (quadratic)或指数函数(exponential function)。

**季节性变动(Seasonal variation)**

按时间变动，呈现重复性行为的序列。

季节性变动通常和日期或气候有关。

季节性变动通常和年周期有关。

**周期性变动(Cyclical variation)**

相对于季节性变动，时间序列可能经历“周期性变动”。

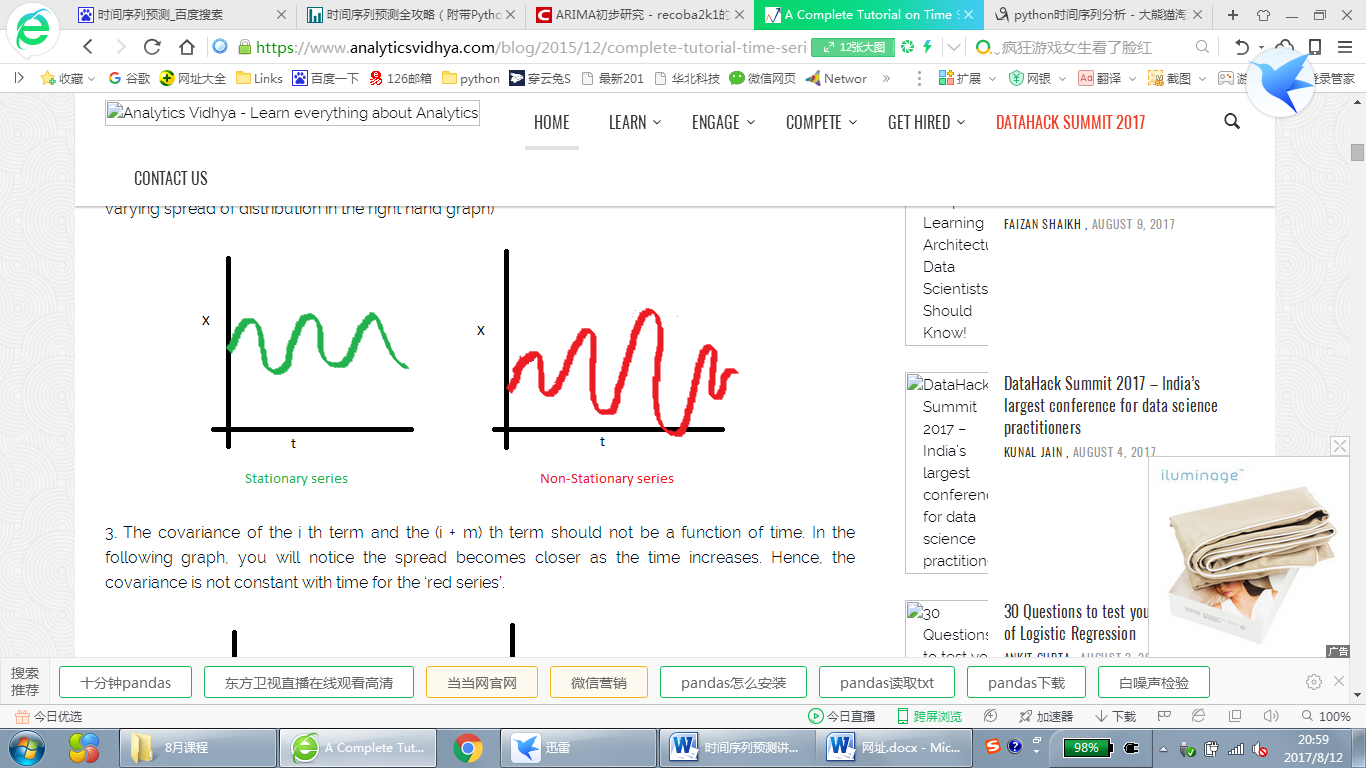
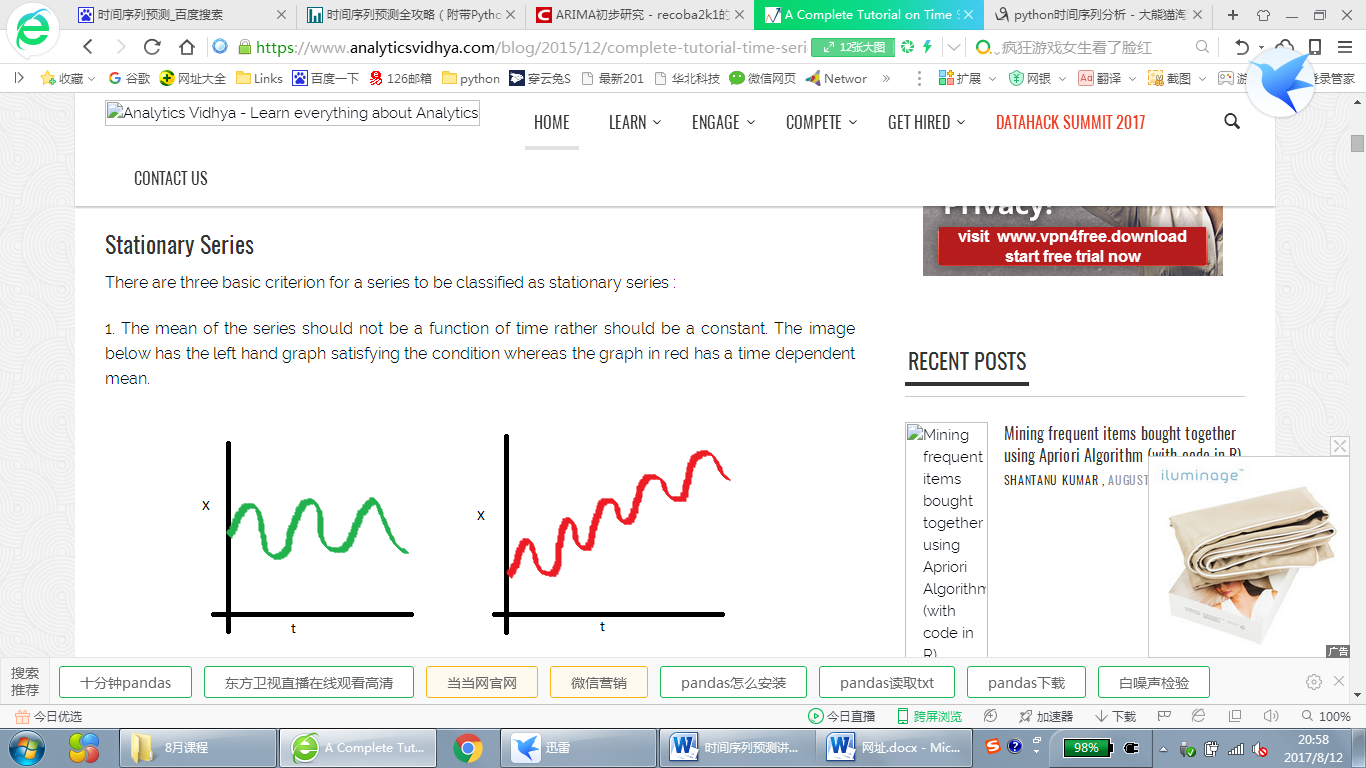
周期性变动通常是因为经济变动。

**随机影响(Random effects)**

**4.什么情况下就是稳定了？**

如果一个时间序列的统计特征如平均数，方差，协方差随着时间保持不变，我们就可以认为它是稳定的。

平均值 方差



协方差



**5.检验方法**

**1）观察法**——绘制滚动统计：我们可以绘制移动平均数和移动方差，观察它是否随着时间变化。随着移动平均数和方差的变化，我认为在任何“t”瞬间，我们都可以获得去年的移动平均数和方差。如：上一个12个月份。但是，这更多的是一种视觉技术。

通俗的说就是通过观察序列的趋势图与相关图是否随着时间的变化呈现出某种规律。所谓的规律就是时间序列经常提到的周期性因素，现实中遇到得比较多的是线性周期成分，这类周期成分可以采用差分或者移动平均来解决，而对于非线性周期成分的处理相对比较复杂，需要采用某些分解的方法。下图为航空数据的线性图，可以明显的看出它具有年周期成分和长期趋势成分。平稳序列的自相关系数会快速衰减，下面的自相关图并不能体现出该特征，所以我们有理由相信该序列是不平稳的。

**2）ADF检验**：这是一种检查数据稳定性的统计测试。无效假设：时间序列是不稳定的。测试结果由测试统计量和一些置信区间的临界值组成。如果“测试统计量”小于“临界值”，我们可以拒绝无效假设，并认为序列是稳定的。

# 当p-value<0.05，且TestStatistic显著小于Critical Value (5%)时，数列稳定

ADF是一种常用的单位根检验方法，他的原假设为序列具有单位根，即非平稳，对于一个平稳的时序数据，就需要在给定的置信水平上显著，拒绝原假设。ADF只是单位根检验的方法之一，

**6.平稳性处理方法**

a. 对数变换

对数变换主要是为了减小数据的振动幅度，对数变换相当于增加了一个惩罚机制，数据越大其惩罚越大，数据越小惩罚越小。这里强调一下，变换的序列需要满足大于0，小于0的数据不存在对数变换。

b. 平滑法

根据平滑技术的不同，平滑法具体分为移动平均法和指数平均法。

移动平均即利用一定时间间隔内的平均值作为某一期的估计值，而指数平均则是用变权的方法来计算均值，可以对最近的值赋予更高的权重。

c. 差分

时间序列最常用来剔除周期性因素的方法当属差分了，它主要是对等周期间隔的数据进行线性求减。前面我们说过，ARIMA模型相对ARMA模型，仅多了差分操作，ARIMA模型几乎是所有时间序列软件都支持的，差分的实现与还原都非常方便。而statsmodel中，对差分的支持不是很好，它不支持高阶和多阶差分，我们可以先用pandas将序列差分好，然后在对差分好的序列进行ARIMA拟合，只不过这样后面会多了一步人工还原的工作。

d. 分解

所谓分解就是将时序数据分离成不同的成分。statsmodels使用的X-11分解过程，它主要将时序数据分离成长期趋势、季节趋势和随机成分。与其它统计软件一样，statsmodels也支持两类分解模型，加法模型和乘法模型，这里我只实现加法，乘法只需将model的参数设置为"multiplicative"即可。

得到不同的分解成分后，就可以使用时间序列模型对各个成分进行拟合。

**7.ARIMA模型**

实际中单凭对时间序列的观察很难确定其属于哪一种模型，而**自相关函数和偏自相关函数**是分析随机过程和识别模型的有力工具

<https://wenku.baidu.com/view/c5ad131b650e52ea551898f6>

* 自回归项的数目Number of AR (Auto-Regressive) terms (p)：AR条件仅仅是因变量的滞后。如：如果P等于5，那么预测x（t）将是x（t-1）。。。（t-5）。
* 移动平均项的数目Number of MA (Moving Average) terms (q):MA条件是预测方程的滞后预测错误。如：如果q等于5，预测x（t）将是e（t-1）。。。e(t-5),e(i)是移动平均叔在第ith个瞬间和实际值的差值。
* 差分数目Number of Differences (d)：它们是非季节性的差值的数目，即这种情况下我们采用一阶差分。所以传递变量令d=0或者传递原始变量令d=1，两种方法得到的结果一样。

**8.RNN**

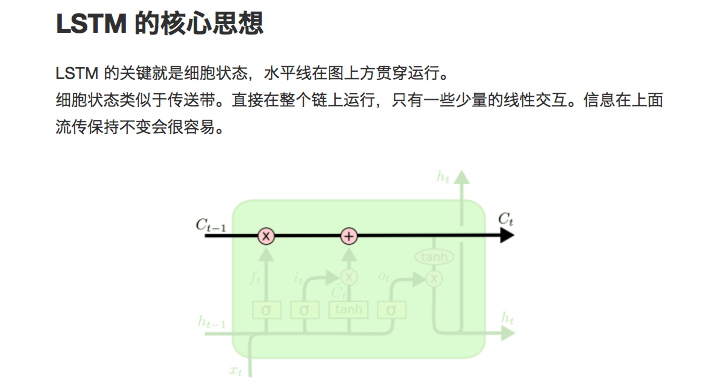
RNNs的目的使用来处理序列数据。在传统的神经网络模型中，是从输入层到隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNNs之所以称为循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNNs能够对任何长度的序列数据进行处理。但是在实践中，为了降低复杂性往往假设当前的状态只与前面的几个状态相关

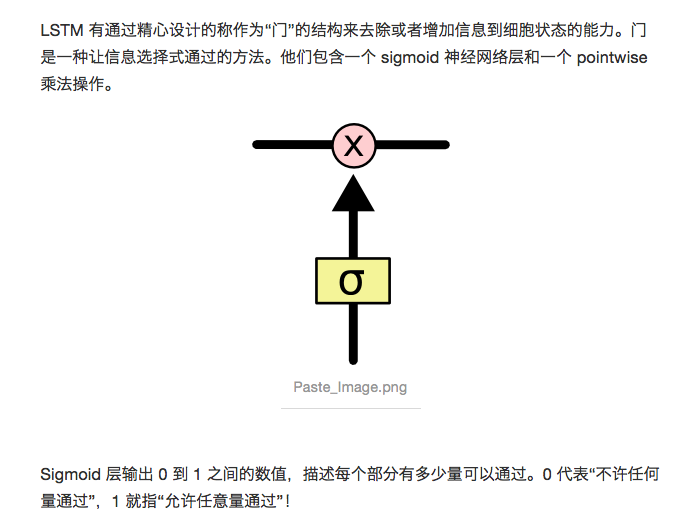
9.LSTM

<http://www.jianshu.com/p/9dc9f41f0b29>

所有 RNN 都具有一种重复神经网络模块的链式的形式。在标准的 RNN 中，这个重复的模块只有一个非常简单的结构，例如一个 tanh 层。

LSTM 同样是这样的结构，但是重复的模块拥有一个不同的结构。不同于 单一神经网络层，这里是有四个，以一种非常特殊的方式进行交互。





**第一步：**在我们 LSTM 中的第一步是决定我们会从细胞状态中丢弃什么信息。这个决定通过一个称为**忘记门层**完成。该门会读取 h\_{t-1} 和 x\_t，输出一个在 0 到 1 之间的数值给每个在细胞状态 C\_{t-1} 中的数字。1 表示“完全保留”，0 表示“完全舍弃”。  
让我们回到语言模型的例子中来基于已经看到的预测下一个词。在这个问题中，细胞状态可能包含当前**主语**的性别，因此正确的**代词**可以被选择出来。当我们看到新的**主语**，我们希望忘记旧的**主语**。

**第二步：**下一步是确定什么样的新信息被存放在细胞状态中。这里包含两个部分。第一，sigmoid 层称 “输入门层” 决定什么值我们将要更新。然后，一个 tanh 层创建一个新的候选值向量，\tilde{C}\_t，会被加入到状态中。下一步，我们会讲这两个信息来产生对状态的更新。  
在我们语言模型的例子中，我们希望增加新的主语的性别到细胞状态中，来替代旧的需要忘记的主语。

**第三步**：现在是更新旧细胞状态的时间了，C\_{t-1} 更新为 C\_t。前面的步骤已经决定了将会做什么，我们现在就是实际去完成。  
我们把旧状态与 f\_t 相乘，丢弃掉我们确定需要丢弃的信息。接着加上 i\_t \* \tilde{C}\_t。这就是新的候选值，根据我们决定更新每个状态的程度进行变化。  
在语言模型的例子中，这就是我们实际根据前面确定的目标，丢弃旧代词的性别信息并添加新的信息的地方。

**第四步：**我们需要确定输出什么值。这个输出将会基于我们的细胞状态，但是也是一个过滤后的版本。首先，我们运行一个 sigmoid 层来确定细胞状态的哪个部分将输出出去。接着，我们把细胞状态通过 tanh 进行处理（得到一个在 -1 到 1 之间的值）并将它和 sigmoid 门的输出相乘，最终我们仅仅会输出我们确定输出的那部分。  
在语言模型的例子中，因为他就看到了一个 **代词**，可能需要输出与一个 **动词** 相关的信息。例如，可能输出是否代词是单数还是负数，这样如果是动词的话，我们也知道动词需要进行的词形变化。