**微博不实信息的数据爬取与分析**

作者 陈哲安 王源毅

学院 理学院

专业 数学与应用数学

学号 2018212432 2018212466

班级 2018214101

2021年5月

# 数据爬取

# 数据处理

数据清洗

1.去重

针对爬取到的每条数据，都含有“站方判定”与“被举报微博发布时间”两项特征，我们先对“被举报微博发布时间”进行处理：由于爬取到的“时间”这一特征是具体年月日形式，我们将其按照月为划分，形成一项新的特征“时间戳”。

由于“站方判定”中，微博平台已经将相同的谣言用同一叙述方式总结出，如：博主A发布谣言“”；博主B发布谣言“”。这两者为同一谣言内容，在“微博社区管理中心”板块，“站方判定”内容中，自动将其归为“”同一表述。我们先根据上一步设置的“时间戳”特征，在相邻两个“时间戳”：即本月、下月两个时间内，进行“站方判定”内容的相似度对比，相同的或相似度高的数据删除，不同数据保存，每条谣言只保留最早一条数据。

由于微博内容中往往包含过多无用信息，拆分后的词袋中会包含与正文表达主题不相关的动词和实词，这些词会降低对微博正文主题间相似度的判定，因此我们使用改进后的杰卡德相似度计算两条微博内容的相似度。



其中A，B是两个文本拆分后的词组集合；|·|是集合中元素的数量

数据中还有部分谣言，显示“被举报内容已被删除”，我们直接删除了这些数据。处理后由9986条数据减少至1676条

2.简化与构建模型

在处理后的数据中，我们对原始的“微博内容”进行分析。首先导入停用词表，将所有微博正文拆解成单词，词袋中共有9629个元素。再对拆解后的单词集合进行操作：通过调用jieba库，我们统计了词袋中每一个词的TF-IDF取值，选取了排名最高的2000个词作为最终词袋的组成元素。下图为词袋中的最终2000个词。



根据构建成的词袋，对1676条数据分别构建成相应的词袋向量，最终构成1676\*2000的词袋矩阵，用于后续LDA主题模型的操作。

**3.疫情相关数据筛选**

我们以“疫情”这一主题为研究目标，选取了“疫情”、“冠状病毒”、“疫苗”、“死亡”、“感染”、“医院”和“确诊”作为关键词，筛选出了

**特征提取**

根据上一步处理完的数据，我们设计了9项特征值来衡量某一微博正文，以便进行后续分析。这9个特征值是：

被举报人性别（0是女性，1代表男性）、地区（1代表南方地区，2代表北方地区，0代表其他）、信用等级（1代表正常，0代表中，-1代表差）、是否会员（0代表否，1代表是）；微博正文有无超链接；字符数、句子数、单词数量、平均句子长度。其中：

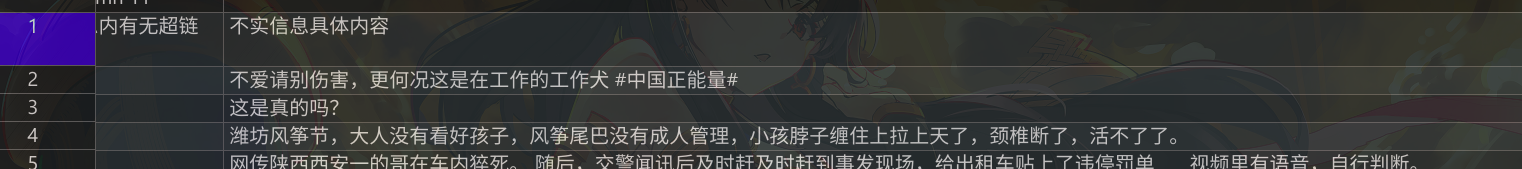
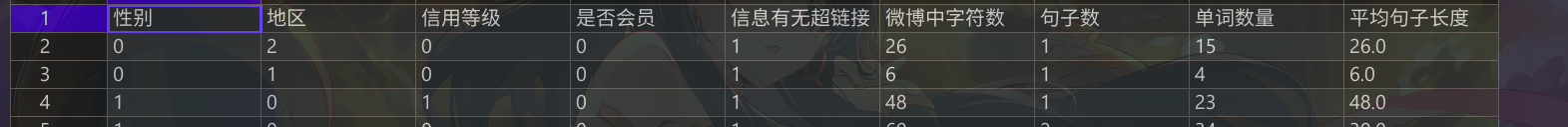
字符数:将每个标记的字符数相加

计算句子数:计算句子的数量(以句点分隔)

单词数量:文本拆分后单词的个数

平均句子长度:句子长度的总和除以句子的数量(字数/句子数量)

以前三项微博正文为例，它的9项特征值和其正文内容如下：



**TF-IDF与降维**

在“数据清洗”过程中，我们构建出了包含2000个重要词汇的词袋和对应微博正文的矩阵。我们利用TF-IDF对词袋归一化，算出了所有词汇的TF-IDF得分，原理以及实现过程如下。

TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。**字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。**

**其中，词频 (TF)** 指的是某一个给定的词语在该文件中出现的次数。这个数字通常会被归一化(一般是词频除以文章总词数), 以防止它偏向长的文件。（同一个词语在长文件里可能会比短文件有更高的词频，而不管该词语重要与否。）公式为：



**逆向文件频率 (IDF)** 的主要思想是：如果包含词条t的文档越少, IDF越大，则说明词条具有很好的类别区分能力。某一特定词语的IDF，可以由总文件数目除以包含该词语之文件的数目，再将得到的商取对数得到。



某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。



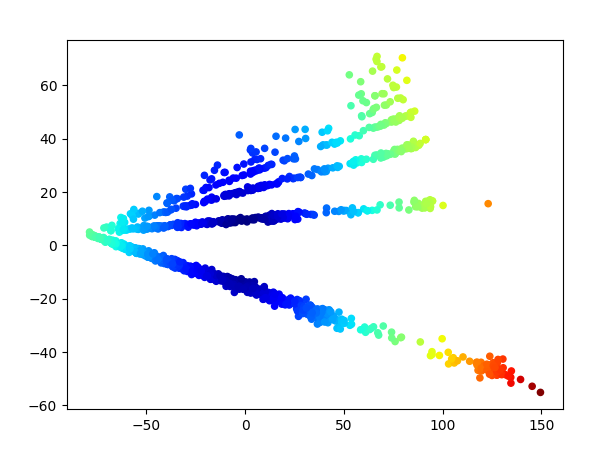
（代码）

# 数据分析和可视化

数据聚类

1. 基于词袋模型矩阵的LDA模型
2. 基于数据特征的GMM分类

根据数据处理过程中使用的主成分分析法，我们将数据的特征通过两个主成分来代替。画出我们每条微博数据在主成分构成的空间中的散点图，如下图所示。



图中的各个点之间有明显的聚类趋势，其中有四组点呈现明显的聚集情况，这些同类的点在二维空间分布上有明显的线性趋势；另外有部分点较为分散，但任然有聚集的趋势。

根据图像中各组点的分布规律，我们进行聚类分析。首先使用K-means聚类，分析发现，聚类效果并不符合直观上的各组点的分布情况（K-means聚类结果如下图）。

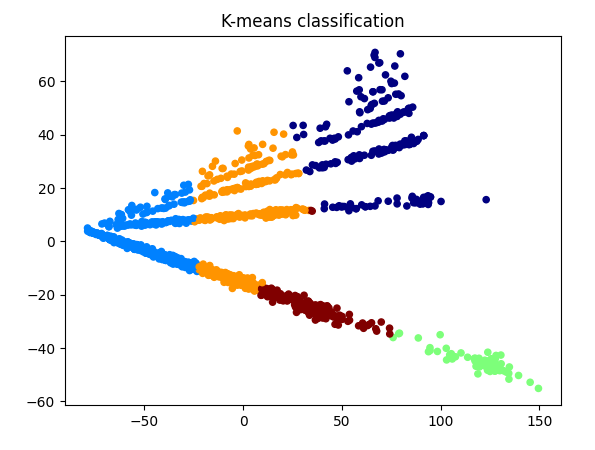


图3.K-means分类、同一类用相同颜色表示，图中分类与实际分类趋势明显不相符

分析后发现，K-means聚类在二维空间上，是通过确定圆点和相应的半径，用圆形去覆盖空间中的样本点，不适合于我们当前数据的这种离散的线性分布，因此我们考虑使用高斯混合模型来进行分类。所谓高斯混合模型，**本质就是融合几个单高斯模型，来使得模型更加复杂，从而产生更复杂的样本。理论上，如果某个混合高斯模型融合的高斯模型个数足够多，它们之间的权重设定得足够合理，这个混合模型可以拟合任意分布的样本。它的具体形式为：**

假设混合高斯模型由K个高斯模型组成（即数据包含K个类），则GMM的概率密度函数如下：



其中，是第k个高斯模型的概率密度函数，可以看成选定第k个模型后，该模型产生x的概率；是第k个高斯模型的权重，称作选择第k个模型的先验概率，且满足。

我们调用sklearn库中EM算法对GMM模型进行参数估计，迭代次数设置为100次，最终聚类效果如下图所示。

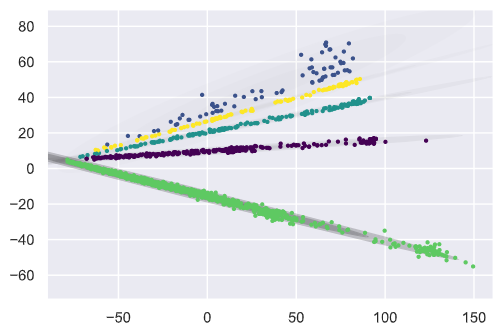


图4.GMM分类结果。同一类用相同颜色表示

从图中看出，模型很好的聚集成5类数据，且每组数据聚类趋势明显。我们用标号0~4表示5个类的类别名称，输出每条微博正文最终分析的类别结果。下图为部分微博正文与主题类别的关系。

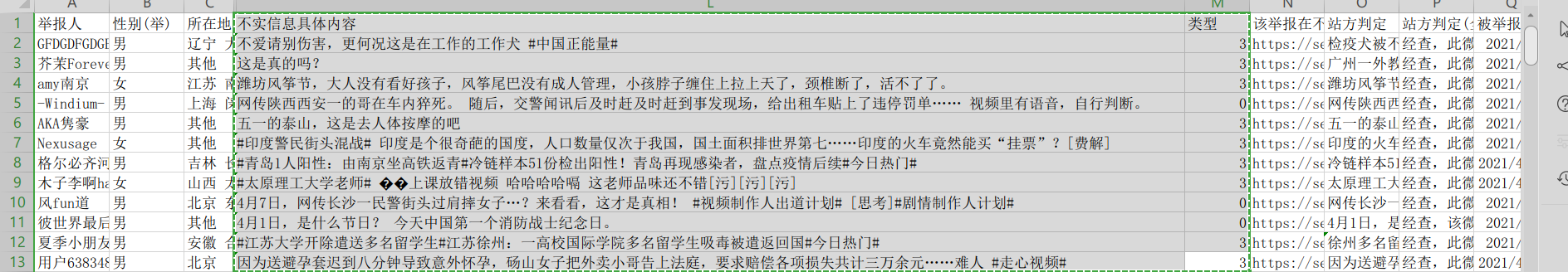
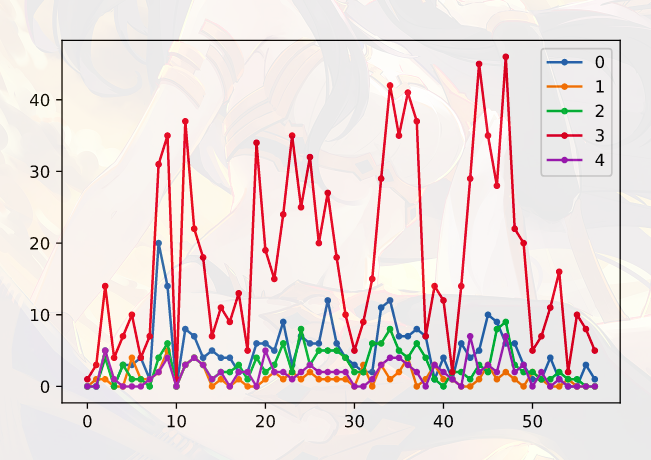
****

图5.部分分类结果

我们对每类数据进行时间序列分析。首先画出数据时序图，以时间\月为横坐标，做出每月该类微博谣言数量随时间变化的折线图。如下图所示。可以看到谣言数量随时间做周期变化，对原数据做ADF检验，发现检验结果p值为0.065，说明原数据不是纯随机性数据，因此对数据进行差分处理。

以第三类数据为例，我们可以看到第三类数据主要和“孩子”“死亡”等关键词有关，主题是青少年和死亡。第三类数据在时序图中由红色曲线表示出。做出的一阶差分结果和二阶差分结果如图9。

再对差分后的数据做ADF检验，发现一阶差分后p值已经小于0.05，并且从图中看出，一阶差分与二阶差分相比随机性变动不大，因此我们使用一阶差分的数据,并采用ARIMA模型对数据进行分析。

****

**图6.谣言数量随时间变化趋势**

****

**图7.第三类谣言的关键词**

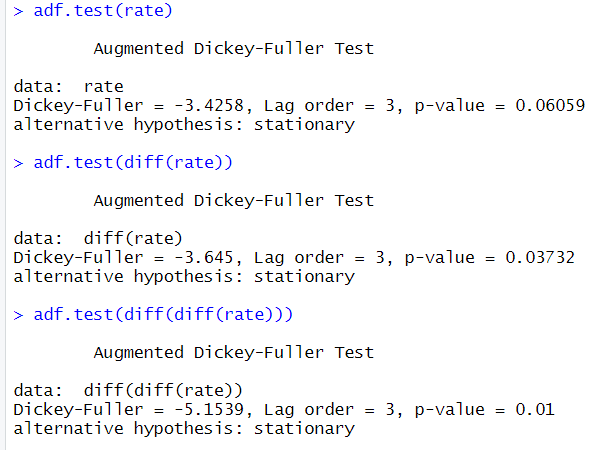


图8.原始数据、一阶差分和二阶差分的ADF检验结果

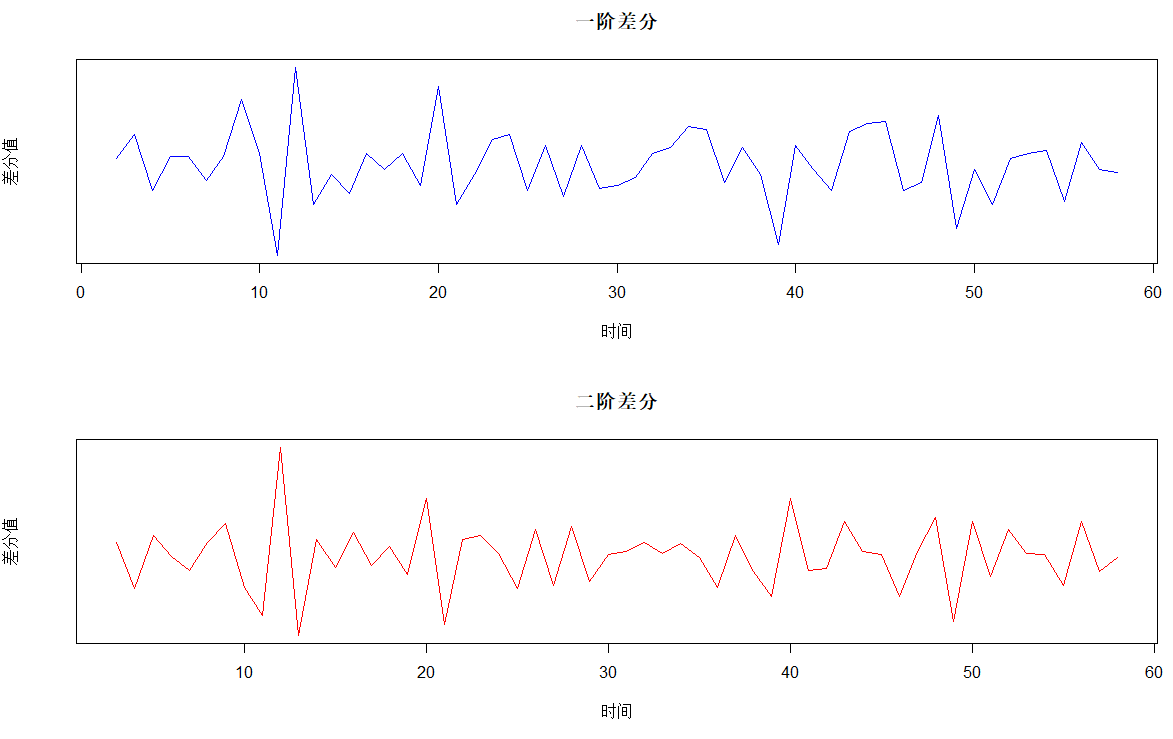


图9.一阶差分和二阶差分结果图

对于一阶差分后的数据，我们判别其ACF与PACF的分布情况。从图10中可以看出，ACF主要在取1时超出标准范围，PACF主要在取值为1、2时超出标准范围，因此我们讨论一下几个模型：ARIMA(1,1,1)，ARIMA(1,1,2)和ARIMA(1,1,3)

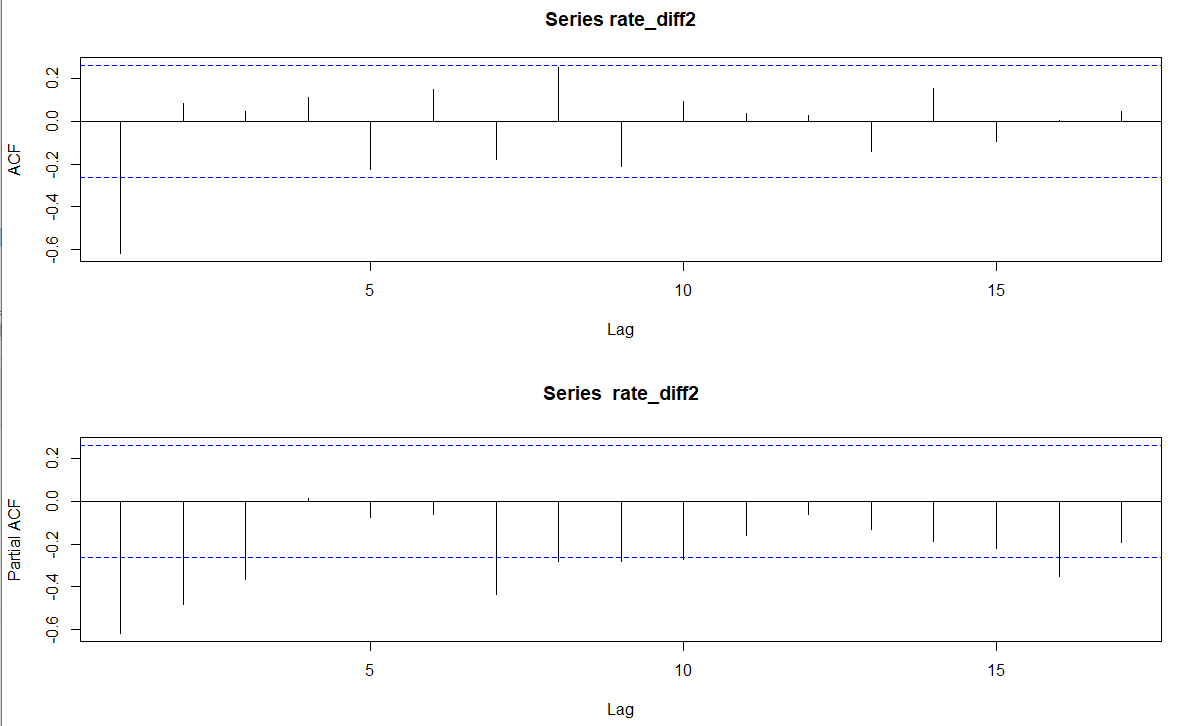


图10.ACF与PACF分析

我们采用AIC与BIC得分来判定模型效果的好坏，得分越低，效果越好，测评如下。

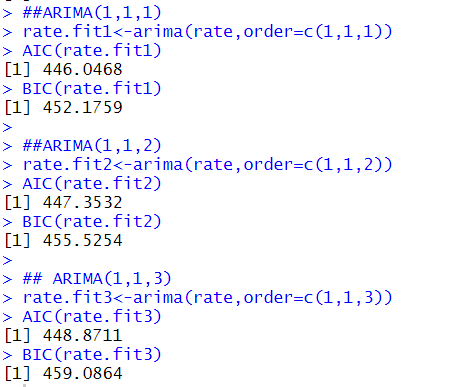


图11.模型测试结果

可以看出，模型ARIMA（1，，1，1）效果最好。进一步对模型检验，检验模型的残差是否时白噪声。从图12和图13可以看出，通过qq图展示结果说明残差服从正态分布，而Ljung-Box检验证明残差为白噪声。

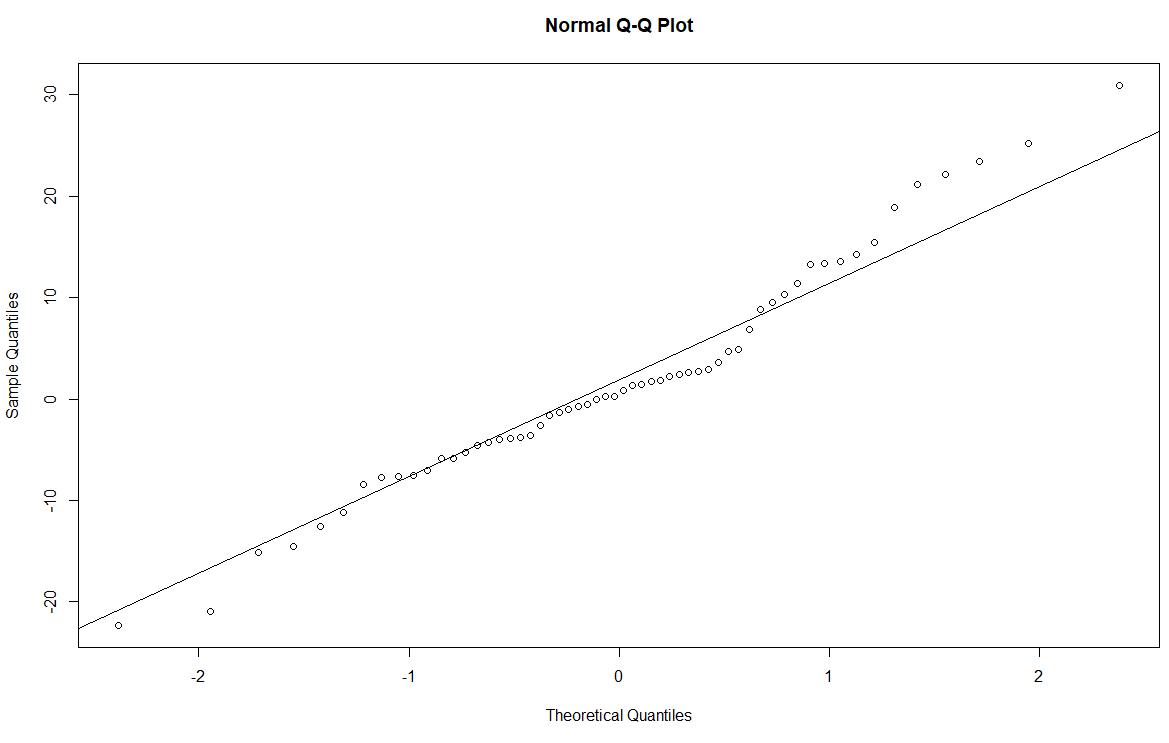


图12.模型残差的qq图

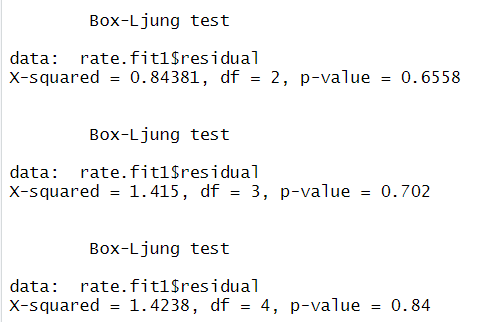


图13.Box-Ljung检验结果

利用ARIMA模型进行预测，可以看到未来几个月可能出现的该类谣言数量趋势。

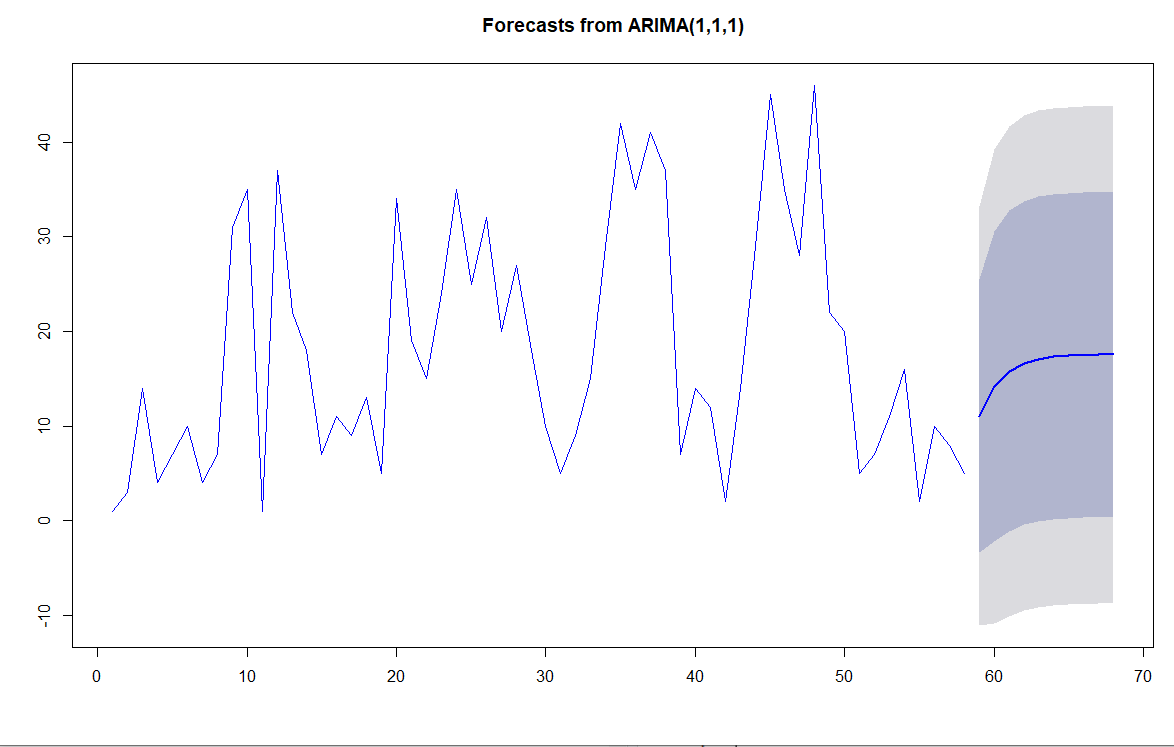


图14.ARIMA模型的预测结果

# 数据分类

在已有的谎言数据集基础上，我们又爬取了736条正确的微博数据，并提取出了每条数据对应的九个特征值，与谎言数据集合在一起构成了新的数据集。我们将数据集随机打乱后，按照训练集：验证集：测试集=8:1:1的比例进行划分，用于训练分类器。

我们选取了支持向量机（SVM）做为分类器，以每条数据的9个特征值为输入，微博内容的真假判定为输出进行训练，求出该模型确定的超平面。

支持向量机（SVM），实质上就是在一个n维空间中存在的一组点，在这组点之间生成一个n-1维的超平面，将这组点分割为两部分，每部分点到这一超平面的距离之和达到最小，而求解这一超平面的过程就是建立SVM模型的过程，模型的最终成果就是这一超平面。

具体到我们的实验中，我们首先在我们的训练集上，训练出100个模型。这些模型的参数，分别是：松弛变量系数从0.1到1，gamma参数从0.1到1，每隔0.1取一个值，两两组合共组成100个模型。

（训练集代码）

利用我们的测试集，我们检验每一个模型的拟合效果，发现在松弛变量取0.9，gamma取0.4时模型拟合效果最好，正确率达到79.%。下图是测试过程和最佳模型的测试结果。

（测试集代码）

再使用验证集对选择到的松弛变量取0.9，gamma取0.4的模型进行验证，得到最终正确率为（），说明模型整体效果良好，准确率可以接受。下图为验证过程。

（测试集代码）

(考虑画个图)

# 总结