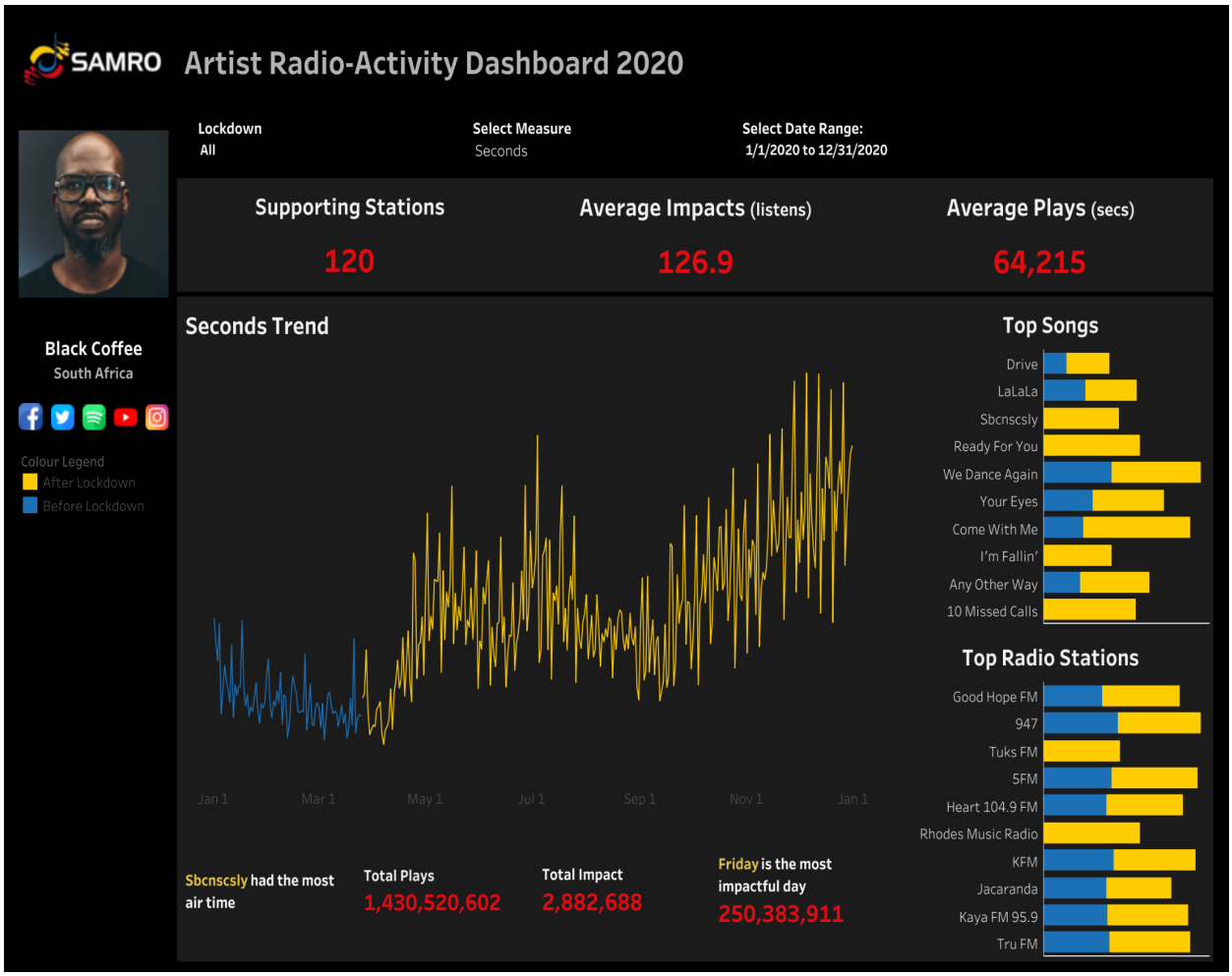
CETM25

Artist Radio-activity Dashboard Report

Mark Stent - 4 January 2022

Wordcount: 2526



Introduction

This report covers the process and management of the SAMRO (South African Music Rights Organisation) Radio-activity dashboard I have created as a proof-of-concept design for this module. I was permitted to use a one-year time frame, as it would provide better insight into the data and the problem. Please refer to my first assignment in Appendix 5 for this project's initial goals and objectives. The dashboard is available here:

Tableau dashboard

All data used in this project has been collected and used according to the GDPR (2016/679) and the Data Protection Act (1998).

**I have created a Github repository for the files in the appendix:*

Github CETM70 repository

Section 1

Selection Of The Visualisation Software And Hardware

From a software perspective, the project breaks down into two layers: Exploratory data analysis (EDA) and visualisation/dashboarding.

Python is an open-source language that is excellent for analysis work. It is widely accepted, extremely portable, and offers various libraries to perform any required task. Libraries are reusable chunks of code that assist in speeding up or simplifying problem-solving. These libraries are the main reasoning for using Python in this project. The specific libraries used are listed below:

- Numpy offers a comprehensive suite of mathematical functions and tools used in this project for statistical analysis and other analysis functions.

- Pandas is a data manipulation and data analysis tool. This project uses Pandas to collate, manipulate, and manipulate the dataset for analysis.
- Scipy is a library offering fundamental algorithms for scientific computing. This project uses Scipy for statistical analysis.
- Statsmodels offers classes and functions that cover many statistical models. For example, this project uses Statsmodels for time series analysis.
- Outlier analysis was done with Facebook Prophet.

Jupyter is an integrated development environment (IDE) used to compile code in an 'all in one place' manner, allowing equations, images, plots, and code to be hosted in a single interface; this allows for faster analysis and exploration of the data in a data science project. These qualities make Jupyter notebooks ideal for this project.

Tableau handled the visualisation/dashboarding component. Tableau is widely used for business intelligence and is known for creating powerful visuals backed with excellent support for big data. It is also relatively easy to use and understand. In addition, Tableau has a public version that allows the hosting of smaller projects on a shared server with all of the enterprise features enabled; this made it ideal for use in this project.

Product Development / Software Engineering Methodology

Agile is a mindset that enables users to adapt to an ever-changing environment. Scrum is a lightweight framework that helps people, teams, and organisations generate value through adaptive solutions for complex problems in an agile environment (scrum.com, Nov 2020). Since this project is exploratory, the agile approach fits perfectly.

The scrum methodology calls for short sprints of time-boxed work to complete a defined amount of work and an organised process to manage time and project outputs. Accordingly, I followed the below agile stages:

- Stage one - Requirements gathering and product backlog creation

- Stage two - Exploratory data analysis
- Stage three - Tool development and creation

Since this was a small project with few stakeholders, all three stages were assigned to one sprint of 4 weeks. The project status underwent daily review, updating current progress iteratively in the backlog file (Appendix 2). The sprint review meeting would decide the next steps for the project.

Radiomonitor.com provided the original data in a .CSV file. This data underwent the following pre-processing to clean the data for analysis and output into Tableau:

- Pandas was used to create a data frame.
- Date column converted to a date/time datatype.
- Leading and trailing whitespace removed from around the data
- Remove all channels where impacts = 0 as they are assumed not to have official RAMS figures.
- Impute null values with 0's
- Convert 'impacts' and 'seconds' columns to numerical data types
- Get rid of rows that don't make sense, i.e. where there are no seconds of play but have > 0 impacts (this could be due to data entry errors)
- Changing of column names to more readable names
- Statistical analysis of data to check for any statistically significant events and changes over the period includes tests for normal distribution, Bartlett's test for differences in variance and t-test for difference in means (Appendix 3).

System Testing

Testing of the dataset broke into two phases:

- Phase 1 - Testing and exploration of the raw data set for insights (see Appendix 6)
- Phase 2 - Testing the final visualisation

The goal of phase 1 was to do a deeper dive into the data, run some initial statistical tests to check the shape and spread of the data, and do a time series analysis to check for any seasonality or outliers.

The first test checked if the 'seconds' and 'impacts' columns fitted a normal distribution. Using a combination of analysis histograms, QQ plots, kurtosis, and skewness metrics, we can assume that 'seconds' follows a normal distribution, while 'impacts' does not.

Bartlett's (Snedecor and Cochran, 1983) test for significant change in variance before and after the lockdown was tested and found to be significant, implying that Black Coffee's airtime increased after the lockdown period. However, more analysis is needed to decide if the lockdown was the reason for this increase.

A t-test (Box, Joan. Fisher, 1987) for change in means after the lockdown date was performed on the data and was found to be significant. This test implies that mean seconds played increased after the lockdown date; this could be due to more people listening while at home during the lockdown. More investigation is required.

Time series analysis showed an overall upward trend in airtime. A definite seasonality exists on Saturdays; this makes sense as weekend radio shows have more freedom in playing tracks for extended periods, with less talking.

A Facebook Prophet model using 99 percent confidence intervals of predicted means was fitted to detect outliers. Only seven outliers occurred, and all on Fridays.

Phase 2 used the cleaned and processed dataset after testing, decreasing the probability of data-related errors in the visualisation tool. In addition, each feature was tested using every

permutation of interactivity on the visualisation tool for 'breaking' and accuracy. As a result, no visible errors or software errors occurred.

For production versions of the tool, a layer for error checking and 'broken data' needs to be developed.

User Evaluation

This project was conducted using Scrum methodology. In Scrum, the sprint review is employed to review the outcomes of a sprint and determine future adaptations or plans. Inviting the product users to this review is an excellent way of validating the project features and results.

The best way to get honest feedback from users is to hand them the tool and let them use it. Then, after they have had time to 'test drive' the product, feedback is collected.

Feedback is documented by asking questions to the users that align with the original project goals, for example:

- Do you understand what the dashboard is depicting?
- Is the layout intuitive?
- Is there anything you don't understand or doesn't make sense to you?
- Do you find the experience pleasant/unpleasant? Why/why not?
- Do any new ideas come to mind that could be added to the tool to make your job easier?
- What insights did you get from using the tool?

The vital part of collecting this kind of information is to make sure that the user feels safe expressing his opinion, setting clear goals for the meeting, and keeping the interactions light-hearted and non-judgemental; this will encourage honest and objective feedback.

Matching the results of this review session against the initial project goals would dictate the success of the first sprint. Any changes or new requirements would be updated in the project backlog and included in the plan for the second sprint.

Since the primary goal of this project was exploratory, the results of the first sprint review would decide the direction of the project going forward.

Section 2

Time Management

I am currently working full time as a data analyst and doing my master's degree part-time; I also have three daughters and a family. With this in mind, time management has been a massive influence on the quality of my work output. However, with the decision to complete this assignment with an agile mindset and using Scrum processes, achieving timeous goals was made much more straightforward. The Pomodoro technique was another tool I used to help me meet my goals.

The project backlogs (Appendix 2) main aim is to document project outcomes, assign priorities, and estimate the effort required for each user story. I used this document to plan the time I spent on each story and tried to keep it within the target hours as much as possible. In addition, because each story was assigned a priority, I was able to focus on the most critical tasks first. If any tasks overshot the estimation, I re-adjusted the estimates based on what was needed.

The Pomodoro technique was developed in the 1980s by Francesco Cirillo to encourage users to work with the time they have rather than against it.

The technique aims to achieve total concentration with minimal distraction for short periods; these periods are affectionately named 'pomodoros'. I blocked my time into 25-minute 'pomodoros' and 5-minute breaks. During each 25-minute block, I allowed no distractions from social media, phone calls or anything else. I kept track of these sessions and aligned them with the estimated hours in the project backlog.

Overall, this process worked very well, and I was able to keep within close distance of the original backlog estimations. I put this success down to planning and using the Scrum methodology to achieve the project's goals.

If I had to redo this project, I would spend more time planning and framing the questions before jumping straight into the analysis.

Risk Assessment

The project's initial goal was exploratory; therefore, the overall risk is very low, but this would change if the proof-of-concept goes into production. After discussions with the stakeholders in the first sprint planning meeting, highlighted the following risks:

- Accuracy of data - inaccurate data could lead to wrong decisions. Radiomonitor provided the data and has assured that the data is around 95% accurate. There are some minor anomalies where the algorithm detects plays under one second and tracks them as impacts. The data pre-processing cleaned out these anomalies.
- Choice of the artist - Black Coffee is a well-respected artist, and his radio airtime support may not be representative of the South African radio market. In the next sprint, other artists will be tested and added to the dashboard that are more representative of the typical South African artist population.
- Radio station outstanding payments - not all radio stations are up to date with paying royalties, and the figures used in the tool may not translate to accurate royalty payments. If required, the team will undertake a more profound analysis in the next sprint to assess the impact of non-paying radio stations.
- Complexity and cost of the next phase of the project - Depending on how the requirements evolve for the final product, complexity and expenses for the technology required to deploy the visualisation tool may be too expensive.

Quality Control

Using Scrum as a project management methodology requires constant evaluation and re-evaluation in an iterative process at each project stage.

Building plans for quality assurance to the customer and quality control for the project is critical for meeting project deliverables. Using various metrics or questions at each iteration will keep the quality goals of the project in check. Questions include:

- Is the visualisation tool meeting the initial project goals?
- Is the visualisation tool doing what it is supposed to do?
- Is it responding to the questions it is supposed to answer?
- Is the customer satisfied with the output?

In each feedback meeting, these questions are repeated to ensure that the project is going in the correct direction. When a quality problem is identified, the process is reactionary, and solutions to the problems are suggested—for example, modifying the path rather than completely recreating some sections by catching deliverables that don't satisfy the agreed-upon standards is more efficient.

In short, for this project, quality control standards were measured by how closely the visualisation tool met the project goals.

Customer Relationship Management

In a Scrum environment, customer management is managed through the project backlog (Appendix 2), which contains a list of outcomes for the project. In addition, the project backlog provides a system for cleaning up customer service concerns: defects and problems are managed within the sprint period. Issues are measured, reviewed and improved.

Each iteration of the visualisation tool would involve the SAMRO and stakeholder teams, addressing potential issues and questions. Any changes would be added to the project backlog to manage current or future sprints.

Product Market Strategy

The first sprint review discussed the results of the initial exploratory analysis. The project goals were as follows:

Functional requirements:

- Visualise changes in spin data and radio impact data, enabling users to view changes over time.
- The tool should allow filtering individual radio stations or subsets of stations.
- Create a view to visualise the distribution of spins (measured in 'seconds played') across radio stations for individual titles to see the changes in radio support of titles.
- A view of summary spins per weekday to see how listenership is affected by the day of the week.
- Annotate visualisations with essential dates such as lockdown start date.

Non-Functional requirements:

- Views should be easy to use and uncomplicated to understand.
- The visualisation should be secure and reliable for the reporting to be accurate.
- The tool should cover one year starting on 1 January 2020. The first South African lockdown began on 26 March 2021.
- The date range must be interactive.

The visualisation tool met each of these requirements. But, unfortunately, the visualisation tool did not give a definitive answer to the question about lockdown radio support for Black

Coffee. Therefore, deeper analysis (possibly a change-point analysis) and more data are needed to answer this question accurately. While Black Coffee's radio support has increased post lockdown, other factors may have contributed to this, such as releasing a new album, media attention and record label plugging into the radio stations.

Despite the visualisation tool not answering the initial question, the SAMRO team have seen other applications for the visualisation tool that could help them with more comprehensive business intelligence. Some of the applications and benefits are:

- Trends regarding radio station spins and impacts
- Trends regarding artist
- Trends in music genre support
- The tool would be helpful for funding and advertising discussions.
- The information could be used to obtain sponsorship for artists from marketers and advertisers.
- The dashboard could assist artists in future music and career decisions.
- The dashboard is helpful for discussions regarding royalty charges for radio stations.
- Radio stations do not always report accurate spin numbers. The visualisation tool could assist in reconciling actual versus reported figures.

The SAMRO team has decided to allow me to develop this project further, and I hope to put the final dashboard together for them by the end of the first quarter of 2022.

Appendix

1 - References

Box, Joan Fisher. “Guinness, Gosset, Fisher, and Small Samples.” Statistical Science, vol. 2, no. 1, 1 Feb. 1987, 10.1214/ss/1177013437. Accessed 5 Apr. 2021.

Jones, Douglas H. “Statistical Methods, 8th Edition George W. Snedecor and William G. Cochran Ames: Iowa State University Press, 1989. Xix + 491 Pp.” Journal of Educational Statistics, vol. 19, no. 3, Sept. 1994, pp. 304–307, 10.3102/10769986019003304. Accessed 2 Apr. 2020.

“NumPy.” Numpy.org, 2021, numpy.org/. Accessed 28 Dec. 2021.

“Pandas - Python Data Analysis Library.” Pydata.org, 2021, pandas.pydata.org/. Accessed 28 Dec. 2021.

“SciPy.” Scipy.org, 2021, scipy.org/. Accessed 28 Dec. 2021.

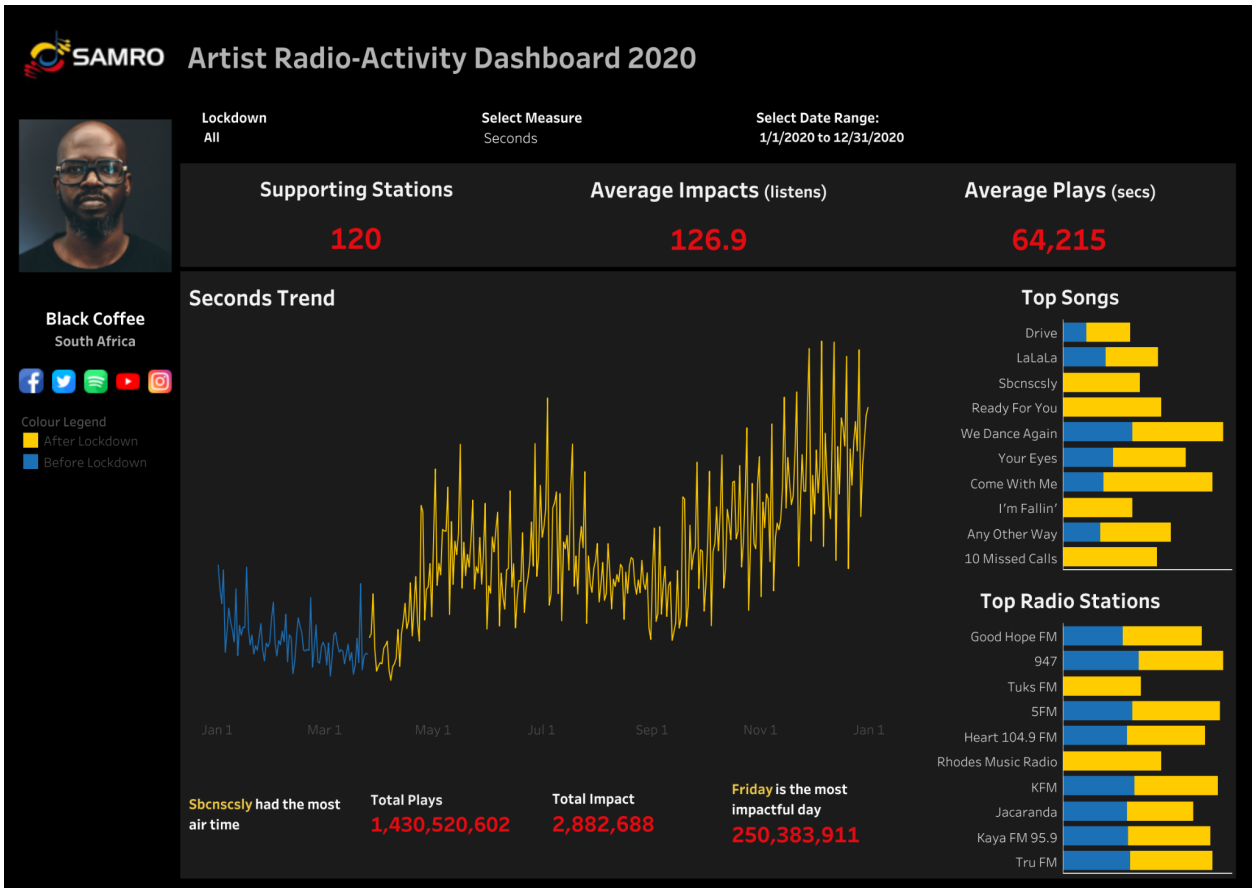
“The Pomodoro Technique® - Proudly Developed by Francesco Cirillo | Cirillo Consulting GmbH.” Francescocirillo.com, 2020, francescocirillo.com/pages/pomodoro-technique. Accessed 29 Dec. 2021.

“We’re Changing the Way You Think about Data.” Tableau, 2019, www.tableau.com/trial/tableau-software?utm_campaign_id=2017049&utm_campaign=Prospecting-CORE-ALL-ALL-ALL-ALL&utm_medium=Paid+Search&utm_source=Google+Search&utm_language=EN&utm_country=SA&kw=tableau&adgroup=CTX-Brand-Core-EN-E&adused=504564000692&matchtype=e&placement=&gclid=Cj0KCQiA5aWOBhDMARIsAIXLlkd0j5X_GXBAI2fygP9VhGeHzmgDQFZNTeYpgYtHgwip3SCmzmxi_YcaAqUyEALw_wcB&gclsrc=aw.ds. Accessed 28 Dec. 2021.

2 - Project Backlog

PRODUCT BACKLOG						
User Story ID	User Story	Estimate (size)	Priority	Sprint	Task owner	Estimated effort (hours)
US001	As a data analyst I need to understand the data better	Medium	1	1	Mark Stent	4
US002	As a data analyst in I need to work with a clean and 'unbreakable' dataset	Small	1	1	Mark Stent	3
US003	As a customer, i need to be able to view artist radio-activity in terms of impacts and seconds	Large	1	1	Mark Stent	10
US004	As a customer, I need to be able to view artist-radio activity before and after the 26 March 2020 lockdown period	Medium	2	1	Mark Stent	4
US005	As a customer, I need to be able see radio-activity insights and metrics in a simplified format	Small	3	1	Mark Stent	2
US006	As a customer, I need to be able to view top songs and radio stations before and after the 26th March 2020 lockdown period	Medium	5	1	Mark Stent	4
US007	As a customer, I need to be able to interactively change dates and measures	Small	5	1	Mark Stent	1
US008	As a customer, I need to be able to visually see the impact of the lockdown period	Small	1	1	Mark Stent	2

3 - Dashboard screenshot



4 - Glossary

RAMS - Radio Audience Measurement - A metric used to measure radio impacts

SAMRO - South African Music Rights Organisation

SPIN - A single play of a music track

SECONDS PLAYED - The total number of seconds of play of a track for a specific spin.

IMPACTS - A metric created by The Broadcast Research Council of South Africa (BRC), a non-profit organisation. 'Impacts' tracks the influence of the radio station by listenership and demographic using Radio Audience Measurement (RAMS) figures.

5 - Assignment 1

The previous assignment for this module can be found on Github if needed for reference.

6 - Jupyter Notebook

Starts on the following page (19 pages):

SAMRO Artist Radio-Activity Dashboard pre-processing and testing

Import relevant packages for preprocessing

In [147]:

```
import numpy as np
import pandas as pd
import scipy.stats as stats
from scipy.stats import shapiro
import matplotlib.pyplot as plt
import pylab
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller, kpss

from adtk.detector import SeasonalAD
from adtk.data import validate_series


import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

import CSV file and output data

In [11]:

```
data_original = pd.read_excel("Black Coffee - 2020.xlsx")
data = data_original.copy()
data.head()
```

Out[11]:

	Station	Artist	Title	Imp's	Sec	Date/time
0	North West FM	Black Coffee feat. Nakhane Toure	We Dance Again	21,000	0	01 Jan 2020 - 00:07
1	MFM	Black Coffee feat. Shekhinah	Your Eyes	0	0	01 Jan 2020 - 00:08
2	947	Black Coffee & David Guetta feat. Delilah Montagu	Drive	11,000	180	01 Jan 2020 - 00:35
3	Eldos FM	Black Coffee feat. Soulstar	You Rock My World	0	0	01 Jan 2020 - 00:39
4	YFM 99.2	Black Coffee feat. Zhao	Any Other Way	29,000	0	01 Jan 2020 - 00:50

Pre-processing of data

In [12]:

```
# Remove leading white spce
for x in data.columns:
    data.rename(columns={x:x.strip()}, inplace=True)

#rename columns to make it easier to read
data.rename(columns={"Imp's": 'Impacts', 'Date/time': 'Date'}, inplace=True)

# remove all channels where impacts = 0 as they are assumed not to have official RAI
data = data[data.Impacts != 0]

#convert date column to a date object to enable Time Series analysis
data['Date']= pd.to_datetime(data['Date'])

#Sort columns by date
data.sort_values(by='Date', inplace=True)

#Convert Impacts into numeric format
data['Impacts'] = data['Impacts'].str.replace(',','')
data['Impacts'] = data['Impacts'].str.extract('(\d+)', expand=False) # remove all no
data['Impacts'] = data['Impacts'].map(lambda x: pd.to_numeric(x)) #convert to intege

# Fill null values and inf values with 0
data['Impacts'].fillna(0, inplace = True)
data.replace([np.inf, -np.inf], 0, inplace=True)

#convert 'Seconds' and 'Impacts' columns to integer types
data["Sec"] = pd.to_numeric(data["Sec"])
data['Impacts'] = data['Impacts'].astype('int64')

#Get rid of rows that dont make sense ie where there are no second of play but have
#this could be because of incorrect data capture or under 1 second of play is record
data = data[-((data["Impacts"] > 0) & (data["Sec"] == 0))]
```

Exploratory code

In [13]:

```
#count number of unique Radio Stations tracked
data.Station.nunique()
```

Out[13]:

```
43
```

In [14]:

```
# Check overview of dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8970 entries, 2 to 22716
Data columns (total 6 columns):
 #   Column   Non-Null Count  Dtype         
---  ------   --------------  -----         
 0   Station  8970 non-null   object        
 1   Artist   8970 non-null   object        
 2   Title    8970 non-null   object        
 3   Impacts  8970 non-null   int64         
 4   Sec      8970 non-null   int64         
 5   Date     8970 non-null   datetime64[ns]
dtypes: datetime64[ns](1), int64(2), object(3)
memory usage: 490.5+ KB
```

In [15]:

```
#Get basic summary statistics of dataset
data.describe()
```

Out[15]:

	Impacts	Sec
count	8970.000000	8970.000000
mean	138217.505463	249.258194
std	157437.301387	93.966489
min	0.000000	0.000000
25%	20000.000000	180.000000
50%	79333.000000	240.000000
75%	203000.000000	312.000000
max	998000.000000	972.000000

In [16]:

```
# Check date ranges of data, since it is one full year we will not need to subset it
print (data.Date.min())
print (data.Date.max())
```

```
2020-01-01 00:35:00
2020-12-31 23:08:00
```

In [17]:

```
original_data = data.copy() #make a copy of the pre-processed data as a backup
data.tail()
```

Out[17]:

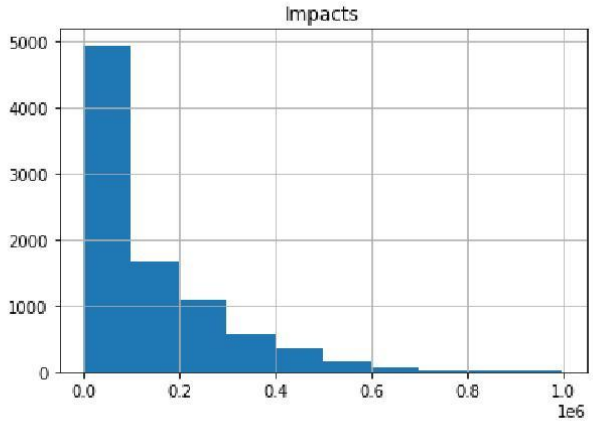
	Station	Artist	Title	Impacts	Sec	Date
22705	Munghana Lonene FM	Black Coffee feat. Sabrina Claudio	Sbcnscsly	584000	432	2020-12-31 19:44:00
22706	Ligwalagwala FM	Black Coffee feat. Sabrina Claudio	Sbcnscsly	348000	354	2020-12-31 20:16:00
22708	Jacaranda	Black Coffee feat. Usher	LaLaLa	233000	204	2020-12-31 20:31:00
22714	KFM	Black Coffee feat. Nakhane Toure	We Dance Again	212000	324	2020-12-31 22:09:00
22716	SAfm	Black Coffee feat. Nakhane Toure	We Dance Again	13000	330	2020-12-31 23:08:00

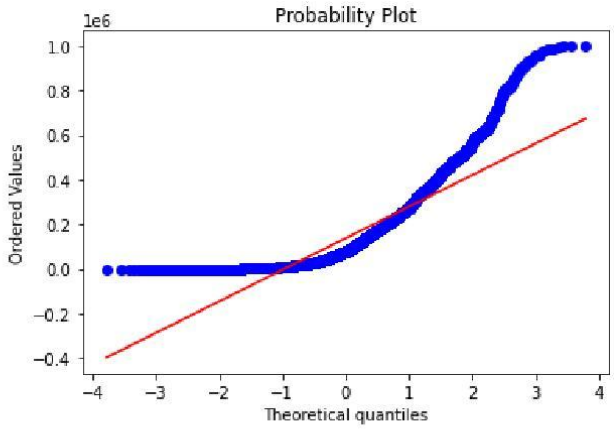
Check for a normal distribution

In [18]:

```
# Check for normality of dataframe for 'Impacts'

# Mean () and Median () not close together
dt = data[['Impacts']]
print(f'Kurtosis: {dt.kurt()}')
print(f'Skewness: {dt.skew()}')

# Do a Shapiro Wilkes test
stat, p = shapiro(dt.iloc[:,:].values)
print(f'Shapiro-Wilkes: Statistic: {stat}, p-value = {p}')

data.hist('Impacts')
plt.show()

#Create a QQ plot to check for normal distribution
stats.probplot(data['Impacts'], dist="norm", plot=pylab)
pylab.show()
```

```
Kurtosis: Impacts    3.768057
dtype: float64
Skewness: Impacts    1.763089
dtype: float64
Shapiro-Wilkes: Statistic: 0.808011531829834, p-value = 0.0
```

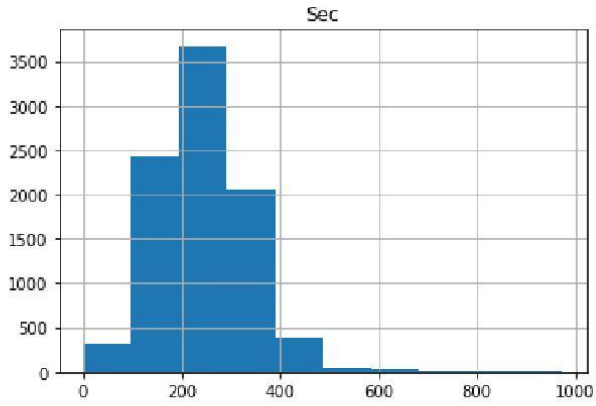


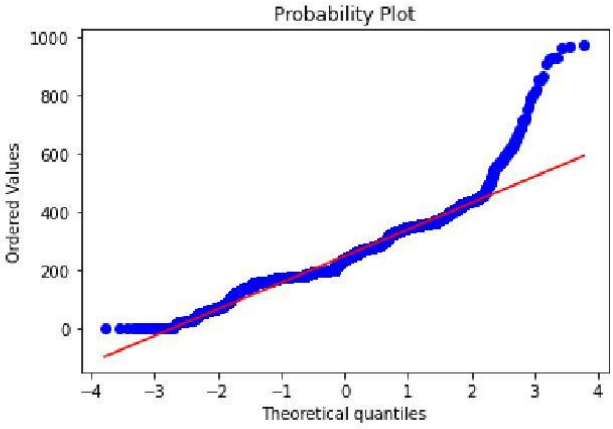


In [19]:

```
# Check for normality of dataframe for 'seconds'
# Mean (249) and Median (240) are close together
# Data looks like it follows a normal distribution
d = data[['Sec']]
print(f'Kurtosis: {d.kurt()}')
print(f'Skewness: {d.skew()}')

# Do a Shapiro Wilkes test
statd, pd = shapiro(d)
print(f'Shapiro-Wilkes: Statistic: {statd}, p-value = {pd}')

data.hist('Sec')
plt.show()

#Create a QQ plot to check for normal distribution
stats.probplot(data['Sec'], dist="norm", plot=pylab)
pylab.show()
```

```
Kurtosis: Sec    4.473664
dtype: float64
Skewness: Sec    1.057147
dtype: float64
Shapiro-Wilkes: Statistic: 0.935672402381897, p-value = 0.0
```





It seems that 'Impacts' does not follow a normal distribution and is skewed right. 'Secs' is very slightly skewed

right but resembles a normal distribution. The Quantile Quantile plot (QQ) of the 'Secs' variable show that it is close to a normal distribution. I will assume as such.

Use Bartlett's test to test for change in variance before and after lockdown and album launch

In [20]:

```
# Split data into two datasets, to check seconds played data before and after the re
# Release date: 1 May 2020

PA = data[['Sec', 'Date']]
PreAlbum = PA[PA['Date'] < '2020-05-01']

PostAlbum = PA[PA['Date'] >= '2020-05-01']

#Remove Date component to check for variance of Seconds played values within timefra

PreAlbumSec = PreAlbum['Sec']
PostAlbumSec = PostAlbum['Sec']
```

In [21]:

```
PreAlbumSec.describe()
```

Out[21]:

```
count    2118.000000
mean      224.059490
std        87.013311
min         0.000000
25%       180.000000
50%       198.000000
75%       264.000000
max       930.000000
Name: Sec, dtype: float64
```

In [22]:

```
PostAlbumSec.describe()
```

Out[22]:

```
count    6852.000000
mean      257.047285
std        94.672360
min         0.000000
25%       180.000000
50%       252.000000
75%       330.000000
max       972.000000
Name: Sec, dtype: float64
```

In [23]:

```
# Do Bartlett's test to check significance of variance between two groups
stats.bartlett(PreAlbumSec, PostAlbumSec)
```

Out[23]:

```
BartlettResult(statistic=22.328772996158293, pvalue=2.297384464247441e
-06)
```

Since p value is less than 0.05 there is evidence that the the change in variance is significant after the release of his album release in 2020

In [24]:

```
#bartletts test for equal variance before and after the first lockdown date that sta

PreLockdown = PA[PA['Date'] < '2020-03-26']

PostLockdown = PA[PA['Date'] >= '2020-03-26']
PostAlbum.head()

#Remove Date component to check for variance of Seconds played values within timefra

PreLockdownSec = PreLockdown['Sec']
PostLockdownSec = PostLockdown['Sec']
# Do Bartlett's test to check significance of variance between two groups
stats.bartlett(PreLockdownSec, PostLockdownSec)
```

Out[24]:

```
BartlettResult(statistic=66.36816061913828, pvalue=3.740962870588646e-
16)
```

Since p value is less than 0.05 there is evidence that the the change in variance is significant after the lockdown started.

T-Test for change in means after lockdown

Do a t-test to test for significance of means before and after the lockdown started on Black Coffee's seconds of play

H_0: Lockdown had no effect on Black Coffee's spins after lockdown was announced

H_1: Lockdown had an effect on Black Coffee's spins after lockdown was announced

In [25]:

```
from scipy import stats

t_value,p_value=stats.ttest_ind(PreLockdownSec,PostLockdownSec)

print('Test statistic is %f'%float("{:.6f}".format(t_value)))

print('p-value for two tailed test is %f'%p_value)

alpha = 0.05
```

```
Test statistic is -13.814048
p-value for two tailed test is 0.000000
```

The p value is less than 0.05 therefore we reject the null hypothesis. Therefore lockdown Black Coffee's spins changed significantly after the lockdown date. This will need to be analysed as its possible that spins did not change due to lockdown, there could have been other factors.

In [26]:

```
# Export data to CSV for use in Tableau
data.to_csv('BlackCoffee_Cleaned_dataset.csv')
```

Time series analysis of Secs

In [27]:

```
# create a new data set to do a basic TS analysis for Seconds

df = data[['Sec', 'Date']]
df.head()
```

Out[27]:

	Sec	Date
2	180	2020-01-01 00:35:00
5	216	2020-01-01 00:56:00
6	198	2020-01-01 00:57:00
12	198	2020-01-01 04:39:00
15	66	2020-01-01 05:08:00

In [28]:

```
#use the time variable as the new index to the dataframe and drop it from the main a

df.set_index('Date', inplace = True)

df
```

Out[28]:

Date	Sec
2020-01-01 00:35:00	180
2020-01-01 00:56:00	216
2020-01-01 00:57:00	198
2020-01-01 04:39:00	198
2020-01-01 05:08:00	66
...	...
2020-12-31 19:44:00	432
2020-12-31 20:16:00	354
2020-12-31 20:31:00	204
2020-12-31 22:09:00	324
2020-12-31 23:08:00	330

8970 rows × 1 columns

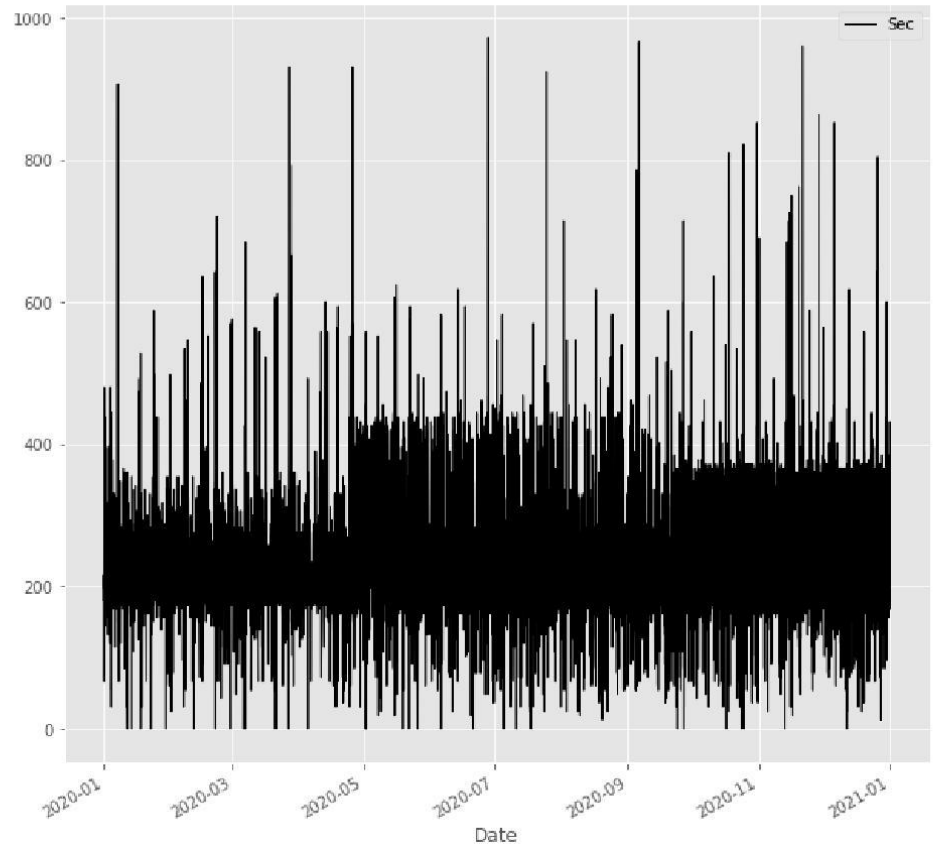
In [29]:

```
# Check dataframe info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 8970 entries, 2020-01-01 00:35:00 to 2020-12-31 23:08:0
0
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Sec     8970 non-null   int64
dtypes: int64(1)
memory usage: 140.2 KB
```

In [30]:

```
#plot the series
plt.style.use('ggplot')
plt.rcParams['figure.figsize']=(10,10)
df.plot(linewidth=1, color='black')
plt.show()
```

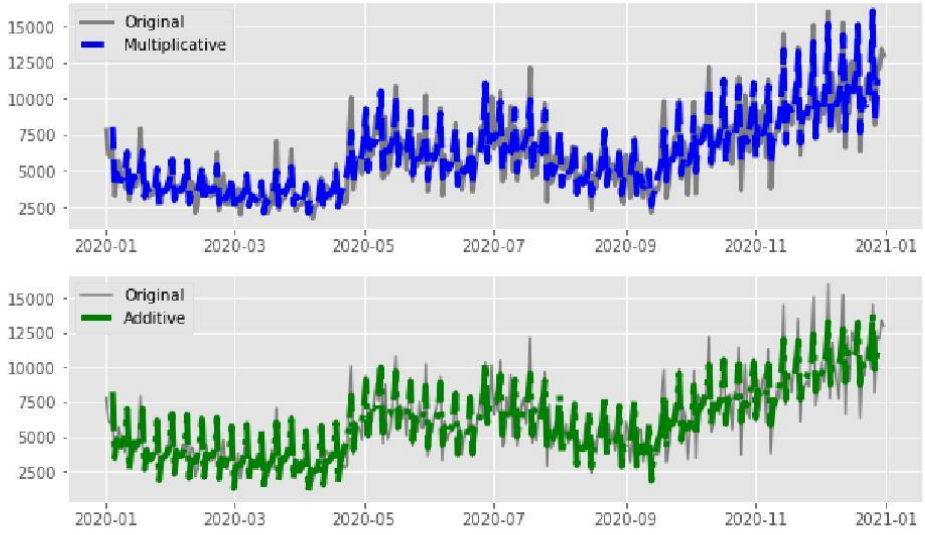


In [32]:

```
# produce a multiplicative and additive decomposition of the series

sdc_a = seasonal_decompose(dfsum, model='a') # this is the model-parameter's default
sdc_m = seasonal_decompose(dfsum, model='m')

additive_model = sdc_a.trend+sdc_a.seasonal # deterministic part of this decompositi
multiplicative_model = sdc_m.trend*sdc_m.seasonal
```

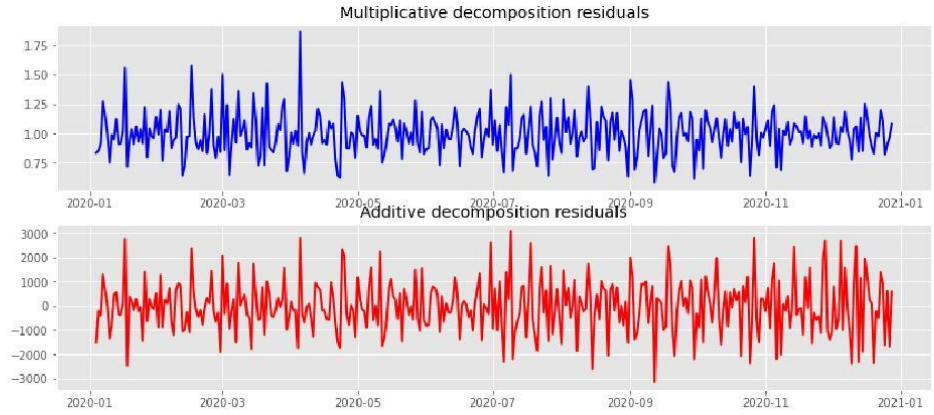
In [33]:

```
#Plot the different seasonal models

plt.rcParams['figure.figsize']=(10,6)
plt.style.use('ggplot')
plt.figure(1)
plt.subplot(2,1,1)
plt.plot(dfsum, label = 'Original', color='grey',linewidth=3)
plt.plot(multiplicative_model, label = 'Multiplicative',linestyle='--',color='blue',
plt.legend(loc = 'best')
plt.subplot(2,1,2)
plt.plot(dfsum, label = 'Original',color='grey')
plt.plot(additive_model, label = 'Additive',linestyle='-.', color='green', linewidth
plt.legend(loc = 'best')
plt.show()
```



In [34]:

```
#plot the two decomposition residuals
plt.style.use('ggplot')
plt.rcParams['figure.figsize']=(14,6)
x = dfsum.index
residual_m = sdc_m.resid
plt.subplot(2,1,1)
plt.title('Multiplicative decomposition residuals')
plt.plot(x, residual_m, 'b', linewidth=2)
residual_a = sdc_a.resid
plt.subplot(2,1,2)
plt.title('Additive decomposition residuals')
plt.plot(x, residual_a, 'r', linewidth=2)
plt.show()
```



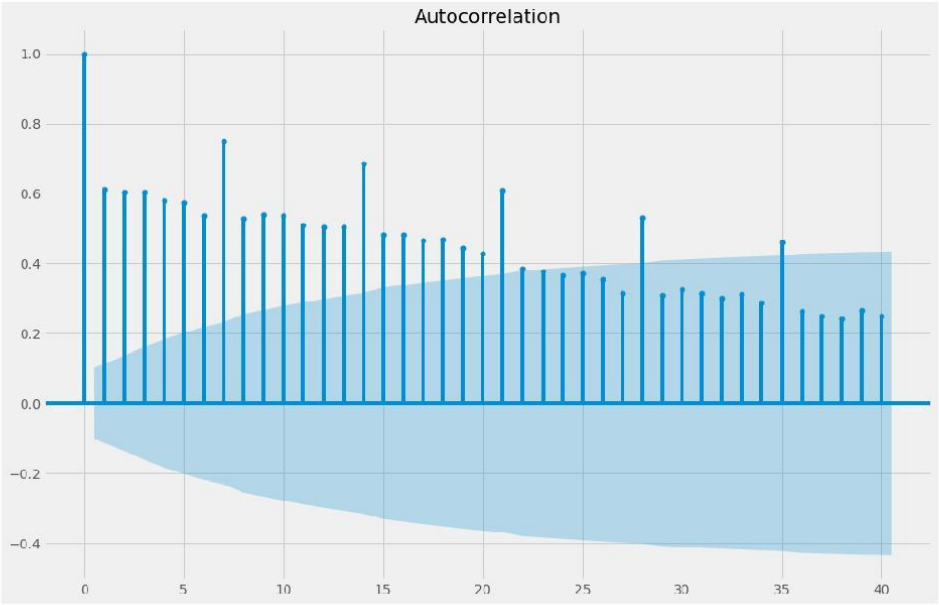
In [35]:

```
# use the multiplicative model as variance is smaller
# Take the log of the multiplicative model
df_log = np.log(dfsum)
```

Acf Plot

In [36]:

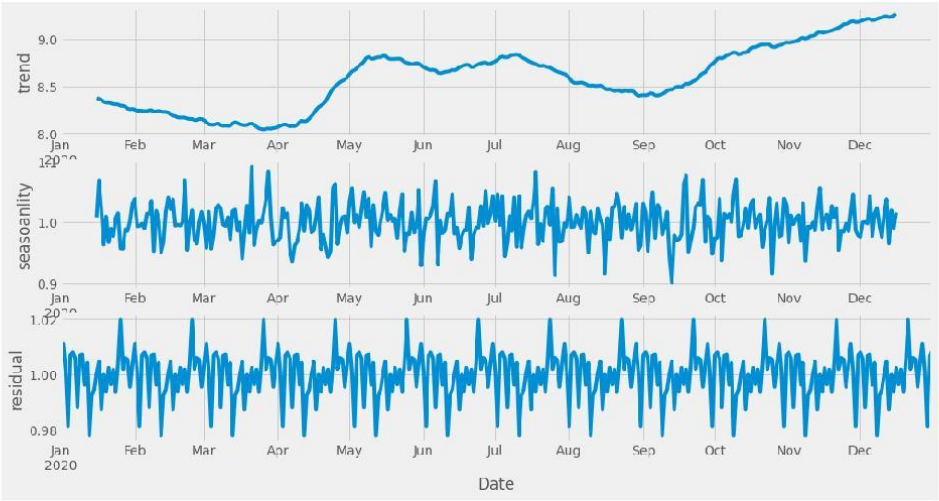
```
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize']=(15,10)
plot_acf(df_log,lags=40)
plt.show()
```



There is clearly a seasonal component every Saturday. This could mean that stations are playing Black Coffee's tracks for a longer time period on saturday's and more regularly.

In [37]:

```
# Decompose series into its trend and seasonal parts

res = seasonal_decompose(df_log, model = "multiplicative",period = 30)

fig, (ax1,ax2,ax3) = plt.subplots(3,1, figsize=(15,8))
res.trend.plot(ax=ax1,ylabel = "trend")
res.resid.plot(ax=ax2,ylabel = "seasoanlity")
res.seasonal.plot(ax=ax3,ylabel = "residual")
plt.show()
```



There seems to be a clear upward trend and some seasonality, this can be investigated further if needed.

Outlier detection

In [112]:

```
# Create a datafram to check for outliers
df2 = data[['Sec', 'Date']]
df2.rename(columns={ "Date": "ds", "Sec": "y"}, inplace=True)
df2.set_index('ds', inplace = True)
#
#df2.resample('D').sum()
#df2['ds']= pd.to_datetime(df2['ds'])
df2sum = df2.resample('D').sum()
df2sum.reset_index(inplace=True)
df2sum
```

Out[112]:

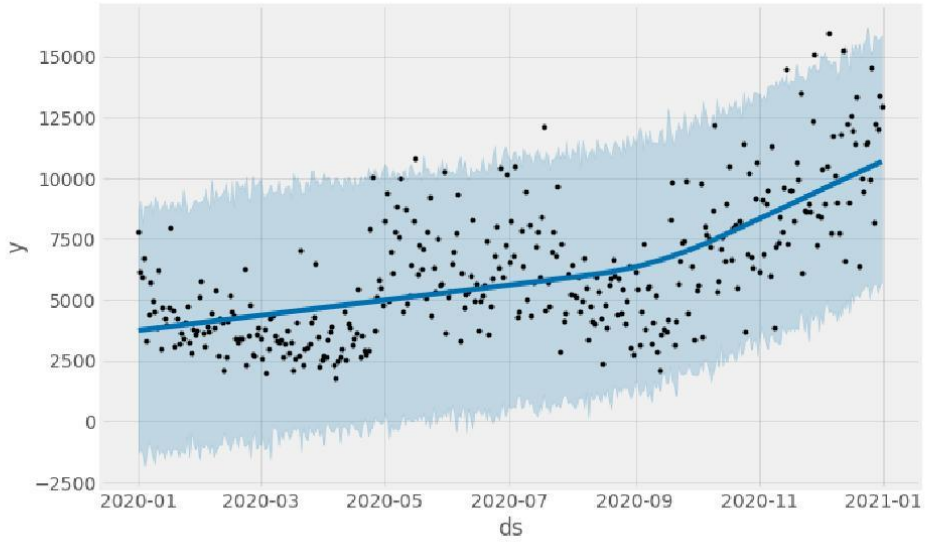
	ds	y
0	2020-01-01	7818
1	2020-01-02	6156
2	2020-01-03	5958
3	2020-01-04	6714
4	2020-01-05	3294
...	...	...
361	2020-12-27	8190
362	2020-12-28	12228
363	2020-12-29	12042
364	2020-12-30	13410
365	2020-12-31	12978

366 rows × 2 columns

In [119]:

```
#create a plot to show outliers above the 99 percent confidence interval for predict
from fbprophet import Prophet
import altair as alt
def fit_predict_model(dataframe, interval_width = 0.99, changepoint_range = 0.8):
    m = Prophet(daily_seasonality = False, yearly_seasonality = False, weekly_seasor
                seasonality_mode = 'multiplicative',
                interval_width = interval_width,
                changepoint_range = changepoint_range)
    m = m.fit(dataframe)

    forecast = m.predict(dataframe)
    forecast['fact'] = dataframe['y'].reset_index(drop = True)
    print('Displaying Prophet plot')
    fig1 = m.plot(forecast)
    return forecast
# Display outputs
pred = fit_predict_model(df2sum)
```

```
Initial log joint probability = -18.6621
    Iter      log prob        ||dx||      ||grad||       alpha      al
pha0  # evals  Notes
      69       559.264    0.00339809       80.1847    3.42e-05
0.001      117  LS failed, Hessian reset
      99       559.598   0.000509851        68.016       0.616
0.616      154
    Iter      log prob        ||dx||      ||grad||       alpha      al
pha0  # evals  Notes
     199       559.676    1.0711e-06       71.3944      0.5954      0.
5954      280
    Iter      log prob        ||dx||      ||grad||       alpha      al
pha0  # evals  Notes
     208       559.678   7.81549e-06       72.4875   9.793e-08
0.001      333  LS failed, Hessian reset
     293        559.68   2.03671e-09        78.184    0.001314
1      459
Optimization terminated normally:
  Convergence detected: absolute parameter change was below tolerance
Displaying Prophet plot
```



In [120]:

```
def detect_anomalies(forecast):
    forecasted = forecast[['ds','trend', 'yhat', 'yhat_lower', 'yhat_upper', 'fact']
    #forecast['fact'] = df['y']

    forecasted['anomaly'] = 0
    forecasted.loc[forecasted['fact'] > forecasted['yhat_upper'], 'anomaly'] = 1
    forecasted.loc[forecasted['fact'] < forecasted['yhat_lower'], 'anomaly'] = -1

    #anomaly importances
    forecasted['importance'] = 0
    forecasted.loc[forecasted['anomaly'] ==1, 'importance'] = \
        (forecasted['fact'] - forecasted['yhat_upper'])/forecast['fact']
    forecasted.loc[forecasted['anomaly'] ==-1, 'importance'] = \
        (forecasted['yhat_lower'] - forecasted['fact'])/forecast['fact']

    return forecasted

pred = detect_anomalies(pred)
```

In [118]:

```
#create a list of anomalies that lie outside the 99 percent confidence interval
anoms = pred[pred.anomaly == 1]
anoms
```

Out[118]:

	ds	trend	yhat	yhat_lower	yhat_upper	fact	anomaly	importance
129	2020-05-09	5062.055836	5062.055836	-150.786232	9701.056949	9990	1	0.028923
136	2020-05-16	5133.561341	5133.561341	512.659916	10505.086482	10830	1	0.030001
199	2020-07-18	5777.110851	5777.110851	771.365815	11065.441282	12138	1	0.088364
318	2020-11-14	8841.654228	8841.654228	3224.100983	14414.595445	14478	1	0.004379
332	2020-11-28	9389.481523	9389.481523	4085.983047	14380.998292	15090	1	0.046985
339	2020-12-05	9663.395170	9663.395170	4436.628083	14913.172874	15978	1	0.066643
346	2020-12-12	9937.308817	9937.308817	5282.605199	15082.600278	15234	1	0.009938

In [155]:

```
#Check for which day of the week the outliers sit: 0 = Monday, 1 = Tues etc
s = anoms.ds
s.dt.dayofweek
```

Out[155]:

```
129    5
136    5
199    5
318    5
332    5
339    5
346    5
Name: ds, dtype: int64
```

All outliers occur on a Friday, this could be because of there is more and longer play of Black Coffee's music on the start of the weekend, or there could be more party related radio shows on that day