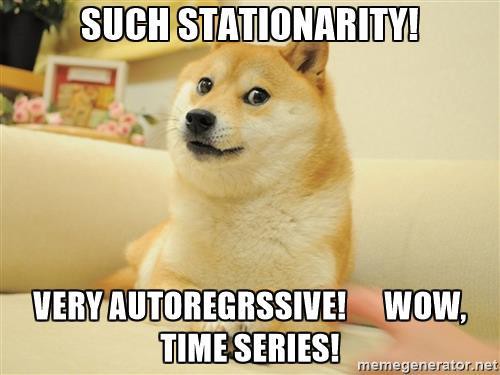
# 开放式机器学习课程。专题9。第一部分。Python中的时间序列分析



你好！

我们继续我们的开放式机器学习课程，以一篇关于时间序列的新文章。

让我们看看如何使用Python中的时间序列，我们可以使用什么方法和模型进行预测；什么是双指数平滑和三指数平滑；如果平稳性不是您喜欢的游戏，该怎么做；如何构建SARIMA并保持活力；如何使用xgboost进行预测。所有这些都将适用于（严酷的）现实世界的例子。

### 文章概要

1. 简介 - 基本定义 - 质量度量
2. 移动，平滑，评估 - 滚动窗口估计 - 指数平滑，霍尔特-温特斯模型 - 时间序列交叉验证，参数选择
3. 计量经济学方法 - 平稳性，单位根 - 摆脱非平稳性 - 萨里玛直觉与模型构建
4. 时间序列上的线性（不完全）模型 - 特征提取 - 线性模型，特征重要性 - 正则化，特征选择 - XGBoost
5. 作业#9

以下内容最好作为

在我的日常工作中，我几乎每天都会遇到时间序列相关的任务。最常见的问题是 - 我们的指标在下一天/周/月将发生什么情况/etc. - 多少玩家将安装应用程序，他们将在网上花费多少时间，用户将执行多少操作，等等。我们可以使用不同的方法来接近预测任务，这取决于所需的预测质量、预测周期的长度，当然，还有时间，我们必须选择特征并调整参数以达到预期的结果。

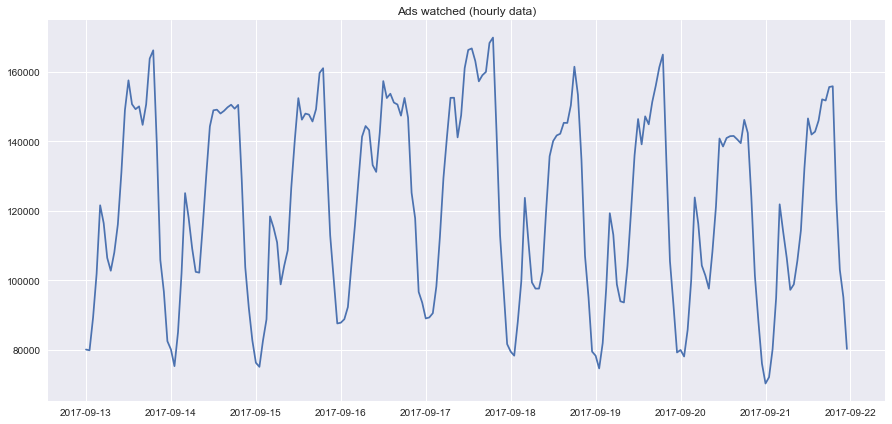
### 介绍

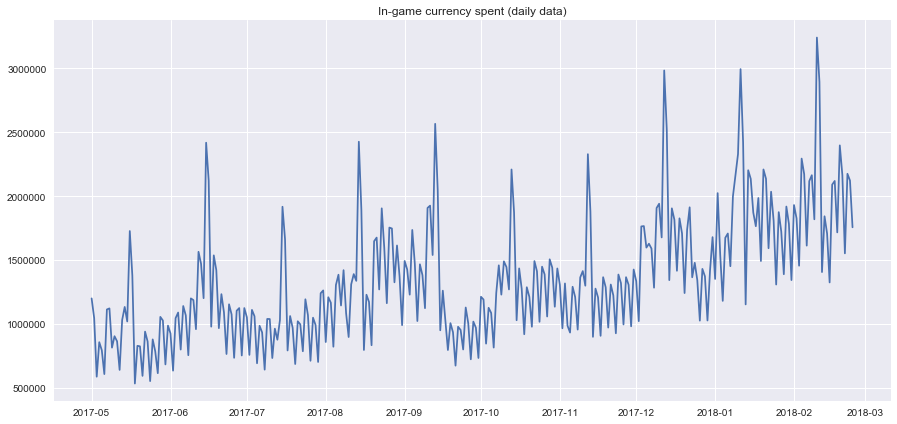
时间序列的小定义：时间序列 - 是按时间顺序索引（或列出或图示）的一系列数据点。

因此，数据是围绕相对确定的时间戳组织的，因此，与随机样本相比，可能包含我们将尝试提取的附加信息。

让我们导入一些库。首先也是最重要的是，我们将需要有大量统计建模功能的库，包括时间序列。对于R afficionados（必须迁移到python）来说，statsmodels看起来肯定很熟悉，因为它支持诸如“工资~年龄+教育”之类的模型定义。

作为一个例子，让我们使用一些真实的移动游戏数据，对玩家观看的每小时广告和每日游戏货币进行分析：





### 预测质量指标

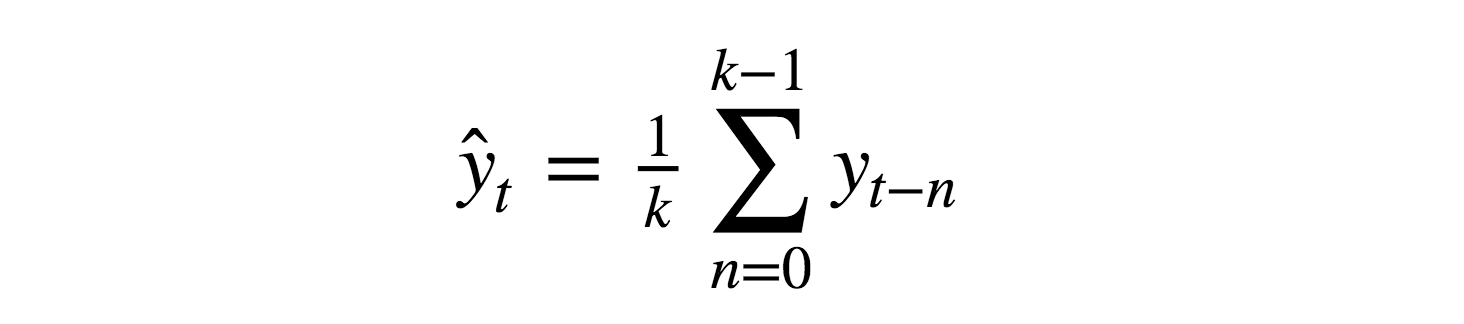
在实际预测之前，让我们先了解如何度量预测的质量，并看看最常用和最广泛使用的度量

* ，决定系数（在计量经济学中，它可以解释为模型解释的方差百分比），（-inf，1]sklearn.metrics.r2\_score
* ，它是一个可解释的度量，因为它与初始序列具有相同的度量单位，[0，+inf）sklearn.metrics.mean\_absolute\_error
* ，同样是一个可解释的度量，特别有趣，因为它对异常值[0，+inf）sklearn.metrics.median\_absolute\_error非常健壮
* ，最常用的是，对大错误给予更高的惩罚，反之亦然，[0，+inf）sklearn.metrics.mean\_squared\_error
* ，实际上与MSE相同，但我们最初取级数的对数，因此我们也注意小错误，通常在数据具有指数趋势时使用，[0，+inf）sklearn.metrics.mean\_squared\_log\_error
* 平均绝对百分比误差，与MAE相同，但百分比， - 当您想向管理层解释模型的质量时非常方便，[0，+inf），不在sklearn中实现

很好，现在我们知道如何衡量预测的质量，我们可以使用什么样的指标，以及如何将结果转化为老板。剩下的就是建立模型。

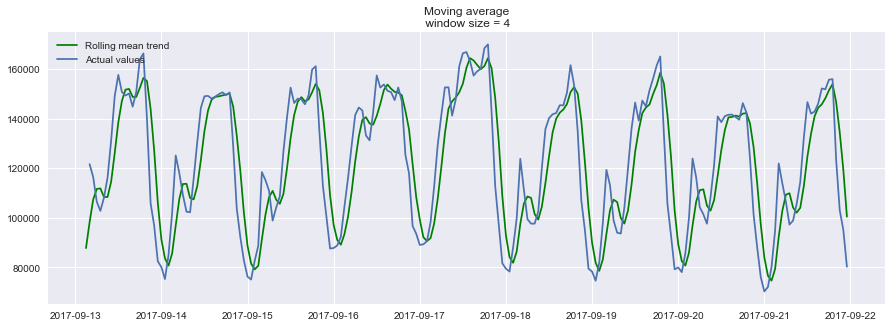
### 移动、平滑、求值

让我们从一个天真的假设开始 - “明天将和今天一样”，但是，与ŷ（t）=y（t-1）这样的模型不同（ŷ（t）=y（t-1）实际上是任何时间序列预测问题的一个很好的基线，有时用任何模型都不可能打败它），我们假设变量的未来值取决于其先前值的平均n，因此我们将使用移动平均值。

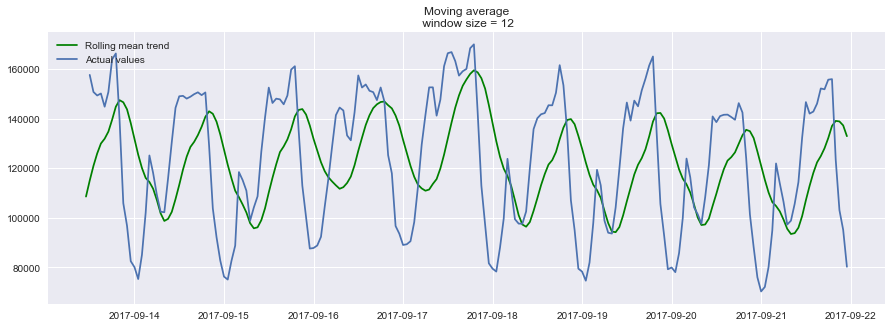


不幸的是，我们不能做长期的预测，为了得到下一步的预测值，我们需要实际观察上一个值。但是移动平均有另一个用例 - 平滑原始时间序列以指示趋势。Pandas有一个可用的实现DataFrame.rolling（window.mean（））。窗口越宽，趋势就越平滑。对于金融业中经常遇到的非常嘈杂的数据，此过程有助于检测常见的模式。

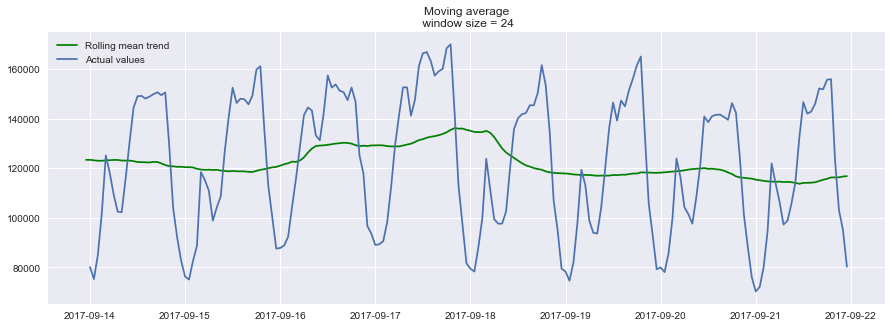
最后4小时平滑绘图移动平均值（ads，4）



按最近12小时的绘图平滑移动平均值（ads，12）



平滑24小时 - 我们得到每日趋势图移动平均值（ads，24）

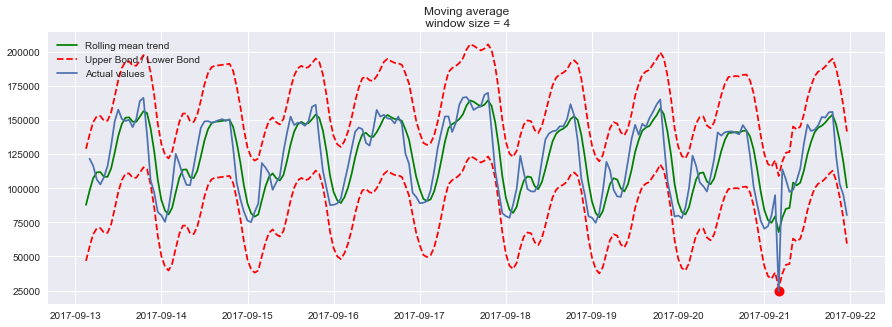


如您所见，对小时数据应用每日平滑处理可以让我们清楚地看到所观看广告的动态。在周末，值较高（周末 - 玩的时间），而工作日通常较低。

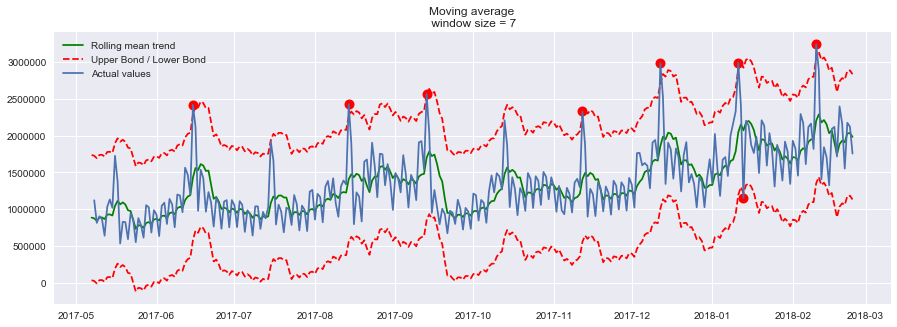
我们还可以为平滑值绘制置信区间图movingaverage（ads，4，plot\_intervals=True）

现在让我们用移动平均值创建一个简单的异常检测系统。不幸的是，在这个特定的系列中，一切或多或少都是正常的，因此我们将有意使数据帧ads中的一个值异常

让我们看看，如果这个简单的方法能够捕捉到异常图移动平均值（ads\_anomality，4，plot\_interval=True，plot\_anomalies=True）

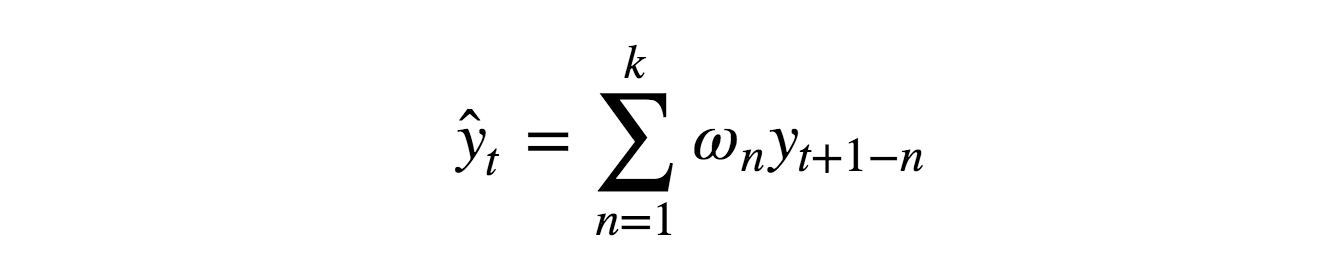


整洁！第二个系列（每周平滑）怎么样？绘图移动平均值（货币，7，绘图间隔=真，绘图异常=真）



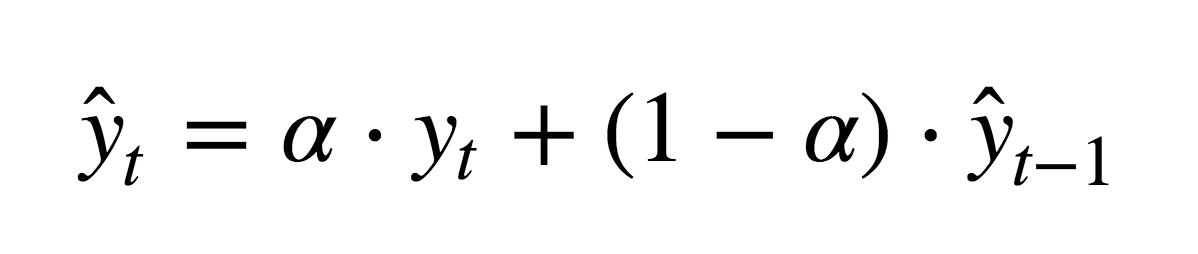
哦，不！这是我们简单方法的缺点，它没有捕捉到我们数据中的月度季节性，并将几乎所有30天的峰值标记为异常。如果你不想有那么多假警报，最好考虑更复杂的模型。

加权平均是对移动平均的一种简单的修正，在移动平均中，观测值有不同的权值加在一起，通常最近的观测值有更大的权值。



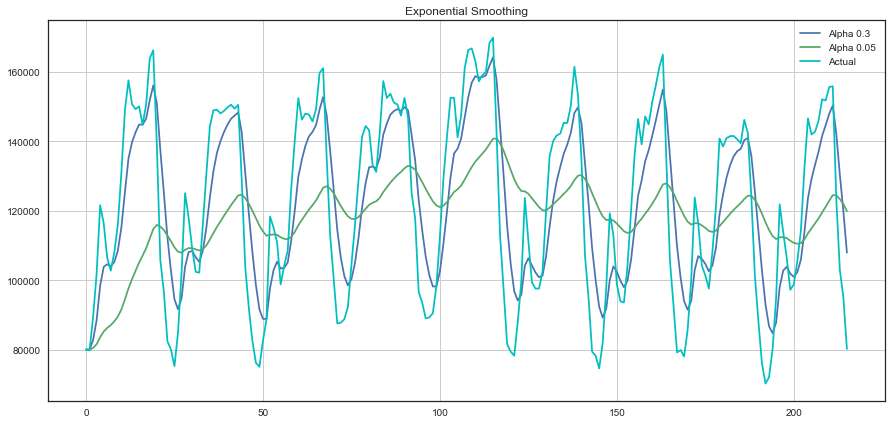
### 指数平滑

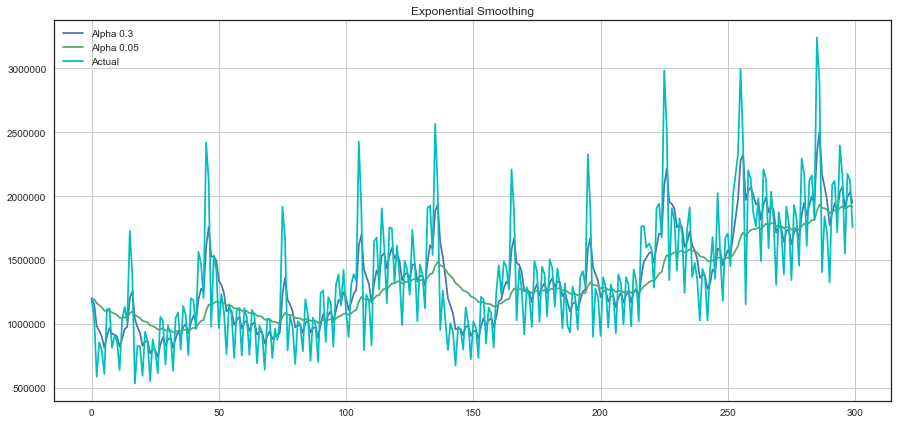
现在让我们来看看，如果我们开始对所有可用的观测值进行加权，而不是对时间序列的最后一个nn值进行加权，同时随着历史数据的进一步回溯，权重呈指数递减，会发生什么。有一个简单的公式可以帮助我们：



这里，模型值是当前真值和以前模型值之间的加权平均值。α权重称为平滑因子。它定义了我们“忘记”最后一个可用的真实观察的速度。α越小，先前模型值的影响越大，序列越平滑。

指数性隐藏在函数 - 的递归性中，我们每次将（1--α）乘以先前的模型值，该模型值反过来也包含（1--α）等等，直到开始。

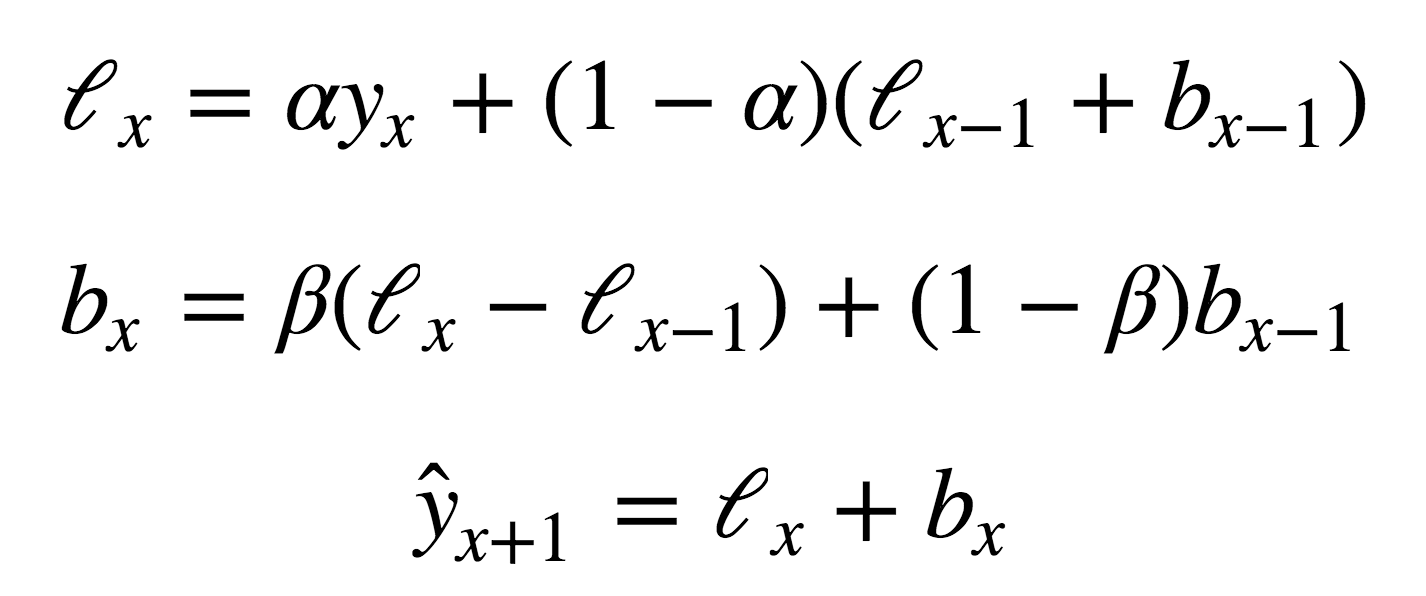




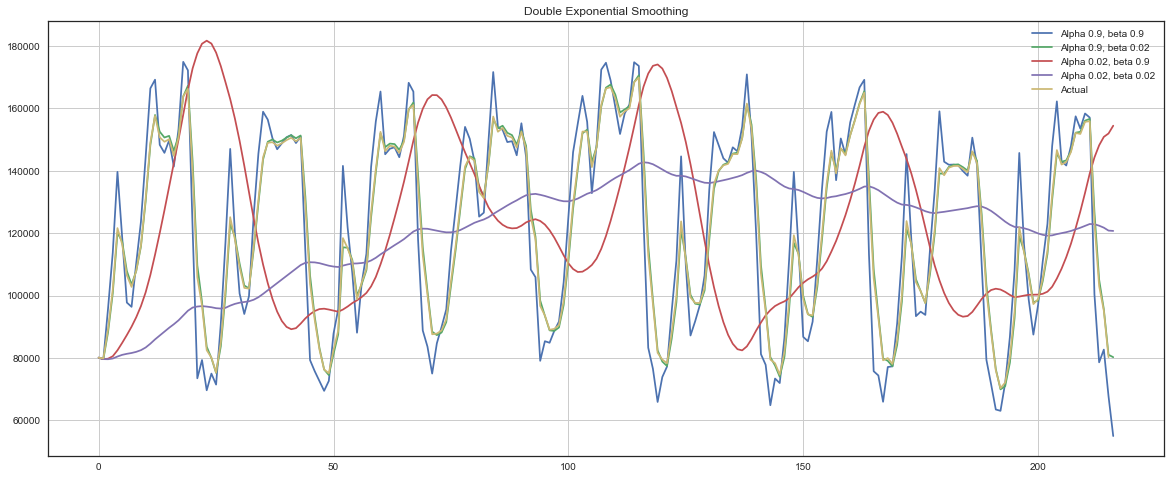
### 双指数平滑

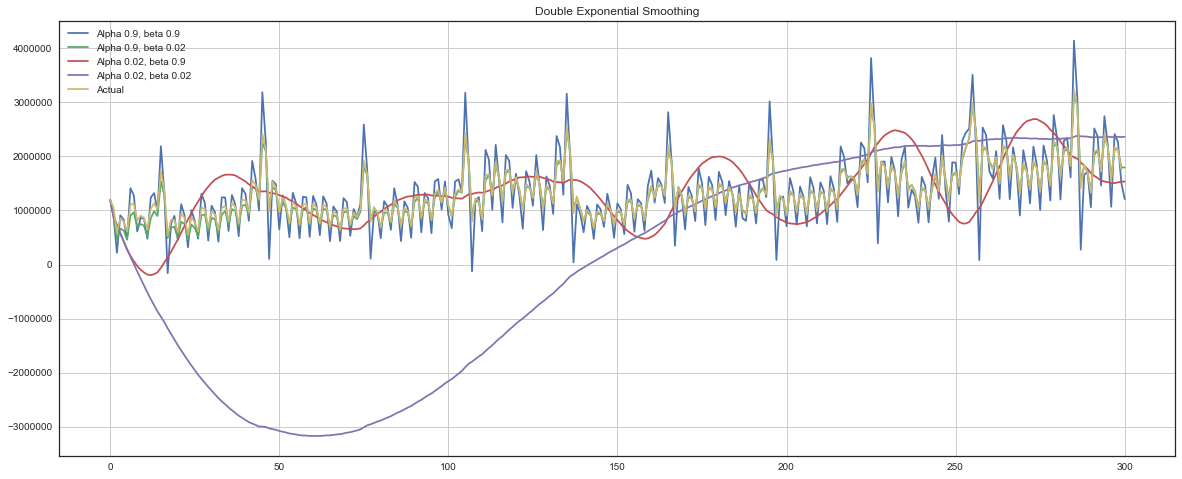
到目前为止，在最好的情况下，我们只能从我们的方法中得到一个单一的未来点预测（以及一些很好的平滑），这很酷，但还不够，所以让我们扩展指数平滑，以便我们可以预测两个未来点（当然，我们也得到一些平滑）。

级数分解应该有助于我们获得两个分量：截距（也就是水平）和趋势（也就是斜率）b。我们已经学会了用以前的方法预测截距（或者预期的级数值），现在我们将对趋势应用同样的指数平滑，天真地相信或者不相信时间的未来方向序列变化取决于以前的加权变化。



结果我们得到了一组函数。第一个术语描述截距，和以前一样，它取决于序列的当前值，而第二个术语现在被分解为水平和趋势的先前值。第二个函数描述趋势 - 它取决于当前步骤的级别变化和趋势的前一个值。在这种情况下，β系数是指数平滑中的一个权重。最后的预测是截距和趋势的模型值之和。





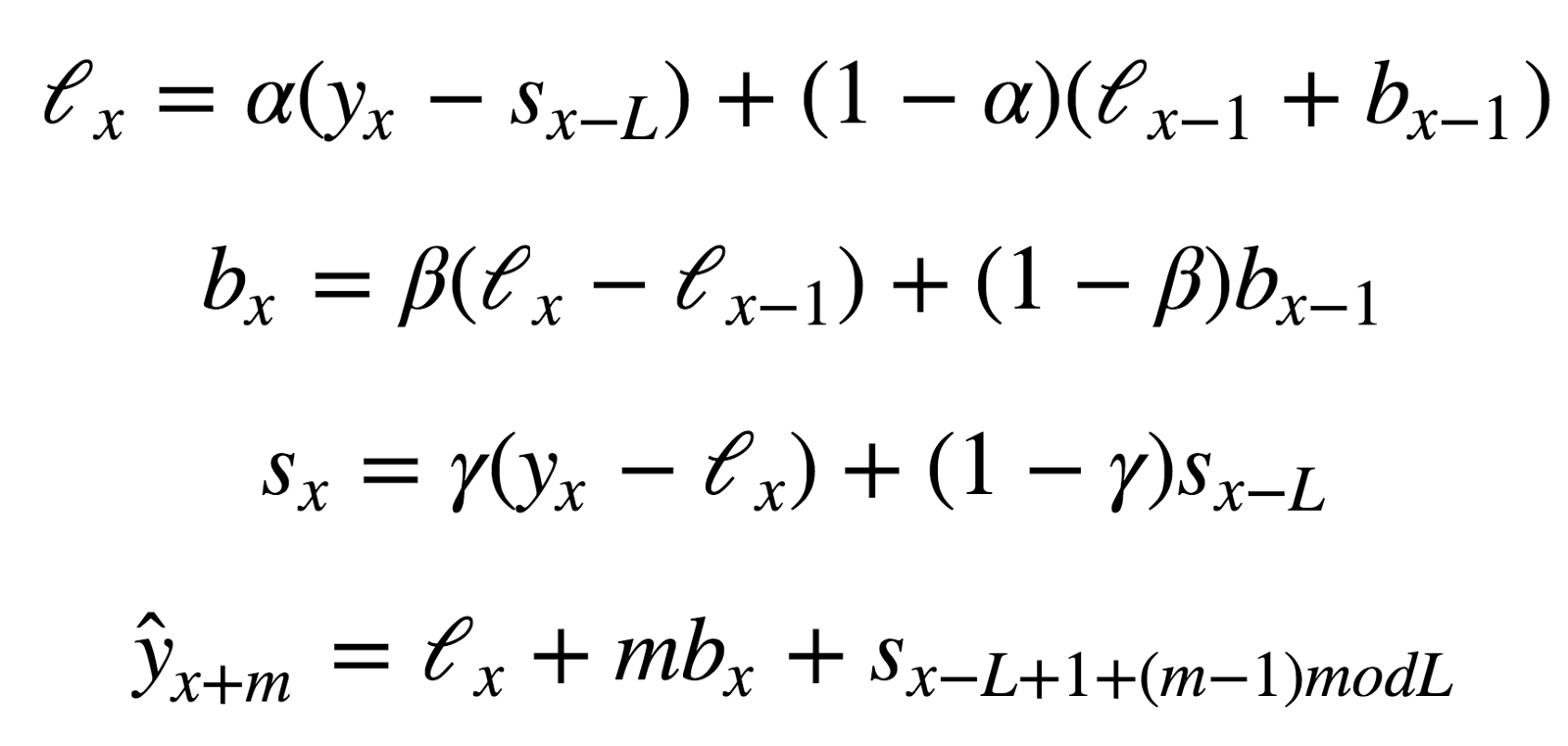
现在我们必须调整两个参数 - α和β。前者负责趋势周围的序列平滑，后者负责趋势本身的平滑。值越大，最新观测值的权重就越大，模型序列的平滑度就越低。参数的组合可能会产生非常奇怪的结果，特别是如果手动设置的话。在三次指数平滑之后，我们会马上研究自动选择参数。

### 三指数平滑a.k.a.霍尔特温特斯

万岁！我们已经成功地达到了指数平滑的下一个变种，这次是三倍。

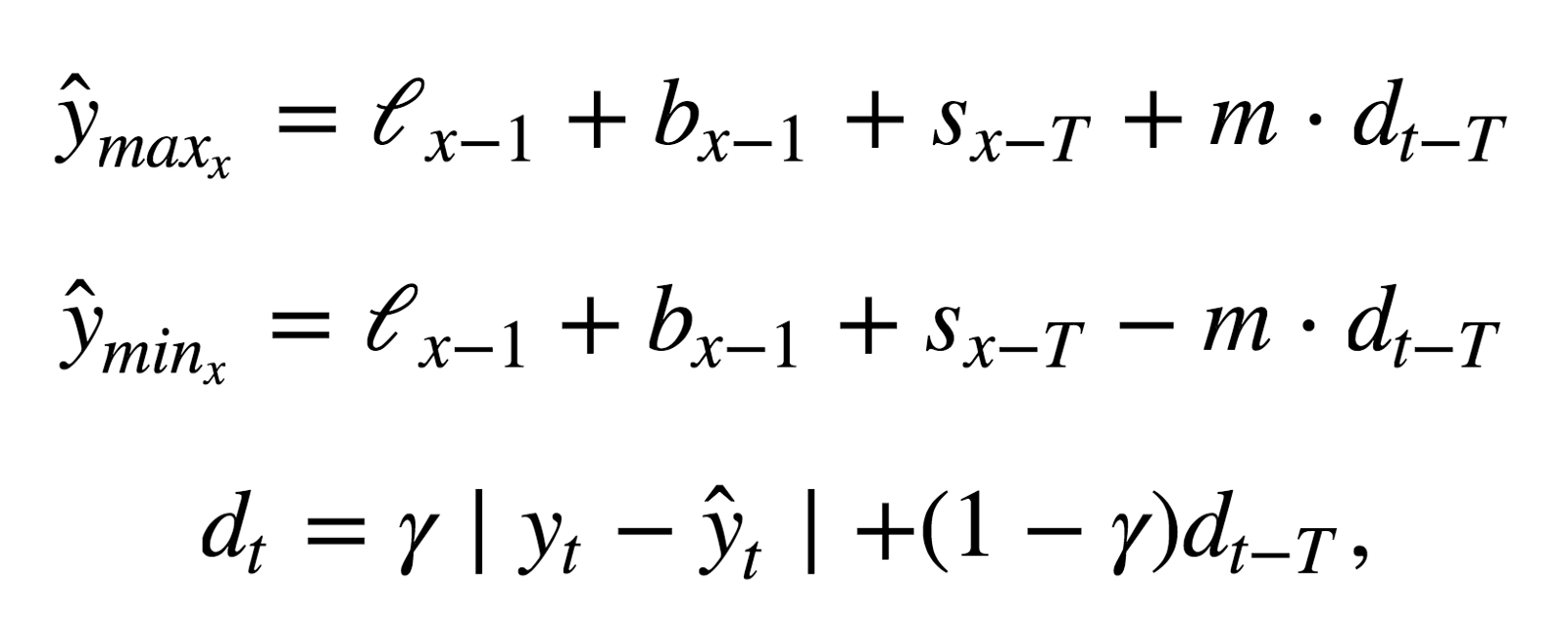
这个方法的思想是我们加上另一个，第三个分量 - 季节性。这意味着，如果我们的时间序列没有季节性，我们就不应该使用这种方法，这在我们的例子中不是这样的。模型中的季节分量将解释截距和趋势附近的重复变化，它将用季节长度来描述，也就是说，用变化重复的周期来描述。对于季节中的每个观测，都有一个单独的组成部分，例如，如果季节长度是7（每周季节性），我们将有7个季节性组成部分，一周中的每一天一个。

现在我们有了一个新系统：



截距现在取决于序列的当前值减去相应的季节性分量，趋势保持不变，季节性分量取决于序列的当前值减去截距和分量的先前值。请考虑到组件在所有可用季节都是平滑的，例如，如果我们有星期一组件，则它将只与其他星期一平均。您可以阅读更多关于平均如何工作，以及如何初步近似的趋势和季节性的组成部分做。现在我们有了季节性因素，我们可以预测不是一个，甚至不是两个，而是任意的毫米未来步骤，这是非常令人鼓舞的。

下面是一个三指数平滑模型的代码，也被它的创建者查尔斯·霍尔特和他的学生彼得·温特斯的姓氏所熟知。另外，模型中还包含了Brutlag方法，以建立置信区间：



其中T是季节长度，d是预测偏差，其他参数取自三次指数平滑。您可以阅读更多关于该方法及其在时间序列异常检测中的适用性的信息。

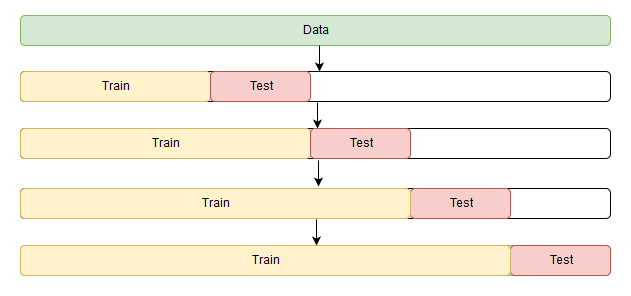
### 时间序列交叉验证

在我们开始建立模型之前，我们先来谈谈如何自动估计模型参数。

这里没有什么不寻常的，因为我们总是要选择一个适合于任务的损失函数，这将告诉我们模型接近数据的程度。然后，通过交叉验证，我们将对给定的模型参数评估我们选择的损失函数，计算梯度，调整模型参数等，勇敢地降到全局误差最小。

问题是如何对时间序列进行交叉验证，因为，你知道，时间序列确实有时间结构，如果不保留这个结构，就不能在一个折叠中随机混合值，否则观测值之间的所有时间依赖都将丢失。这就是为什么我们将不得不使用一个更棘手的方法来优化模型参数，我不知道它是否有一个正式的名字，但在那里，人们可以找到所有的答案，但最终的问题，生命，宇宙，和一切的答案，“滚动交叉验证”是作为一个名字提出的。

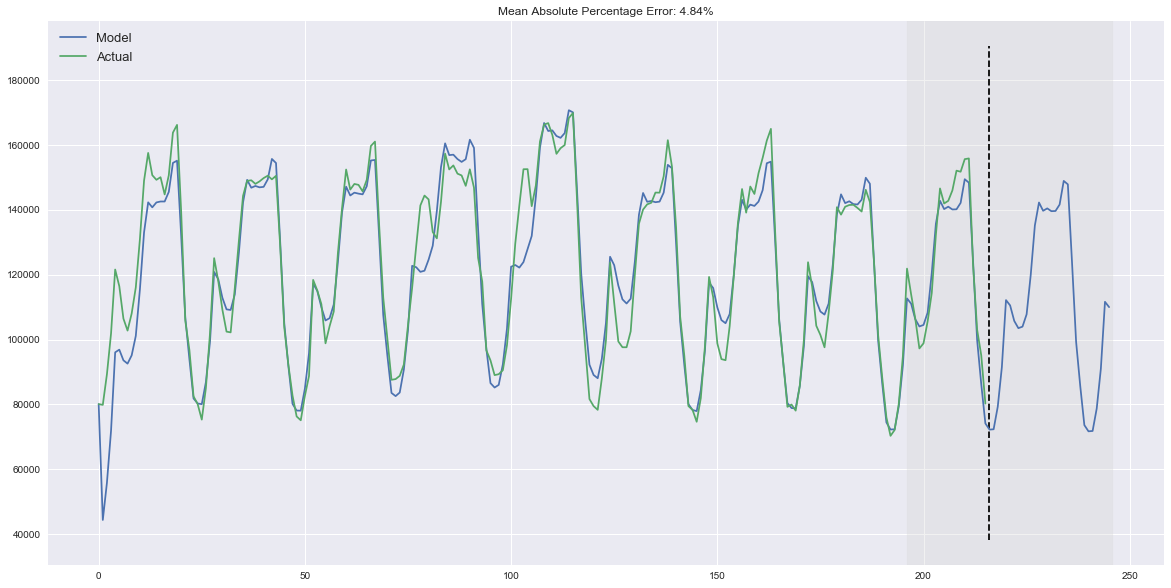
这个想法相当简单，我们在时间序列的一小段上训练我们的模型，从开始到某个t，对下一个t+nsteps进行预测并计算误差。然后我们扩展训练样本直到t+n值，并从t+n到t+2\*n进行预测，然后继续移动时间序列的测试段，直到到达最后一个可用的观测值。因此，在初始训练样本和最后一次观察之间，我们有尽可能多的n倍。

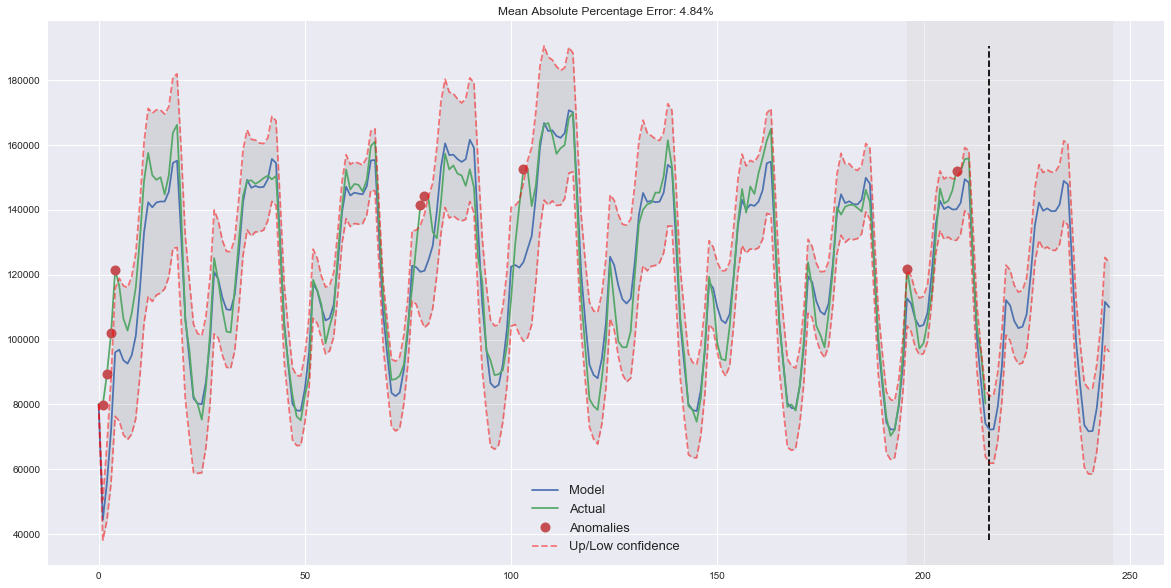


现在，知道如何设置交叉验证，我们将为Holt-Winters模型找到最佳参数，回想一下，我们在ads中有每日季节性，因此slen=24参数

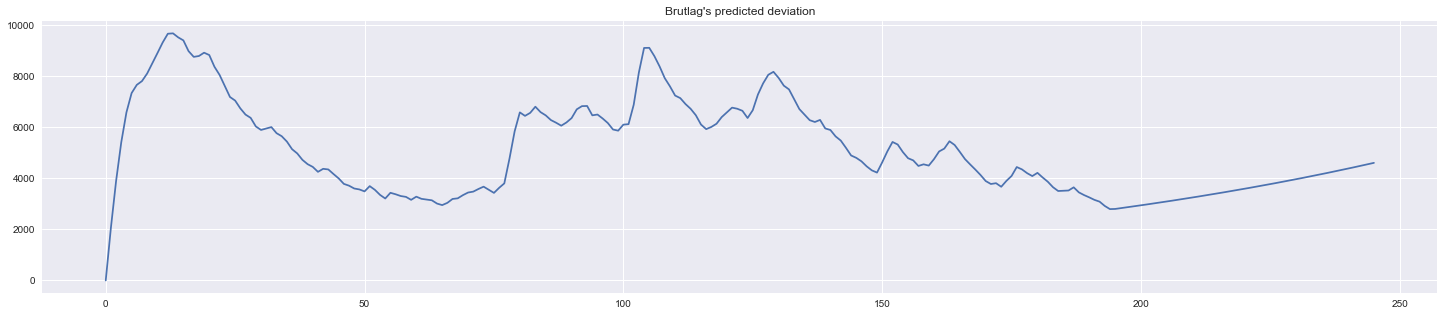
在Holt-Winters模型和其他指数平滑模型中，平滑参数的大小是有限制的，每个参数都在0到1之间，因此为了最小化损失函数，我们必须选择一个支持模型参数约束的算法，在我们的例子中 - 截断牛顿共轭梯度。

图表呈现代码





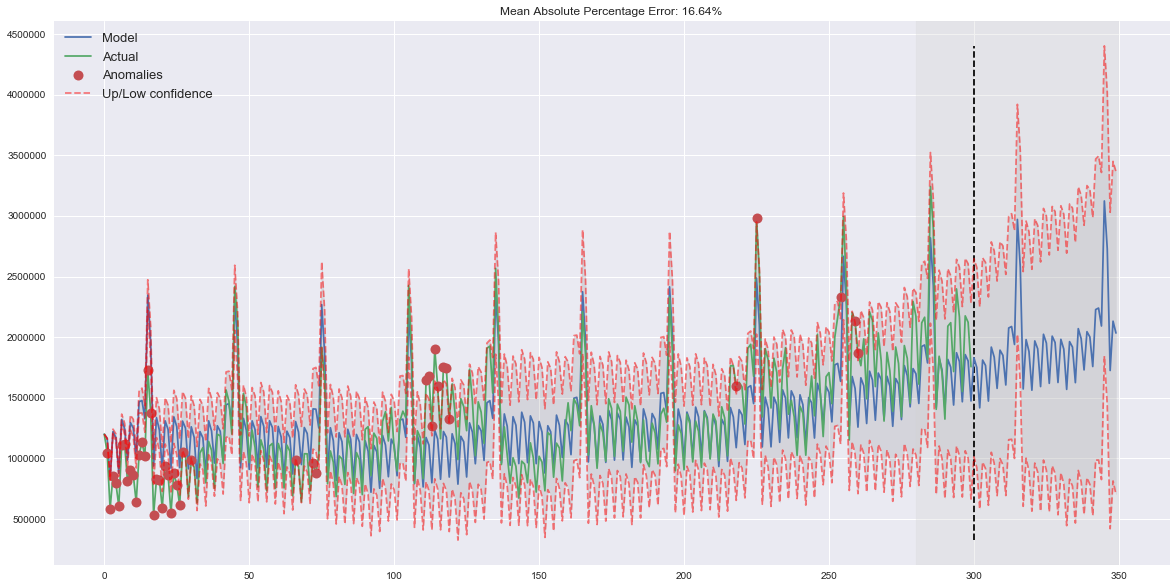
从图表来看，我们的模型能够成功地逼近初始时间序列，捕捉每天的季节性，整体下降趋势，甚至一些异常。如果你看一看模型的偏差，你可以清楚地看到模型对序列结构的变化反应非常剧烈，但是很快就会将偏差恢复到正常值，“忘记”过去。该模型的这一特点使我们能够快速构建异常检测系统，即使是在噪声较大的序列中，也不会花费太多的时间和金钱来准备数据和训练模型。

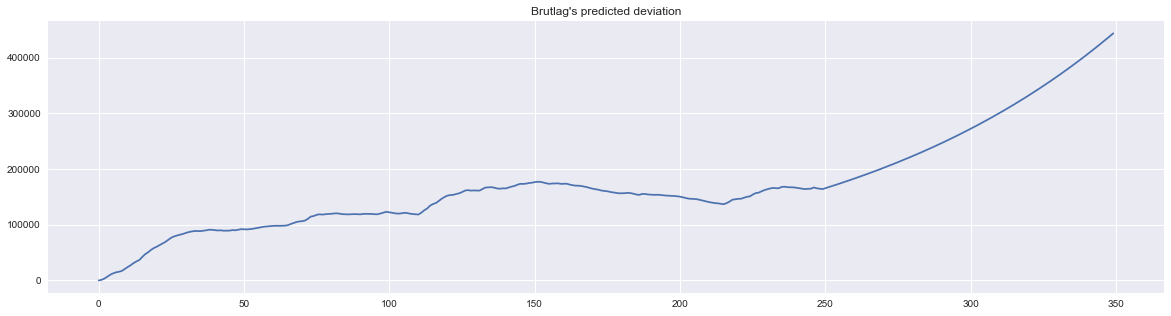


我们将对第二个序列应用相同的算法，如我们所知，它具有趋势性和30天的季节性



看起来很合适，模型捕捉到了上升趋势和季节性峰值，总体上很好地符合我们的价值观





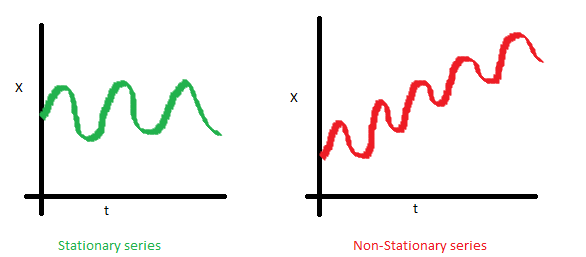
### 计量经济学方法

### 平稳性

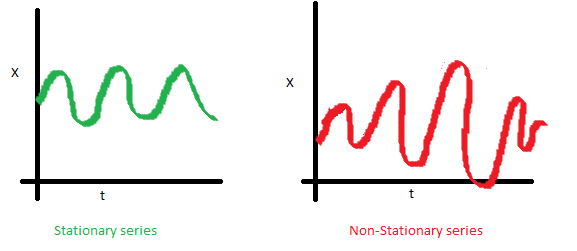
在我们开始建模之前，我们应该提到时间序列的一个重要性质。

如果过程是平稳的，这意味着它不会随着时间而改变其统计特性，即均值和方差不会随着时间而改变（方差的恒定性也被称为），那么协方差函数也不依赖于时间（应该只依赖于观察之间的距离）。您可以在以下文章的图片上看到这一点：

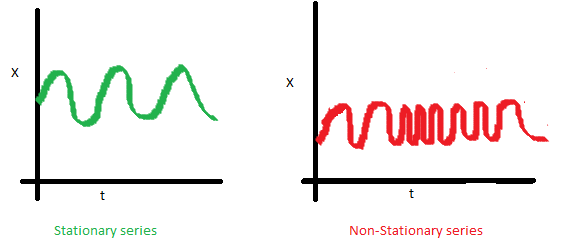
* 下面的红色图表不是平稳的，因为平均值随着时间的推移而增加。



* 我们很不走运，看到值随时间的变化



* 最后，第i项和第（i+m）项的协方差不应是时间的函数。在下图中，您将注意到随着时间的增加，排列变得更近。因此，对于正确的图表，协方差不是随时间变化的常数。



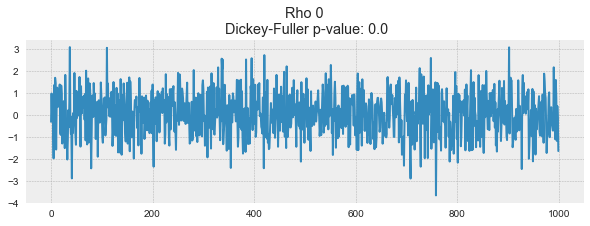
那么，为什么平稳性如此重要呢？因为当我们假设未来的统计特性与当前观测结果没有差异时，很容易对平稳序列进行预测。大多数时间序列模型都是以一种或另一种方式来预测这些性质（例如，均值或方差），这就是为什么如果原始序列不是平稳的，那么预测就错了。不幸的是，我们在课本之外看到的大多数时间序列都是非平稳的，但我们可以（也应该）改变这种情况。

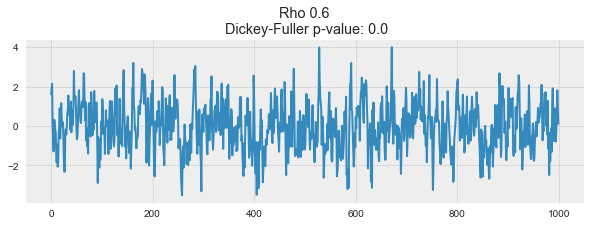
所以，要与非平稳状态作斗争，我们必须了解我们的敌人。让我们看看如何检测它。为了做到这一点，我们现在将看看白噪声和随机行走，我们将学习如何从一个到另一个免费，没有注册和短信。

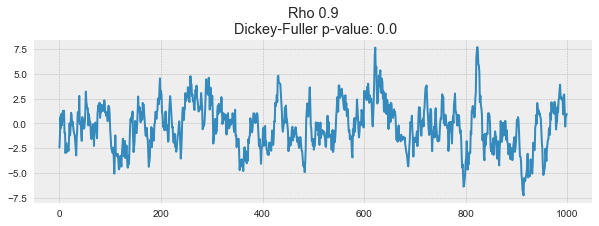
白噪声图：

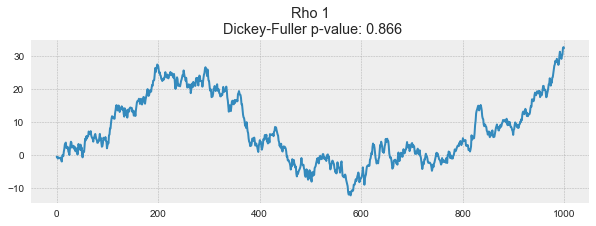
因此由标准正态分布产生的过程是平稳的，在0附近振荡，偏差为1。现在基于这个过程，我们将生成一个新的值，其中每个下一个值都将依赖于前一个值：x（t）=ρ\*x（t-1）+e（t）

图表呈现代码









在第一张图表上，你可以看到与以前相同的静止白噪声。在第二种情况下，ρρ的值增加到0.6，结果图表上出现了更宽的循环，但总体上它仍然是静止的。第三个图表与0平均值的偏差更大，但仍在其周围振荡。最后，ρ等于1的值给出了一个随机游走过程 - 非平稳时间序列。

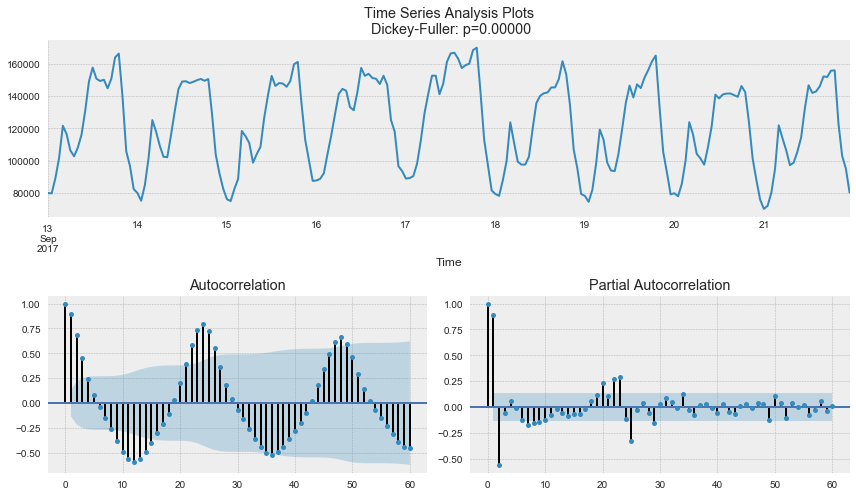
这是因为在达到临界值后，级数x（t）=ρx（t-1）+e（t）不返回其平均值。如果我们从左边和右边减去x（t-1），我们将得到x（t）-x（t-1）=（ρ-1）x（t-1）+e（t），左边的表达式称为第一个差分。如果ρ=1，则第一个差分给出静止白噪声e（t）。这个事实是时间序列平稳性（存在单位根）的主要思想。如果我们能用第一个差分从非平稳数列得到平稳数列，我们称这些数列为1阶积分。检验 - 时间序列的零假设是非平稳的，在前三个图表上被拒绝，在最后一个图表上被接受。我们不得不说，第一个差分并不总是足以得到平稳序列，因为过程可能是d，d>1阶的积分（并且有多个单位根），在这种情况下，使用增广Dickey-Fuller检验一次检查多个滞后。

我们可以使用不同的方法来对抗非平稳性，如各种阶差、趋势和季节性去除、平滑，也可以使用Box-Cox或对数等变换。

### 摆脱非平稳状态建设萨里玛

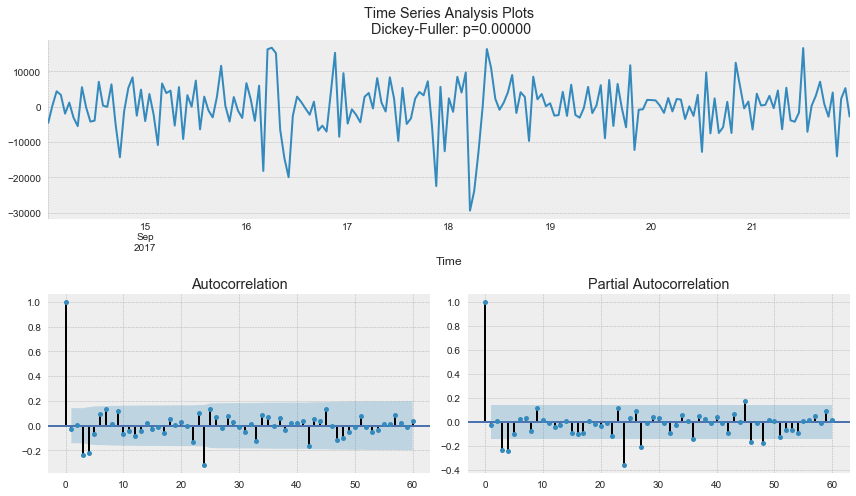
现在让我们建立一个ARIMA模型，通过遍历所有的地狱阶段，使系列静止。

图表呈现代码



令人惊讶的是，初始序列是平稳的，Dickey-Fuller检验否定了单位根存在的空假设。实际上，它可以在曲线图上看到，我们没有明显的趋势，所以平均值是恒定的，整个序列的方差是非常稳定的。唯一剩下的是季节性，我们必须在建模前处理。要做到这一点，让我们以“季节性差异”为例，这意味着简单地从序列本身减去一个等于季节性周期的滞后。

更好的是，可见的季节性消失了，但是自相关函数仍然有太多的显著滞后。为了消除它们，我们将用滞后1从序列中减去第一个差



很 完美！我们的系列现在看起来像是一些难以描述的东西，在零附近振荡，Dickey Fuller表示它是静止的，ACF中的显著峰的数量已经下降。我们终于可以开始做模特了！

### 阿里马家庭速成班

关于模特的几句话。我们将逐字构建全称 - SARIMA（p，d，q）（p，d，q，s），季节自回归移动平均模型：

* AR（p） 自回归模型，即时间序列对自身的回归。基本假设 - 当前序列值取决于其具有一些滞后（或几个滞后）的先前值。模型中的最大滞后被称为P。为了确定初始P，你需要查看PACF图-----找到最大的显著滞后，之后大多数滞后都变得不显著。
* 移动平均模型。在不详细描述的情况下，它对时间序列的误差进行了建模，同样的假设是 - 当前误差依赖于具有一定滞后的前一个，即q。初始值可以在具有相同逻辑的ACF图上找到。

让我们稍微休息一下，把前4个字母组合起来：

AR（p）+MA（q）=ARMA（p，q）

我们这里有一个自回归移动平均模型！如果序列是静止的，则可以用这4个字母来近似。我们继续吗？

* I（d）-集成顺序。它只是使序列平稳所需的非季节性差异的数量。在我们的例子中只有1，因为我们使用了第一个差异。

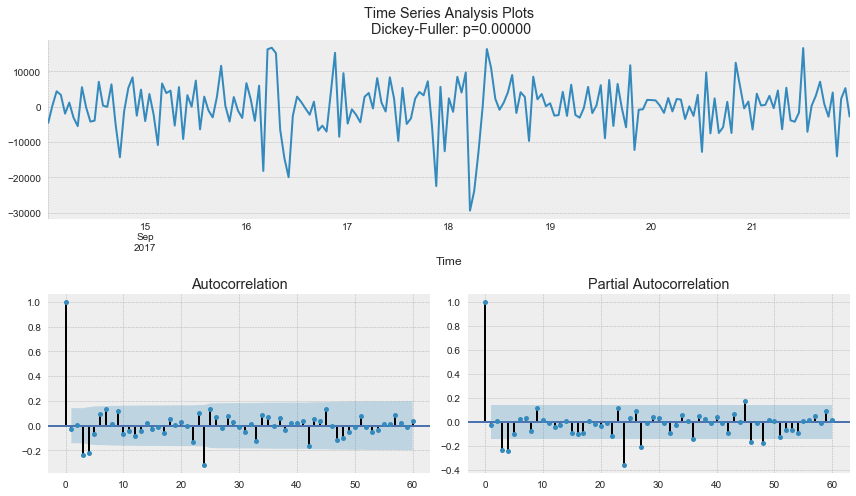
在前面四个字母的基础上加上这封信，我们得到了ARIMA模型，它知道如何利用非季节性差异处理非平稳数据。太棒了，还剩最后一封信了！

* S（S） - 本函对季节性负责，等于系列的季节周期长度

在附上最后一个字母后，我们发现我们得到的不是一个附加参数，而是一行三个 - （P，D，Q）

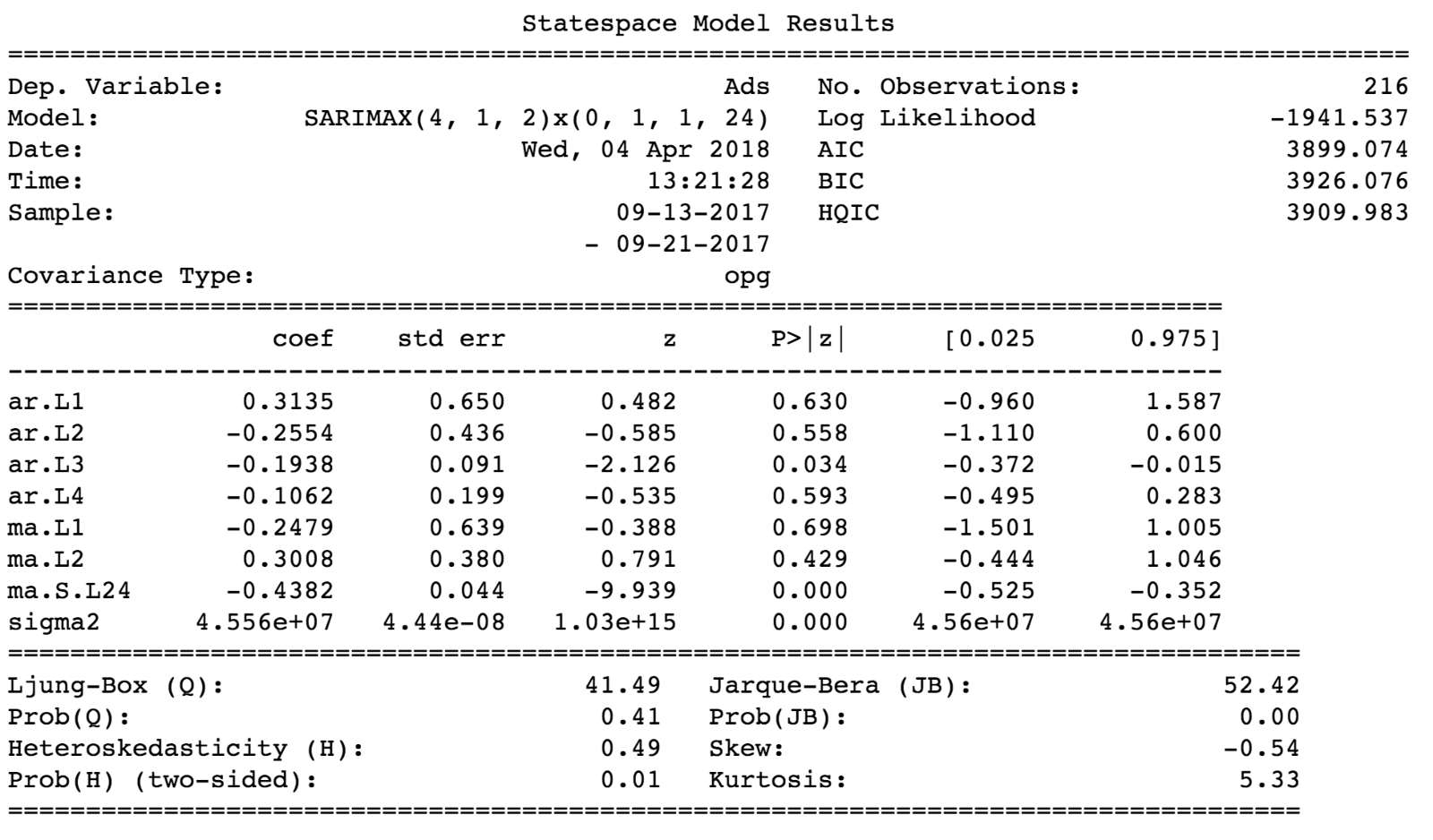
* 模型季节分量的P - 自回归阶，同样可以从PACF中得到，但是这次你需要看看有效滞后的数量，这是季节周期长度的倍数，例如，如果周期等于24，我们看到PACF的24和48滞后是显著的，这意味着初始P应该是2。
* Q- 逻辑相同，但对于季节性成分的移动平均模型，使用ACF图
* 季节整合的D - 顺序。可以等于1或0，具体取决于是否应用了季节性差异

现在，了解如何设置初始参数，让我们再次查看最终绘图并设置参数：tsplot（ads\_diff[24+1:]，lags=60）

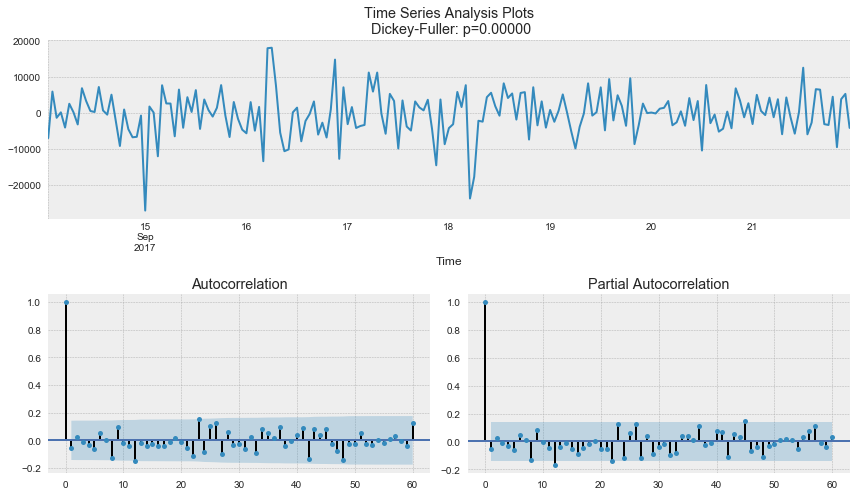


* p最有可能是4，因为这是PACF的最后一个显著滞后，之后其他大多数都变得不显著。
* d等于1，因为我们有第一个差异
* q应该在4左右，在ACF上也可以看到
* P可能是2，因为第24和48个滞后对PACF有一定的影响
* D再次等于1 - 我们进行了季节分化
* Q可能是1，ACF的24小时滞后是显著的，而48小时不是

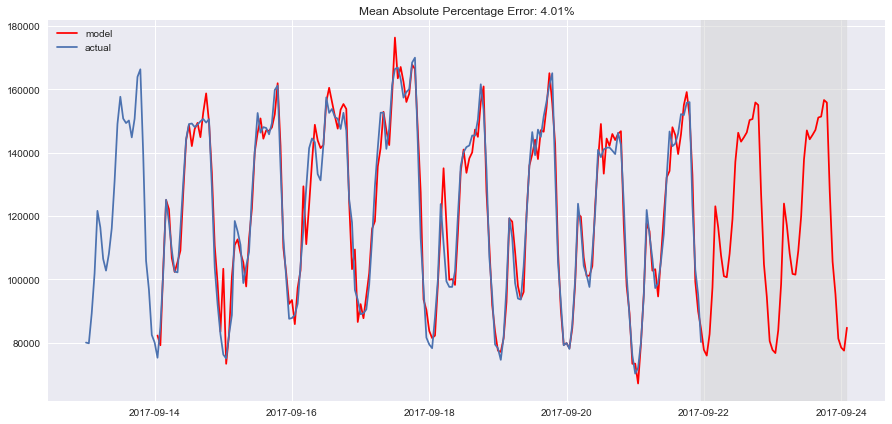
现在我们要测试不同的模型，看看哪一个更好



让我们检查一下tsplot模型的残差（best\_model.resid[24+1:]，lags=60）



很明显，残差是平稳的，没有明显的自相关，让我们用我们的模型来预测



最后我们得到了相当充分的预测，我们的模型平均错误率为4.01%，这是非常好的，但是准备数据、使序列平稳和蛮力参数选择的总体成本可能不值得这样的准确性。

### 时间序列上的线性（不完全）模型

又是一段抒情的离题。在我的工作中，我经常要用唯一的指导原则来建立模型。这意味着一些模型永远不会“生产就绪”，因为它们需要太多的时间来准备数据（例如，SARIMA），或者需要频繁地对新数据进行重新培训（同样，SARIMA），或者很难调整（很好的例子 - SARIMA），因此，从现有的时间序列中选择几个特征并建立一个简单的线性回归，或者说是一个随机森林，通常是非常容易的。又好又便宜。

也许这种方法没有理论的支持，打破了不同的假设（如Gauss-Markov定理，特别是关于误差不相关的假设），但是它在实践中非常有用，并且在机器学习竞赛中经常使用。

### 特写

好吧，模型需要特性，我们只有一维时间序列。我们可以排除哪些功能？

当然是时间序列的滞后

窗口统计：

* 窗口中序列的最大/最小值
* 窗口中的平均值/中值
* 窗口方差
* 等。

日期和时间特征：

* 一小时，一天中的一小时，一周中的一天，你明白了
* 今天是假日吗？也许发生了什么特别的事？使其成为布尔特征

目标编码

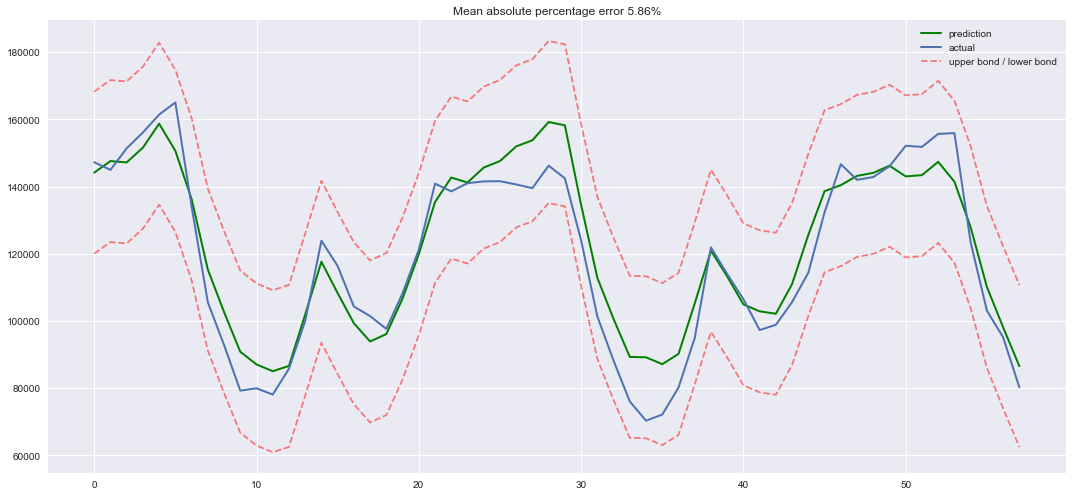
来自其他模型的预测（尽管这样我们可能会失去预测的速度）

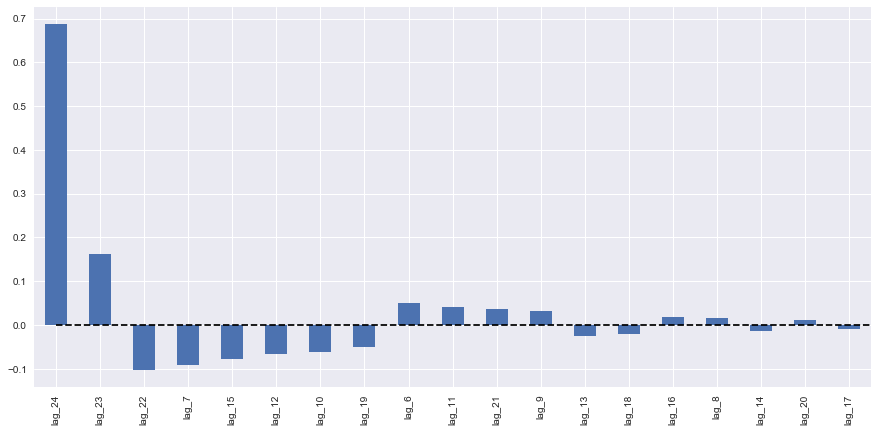
让我们浏览一下这些方法，看看我们能从广告系列中提取什么

### 时间序列滞后

将序列n向后移动，我们得到一个特征列，其中时间序列的当前值与其在时间t-n时的值对齐。如果我们进行1个滞后移动，并对该特征训练一个模型，该模型将能够在观察到序列的当前状态后预测1个提前步骤。比如说，将滞后时间增加到6，将允许模型提前6步进行预测，但是它将使用6步后观察到的数据。如果在未观察到的时间段内，某个因素从根本上改变了序列，那么模型将无法捕捉到这些变化，并将返回带有大误差的预测。因此，在初始滞后选择过程中，必须在最优预测质量和预测时间长度之间找到一个平衡点。

太好了，我们有一个数据集，为什么不训练一个模型呢？

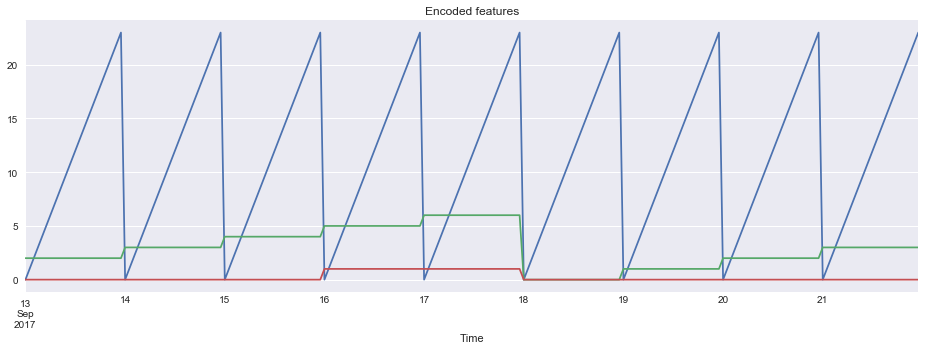




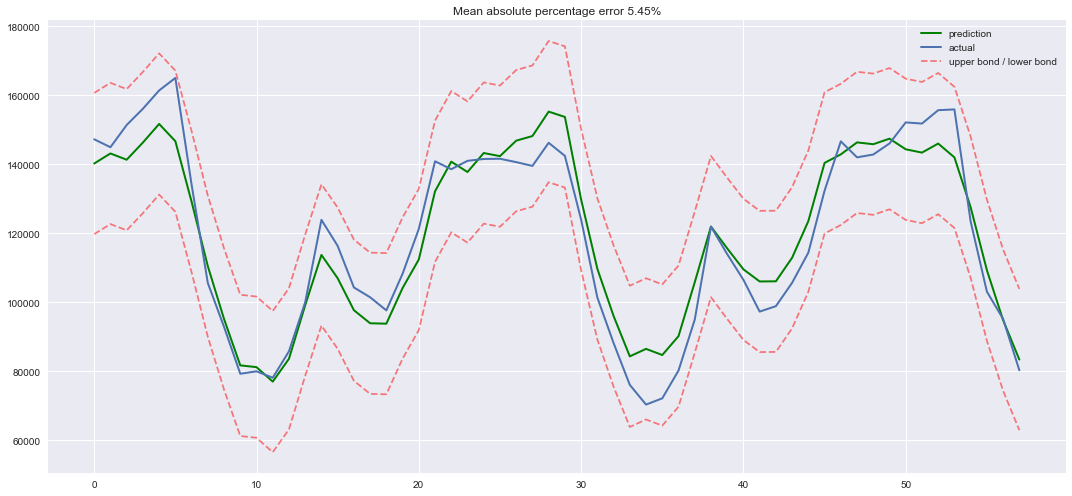
好吧，简单的滞后和线性回归给了我们在质量上离萨里玛不远的预测。有很多不必要的特性，但我们稍后将进行特性选择。现在让我们继续工程！

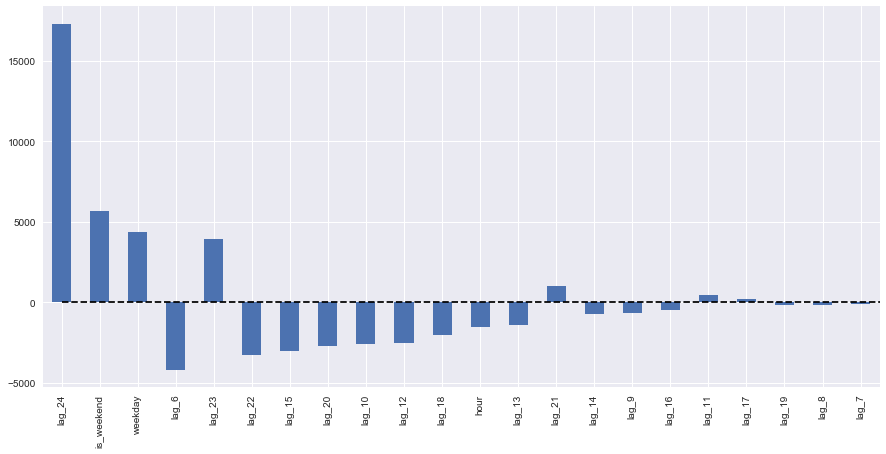
我们将在数据集中添加周末的小时、星期几和布尔值。为此，我们需要将当前数据帧索引转换为日期时间格式，并将小时和工作日从中移出。

我们可以将结果特征可视化



既然现在我们有不同的变量尺度，对于滞后特征有数千个，对于分类特征有数十个，那么将它们转换成相同的尺度来继续探索特征重要性和随后的正则化是合理的。



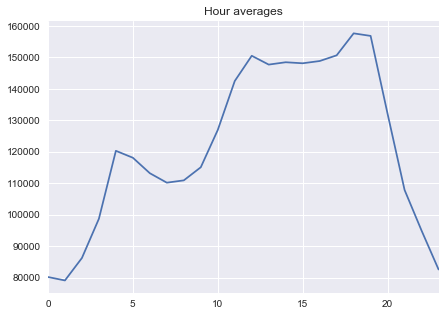


测试误差下降了一点，根据系数图判断，我们可以说工作日和周末是相当有用的特征

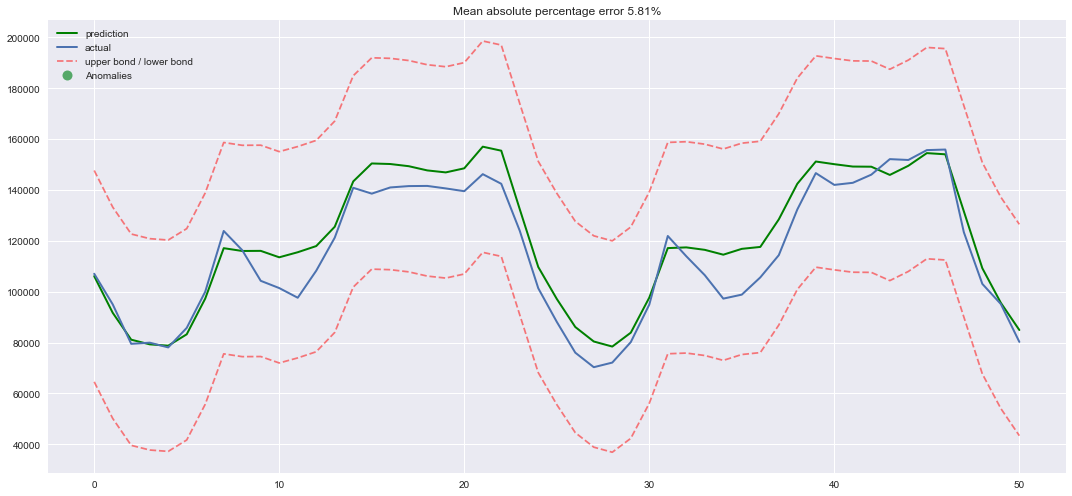
### 目标编码

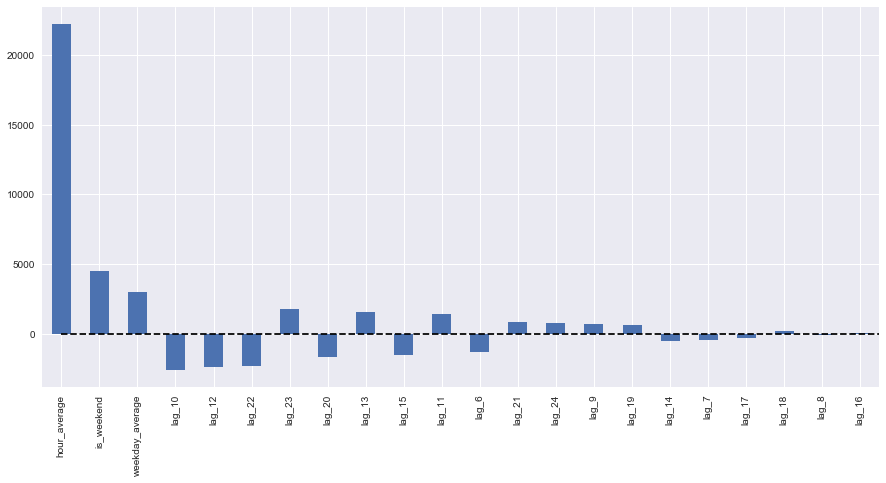
我想添加另一种分类变量的平均值编码。如果不希望使用大量虚拟变量来爆炸数据集，而这些变量可能导致有关距离的信息丢失，并且如果它们由于“0小时<23小时”之类的冲突而不能用作实际值，则可以使用稍微多一些可解释的值对变量进行编码。自然的想法是用目标变量的平均值进行编码。在我们的例子中，一周中的每一天和一天中的每一个小时都可以通过相应的平均广告数量进行编码。确保平均值仅在列车组上计算（或仅在过流交叉验证倍数下计算）非常重要，这样模型就不知道未来会发生什么。

让我们看看小时平均数



最后，将所有转换放在一个函数中





这太合适了！小时平均变量在火车数据集上是如此之大，以至于模型决定把所有的力量都集中在它上面，结果预测的质量下降了。这个问题可以用多种方法来解决，例如，我们可以计算目标编码，而不是整个列车组的编码，而是一些窗口的编码，这样最后一个观察窗口的编码可能会更好地描述当前的序列状态。或者我们可以手动放下它，因为我们确信这只会让事情变得更糟。

### 正则化与特征选择

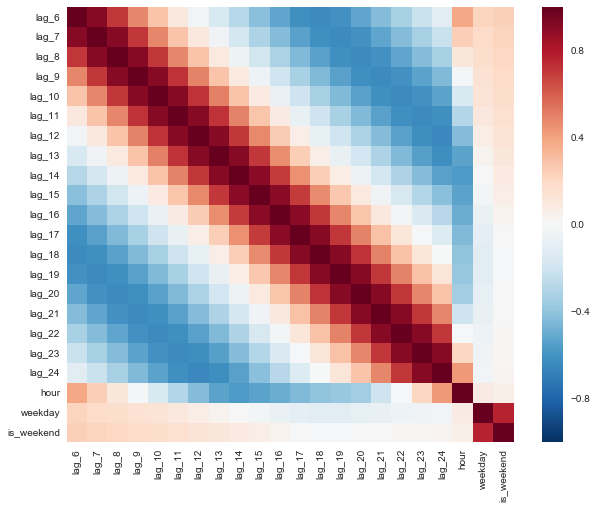
我们已经知道，并不是所有的功能都一样健康，有些功能可能会导致过度拟合，应该删除。除了人工检查外，我们还可以应用正则化。两种最流行的正则化回归模型是岭回归和套索回归。它们都给我们的损失函数增加了一些约束。

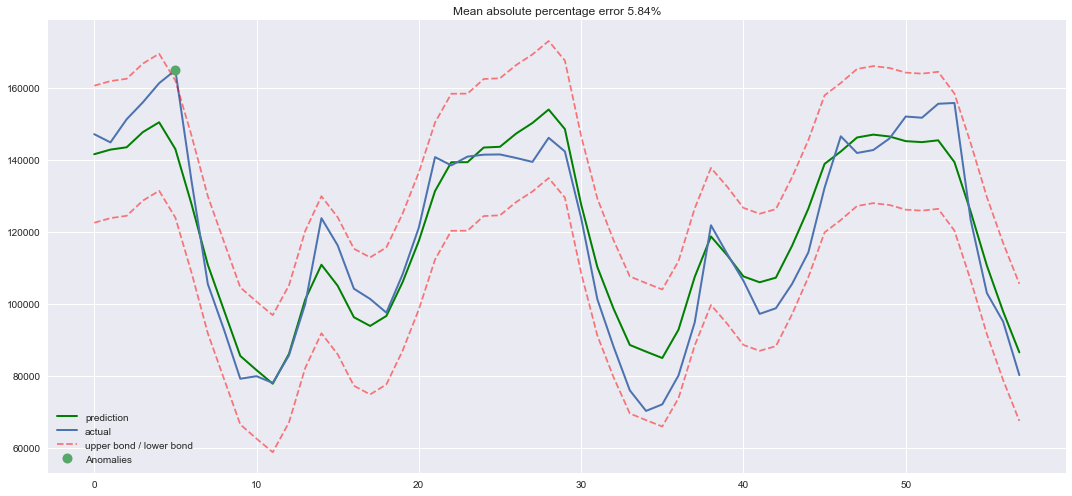
在岭回归的情况下，这些约束是系数的平方和乘以正则化系数。也就是说，系数特征越大，损失就越大，因此我们将尽量优化模型，同时保持系数较低。

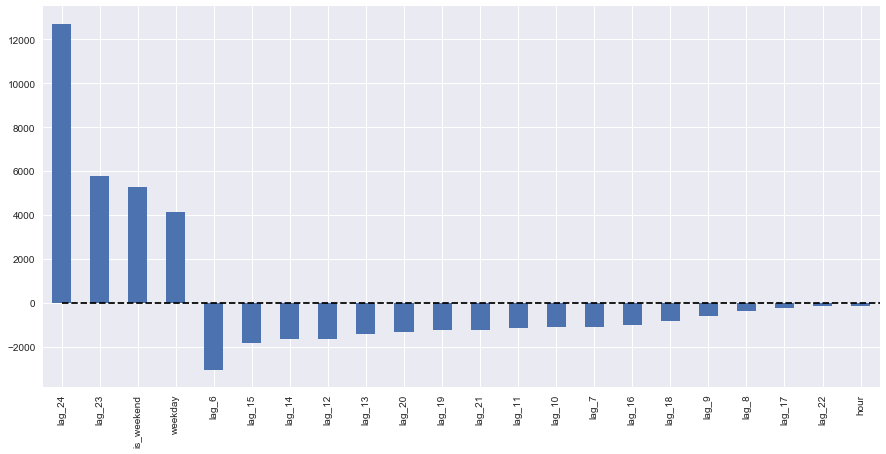
由于这种正则化有一个自豪的名字L2，我们将有更高的偏差和更低的方差，所以模型将更好地推广（至少这是我们希望会发生的）。

第二个模型 -套索回归，这里我们加入的损失函数不是平方而是系数的绝对值，因此在优化过程中，不重要的特征系数可能变成零，所以套索回归允许自动选择特征。这种正则化类型称为L1。

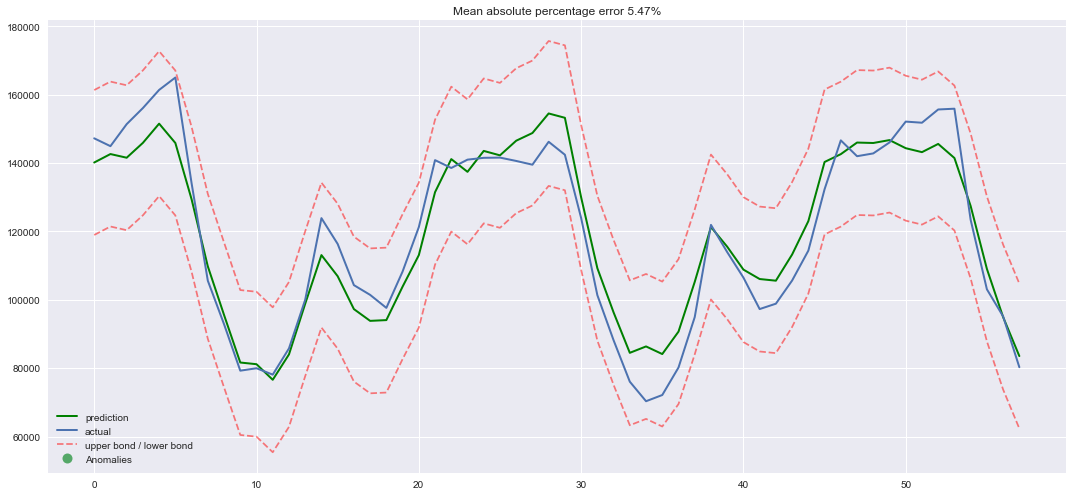
首先，确保我们要删除的内容和数据真正具有高度相关的特性

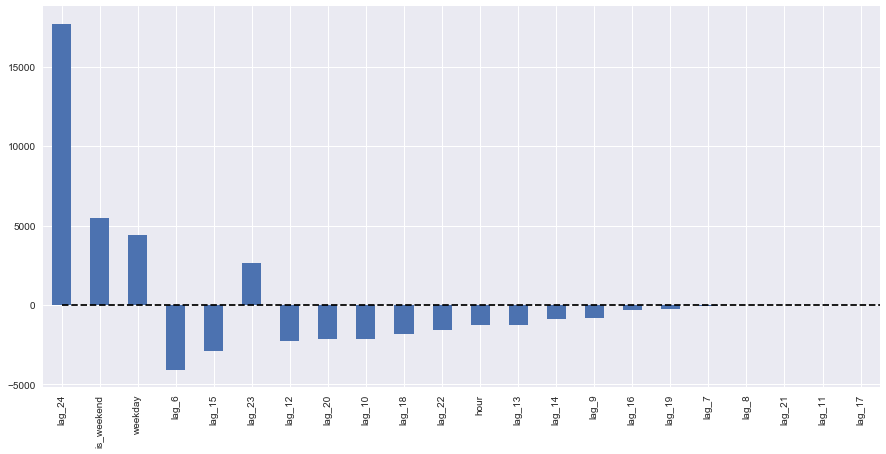






我们可以清楚地看到，当系数在模型中的重要性下降时，系数是如何越来越接近于零的（思想从来没有真正达到过零）



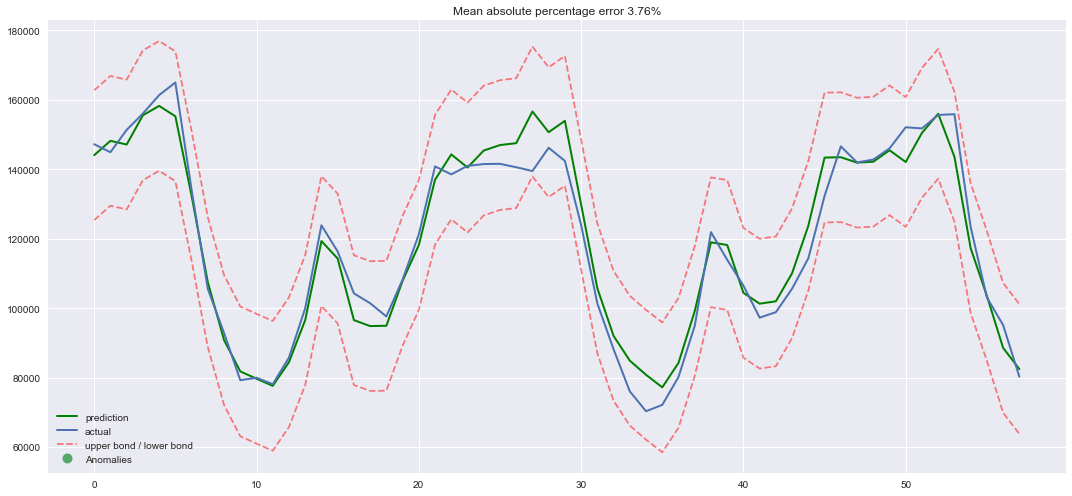


套索回归结果更为保守，从最重要的特征中去掉了23维滞后（也完全去掉了5个特征），这使得预测质量更好。

### 助推

为什么不试试XGBoost呢？





赢家来了！在我们目前尝试的所有模型中，测试集上的最小错误。

然而，这场胜利是决定性的，一旦你掌握了时间序列数据，适应xgboost可能不是最聪明的主意。与线性模型相比，基于树的模型通常不能很好地处理数据中的趋势，因此您必须先取消序列的格式，或者使用一些技巧来实现这种神奇的效果。理想情况下 - 使序列平稳，然后使用XGBoost，例如，可以使用线性模型分别预测趋势，然后添加XGBoost的预测以获得最终预测。

### 结论

我们熟悉了不同的时间序列分析和预测方法。不幸的是，或者说幸运的是，没有什么灵丹妙药可以解决这种问题。上个世纪60年代发展起来的方法（有些甚至在19世纪初）仍然与LSTM和RNN一起流行（本文不涉及）。部分原因在于，预测任务和其他与数据相关的任务一样，在许多方面都很有创意，而且肯定需要研究。尽管有大量的形式化质量度量和参数估计方法，但通常需要为每个时间序列寻找并尝试不同的方法。最后但并非最不重要的是，质量和成本之间的平衡很重要。作为一个很好的例子，这里提到的SARIMA模型在经过适当的调整后不会产生一次或两次惊人的结果，但可能需要许多小时的手鼓舞时间序列操作，因为在同一时间，简单的线性回归模型可以在10分钟内建立，给出或多或少的可比结果。

### Assignment #9

Full versions of assignments are announced each week in a new run of the course (October 1, 2018). Meanwhile, you can practice with a demo version: [Kaggle Kernel](https://www.kaggle.com/kashnitsky/assignment-9-time-series-analysis), [nbviewer](https://mlcourse.ai/notebooks/blob/master/jupyter_english/assignments_demo/assignment09_time_series.ipynb?flush_cache=true).

### Useful resources

* [Online textbook](https://people.duke.edu/~rnau/411home.htm) of the advanced statistical forecasting course of the Duke University — covers in details various smoothing techniques, linear and ARIMA models
* [Comparison of ARIMA and Random Forest time series models for prediction of avian influenza H5N1 outbreaks](https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-15-276) — one of a few where random forest applicability to the tasks of time series forecasting is actively defended
* [Time Series Analysis (TSA) in Python — Linear Models to GARCH](http://www.blackarbs.com/blog/time-series-analysis-in-python-linear-models-to-garch/11/1/2016) ARIMA models family and their applicability to the task of modeling financial indicators (Brian Christopher)

Author: [Dmitry Sergeyev](https://github.com/DmitrySerg). Translated and edited by [Borys Zibrov](https://www.linkedin.com/in/borowis/), and [Yuanyuan Pao](https://www.linkedin.com/in/yuanyuanpao/).