# Complaint Trends QA Data

## Overview

This document explains the usage of a small dataset that is used to test the complaint trends UI. Since the real UI only displays

## Overview of QA Data

Real complaints have been taken from the Consumer Financial Protection Bureau database and mapped to synthetic customers with associated names, ages, US States etc. The counts and numbers of these complaints have been manipulated so that trends can be detected.

The following trends should be visible

|  |  |  |  |
| --- | --- | --- | --- |
| **risk** | **attributes** | **count** | **trend start** |
| High | Sales Practice in CA (Age Group: 30-40) | 225 | Jun 02, 2017 |
| High | Poor Rude Treatment in TX | 477 | Jun 14, 2017 |
| High | Sales Practice (Age Group: 30-40) | 892 | Jun 04, 2017 |
| High | Poor Rude Treatment (Age Group: 30-40) for Closing process | 101 | Jun 15, 2017 |
| Medium | Burdensome Request (Age Group: 0-30) for Underwriting process | 360 | Apr 29, 2017 |
| Medium | Different from Expectation for Appraisal process | 1403 | Jun 12, 2017 |
| Low | Responsiveness in DC for Underwriting process | 101 | Jun 15, 2017 |
| Low | Responsiveness for Underwriting process | 326 | Jun 14, 2017 |

Trends exist for the time period from 1st April to 30th June 2016.

**Assumptions made in this test data**

Each complaint is only assigned to one category (e.g. Delay, Poor treatment etc) and one process (Loan Application, Underwriting etc.) In live usage its possible (and likely) that complaints can have multiple classifications but the data is kept simple here for QA purposes.

## Pre-Processing

1. Check that the config file is up to date and has sensible values. In file **/ml.trend-detection/config/config.cfg:**

The trend detection algorithm has a large set of parameters. Most of these are used to prevent noisy, small or uninteresting trends from crowding out the interesting trends. This table gives a list of parameter values that were used to create the dataset for reference:

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Suggested value |
| fields\_list | List of all the fields that are checked for trends. All combinations of all these fields are checked | THEME,SUB\_THEME,PRODUCT\_REF,GEO\_REF,CUSTOMER\_AGE |
| min\_records | The minimum total number of complaints that must exist in a time-series before the algorithm will check for a trend | 10 |
| min\_per\_day | The minimum number of complaints that must exist in the ‘trending’ part of the time series on any one day before the algorithm will check for a trend.  ie ‘there must be at least 10 complaints on at least one day before this time series will be included as trending in the UI’ | 10 |
| pct\_change\_threshold | The change between the average value and the average value in the trending part: This gives the minimum change before the algorithm will check for a trend. For example, 0.2 = there must be a change of at least 20% up or down from the average count before the trend is displayed in the UI | 0.2 |
| min\_risk\_filter | Each detected trend is given a risk score between 0 and 85 %. Trends with risk scores below the min\_risk\_filter value will not be shown in the UI. | 0.20 |

1. In file '/home/sifsuser/lib/sifs.spark.properties': Check that API url ComplaintTrendsREST = <https://xx.xxx.xxx.xxx:9443/complaintsservices/surveillance/v1/complaint/createTrend> points to the correct trends db2 service

## Viewing the trends in the UI

The discovered trends should look like this. Depending on the config parameters used there may be more or fewer trends shown.

