DATA CLEANSING A ROCK SONG DATASET

WITH PYTHON

DR. ALVIN ANG

CONTENTS

I. Step 1: Reading in the Data 3

A. Import All Libraries 3

B. Load the CSV 4

1. Option 1: Rename the Column 4
2. Option 2: Rename the Column Another Way 5

II. Step 2: Check for NaNs 6

A. df.info() 6

B. Total Up All NaNs 7

C. Display all the Release NaN Rows 8

D. Replace all NaNs with Zeros 9

1. Option 1 9
2. Option 2: Alternative Way to Replace NaN with 0s 10

E. Check NaNs in Release Column 11

III. Step 3: Changing from String to Integer Type 12

A. df.info() 12

B. Attempting to Change Release Column from String to Integer 13

C. Check All Unique Values in Release Column 14

D. Change SONGFACTS to 1972 15

E. Retrying Conversion 16

IV. Step 4: Adjusting the Minimum Release Year 17

A. df.describe() 17

B. Adjusting the Min Release Year 18

C. Imposing a Cut Off for Release Year 19

V. Step 5: Creating a Function to Validate if Release Year < 1970 20

A. Create the Function 20

B. Apply The Function 20

About Dr. Alvin Ang 21

I. STEP 1: READING IN THE DATA

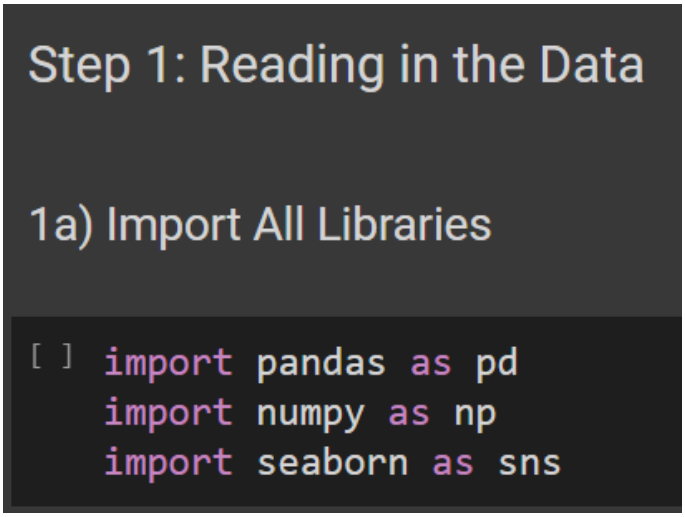
IPYNB:

- https://www.alvinang.sg/s/Data_Cleansing_a_Rock_Song_Dataset_with_Python_by_Dr_Alvin_Ang.ipynb

File::

- https://www.alvinang.sg/s/rock.csv

A. IMPORT ALL LIBRARIES



B. LOAD THE CSV

1. OPTION 1: RENAME THE COLUMN

1b) Load the CSV

1b(i) Option 1: Rename the Column

```
[ ] column_names = ['song',
                    'artist',
                    'release',
                    'song_artist',
                    'first',
                    'year',
                    'playcount',
                    'fg',]

    df = pd.read_csv('https://www.alvinang.sg/s/rock.csv',
                     names=column_names,
                     skiprows=1)
```

	A	B	C	D	E	F	G	H
1	Song Clean	ARTIST CLEAN	Release	COMBINED	First?	Year?	PlayCount	F*G
2	Caught Up in You	.38 Special	1982	Caught Up in You by .38 Special	1	1	82	82
3	Fantasy Girl	.38 Special		Fantasy Girl by .38 Special	1	0	3	0
4	Hold On Loosely	.38 Special	1981	Hold On Loosely by .38 Special	1	1	85	85
5	Rockin' Into the Night	.38 Special	1980	Rockin' Into the Night by .38 Special	1	1	18	18
6	Art For Arts Sake	10cc	1975	Art For Arts Sake by 10cc	1	1	1	1
7	Kryptonite	3 Doors Down	2000	Kryptonite by 3 Doors Down	1	1	13	13
8	Loser	3 Doors Down	2000	Loser by 3 Doors Down	1	1	1	1
9	When I'm Gone	3 Doors Down	2002	When I'm Gone by 3 Doors Down	1	1	6	6
10	What's Up?	4 Non Blondes	1992	What's Up? by 4 Non Blondes	1	1	3	3
11	Take On Me	a-ha	1985	Take On Me by a-ha	1	1	1	1
12	Baby, Please Don't Go	AC/DC		Baby, Please Don't Go by AC/DC	1	0	1	0
13	Back In Black	AC/DC	1980	Back In Black by AC/DC	1	1	97	97

```
df.head()

#load the dataset and rename the columns
```

	song	artist	release	song_artist	first	year	playcount	fg
0	Caught Up in You	.38 Special	1982	Caught Up in You by .38 Special	1	1	82	82
1	Fantasy Girl	.38 Special	NaN	Fantasy Girl by .38 Special	1	0	3	0
2	Hold On Loosely	.38 Special	1981	Hold On Loosely by .38 Special	1	1	85	85
3	Rockin' Into the Night	.38 Special	1980	Rockin' Into the Night by .38 Special	1	1	18	18
4	Art For Arts Sake	10cc	1975	Art For Arts Sake by 10cc	1	1	1	1

2. OPTION 2: RENAME THE COLUMN ANOTHER WAY

1b)(ii) Option 2: Rename the Column

```
df = pd.read_csv('https://www.alvinang.sg/s/rock.csv')

rename_map = {
#Original column: renamed column
    'Song Clean':    'song',
    'ARTIST CLEAN':  'artist',
    'Release Year':  'release',
    'COMBINED':      'song_artist',
    'First?':        'first',
    'Year?':         'year',
    'PlayCount':     'playcount',
    'F*G':           'fg'
}

df.rename(columns=rename_map, inplace=True)
df.head()
```

II. STEP 2: CHECK FOR NANS

A. DF.INFO()

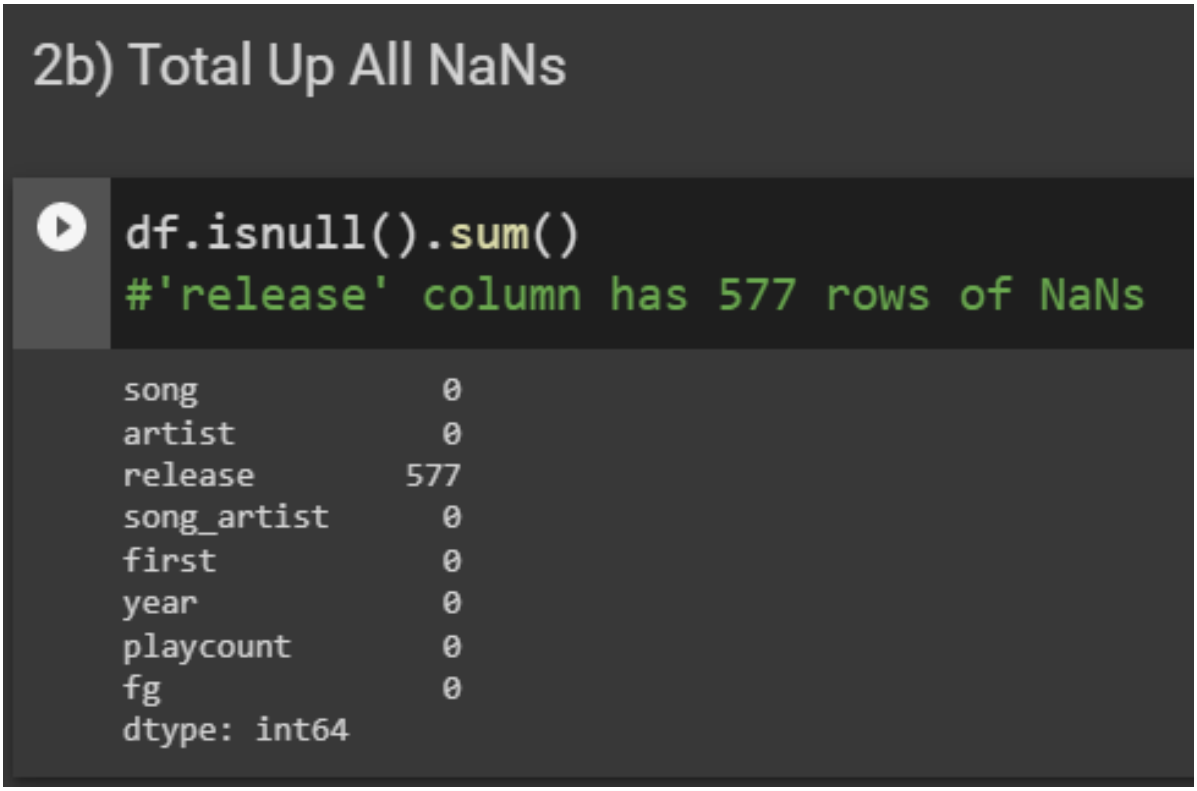
Step 2: Check for NaNs

2a) df.info()

```
df.info()
#'release' column has MANY NaNs --> due to 1653 non-null rows
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2230 entries, 0 to 2229
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   song         2230 non-null   object
 1   artist       2230 non-null   object
 2   release      1653 non-null   object
 3   song_artist  2230 non-null   object
 4   first        2230 non-null   int64 
 5   year         2230 non-null   int64 
 6   playcount    2230 non-null   int64 
 7   fg           2230 non-null   int64 
dtypes: int64(4), object(4)
memory usage: 139.5+ KB
```

B. TOTAL UP ALL NANS



C. DISPLAY ALL THE RELEASE NAN ROWS

2c) Display all the Release NaN Rows

```
[ ] df[df['release'].isnull()]
```

	song	artist	release	song_artist	first	year	playcount	fg
1	Fantasy Girl	.38 Special	NaN	Fantasy Girl by .38 Special	1	0	3	0
10	Baby, Please Don't Go	AC/DC	NaN	Baby, Please Don't Go by AC/DC	1	0	1	0
13	CAN'T STOP ROCK'N'ROLL	AC/DC	NaN	CAN'T STOP ROCK'N'ROLL by AC/DC	1	0	5	0
16	Girls Got Rhythm	AC/DC	NaN	Girls Got Rhythm by AC/DC	1	0	24	0
24	Let's Get It Up	AC/DC	NaN	Let's Get It Up by AC/DC	1	0	4	0
...	...	...	...	...	...	...	...	...
2216	I'm Bad, I'm Nationwide	ZZ Top	NaN	I'm Bad, I'm Nationwide by ZZ Top	1	0	10	0
2218	Just Got Paid	ZZ Top	NaN	Just Got Paid by ZZ Top	1	0	2	0
2221	My Head's In Mississippi	ZZ Top	NaN	My Head's In Mississippi by ZZ Top	1	0	1	0
2222	Party On The Patio	ZZ Top	NaN	Party On The Patio by ZZ Top	1	0	14	0
2225	She Loves My Automobile	ZZ Top	NaN	She Loves My Automobile by ZZ Top	1	0	1	0

577 rows × 8 columns

D. REPLACE ALL NANS WITH ZEROS

1. OPTION 1

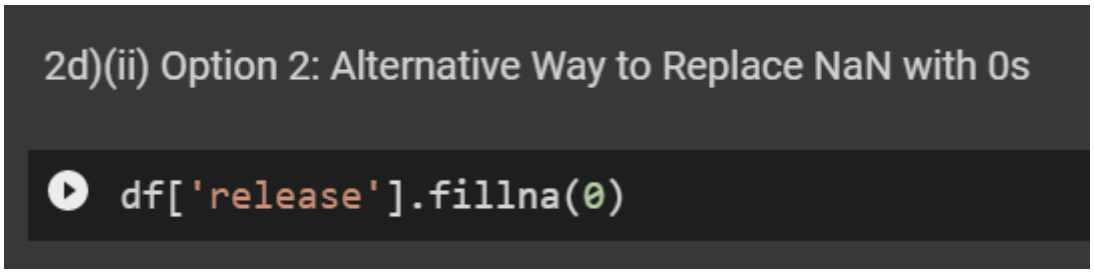
2d) Replace all NaNs with Zeros

2d(i) Option 1

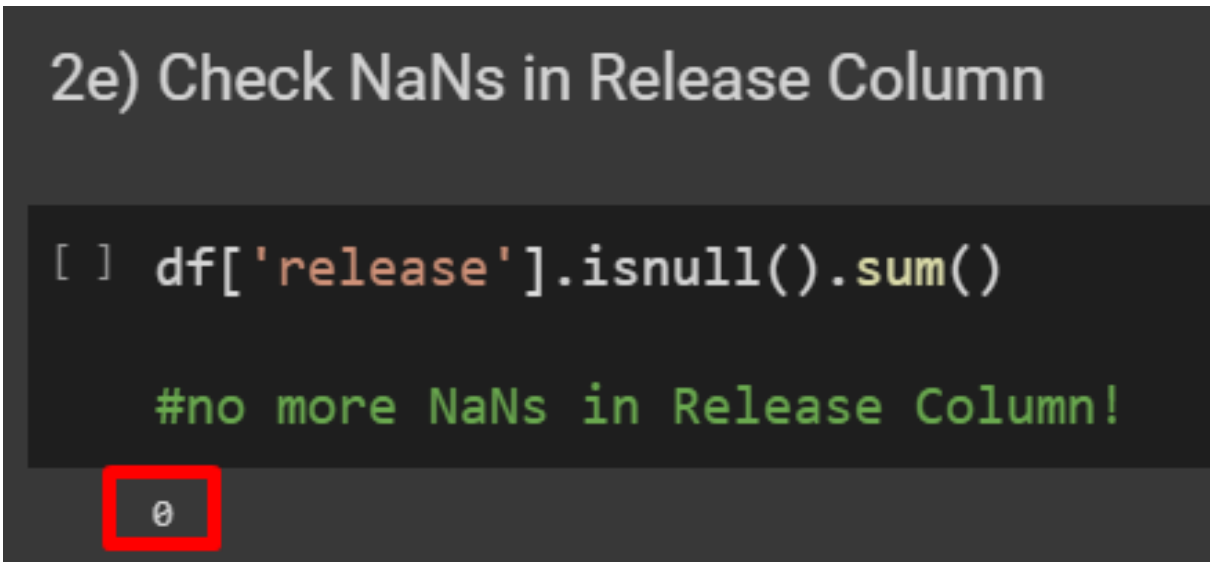
```
[ ] df.loc[df['release'].isnull(), 'release'] = 0
    df.head()
```

	song	artist	release	song_artist	first	year	playcount	fg
0	Caught Up in You	.38 Special	1982	Caught Up in You by .38 Special	1	1	82	82
1	Fantasy Girl	.38 Special	0	Fantasy Girl by .38 Special	1	0	3	0
2	Hold On Loosely	.38 Special	1981	Hold On Loosely by .38 Special	1	1	85	85
3	Rockin' Into the Night	.38 Special	1980	Rockin' Into the Night by .38 Special	1	1	18	18
4	Art For Arts Sake	10cc	1975	Art For Arts Sake by 10cc	1	1	1	1

2. OPTION 2: ALTERNATIVE WAY TO REPLACE NAN WITH 0S



E. CHECK NANS IN RELEASE COLUMN



III. STEP 3: CHANGING FROM STRING TO INTEGER TYPE

A. DF.INFO()

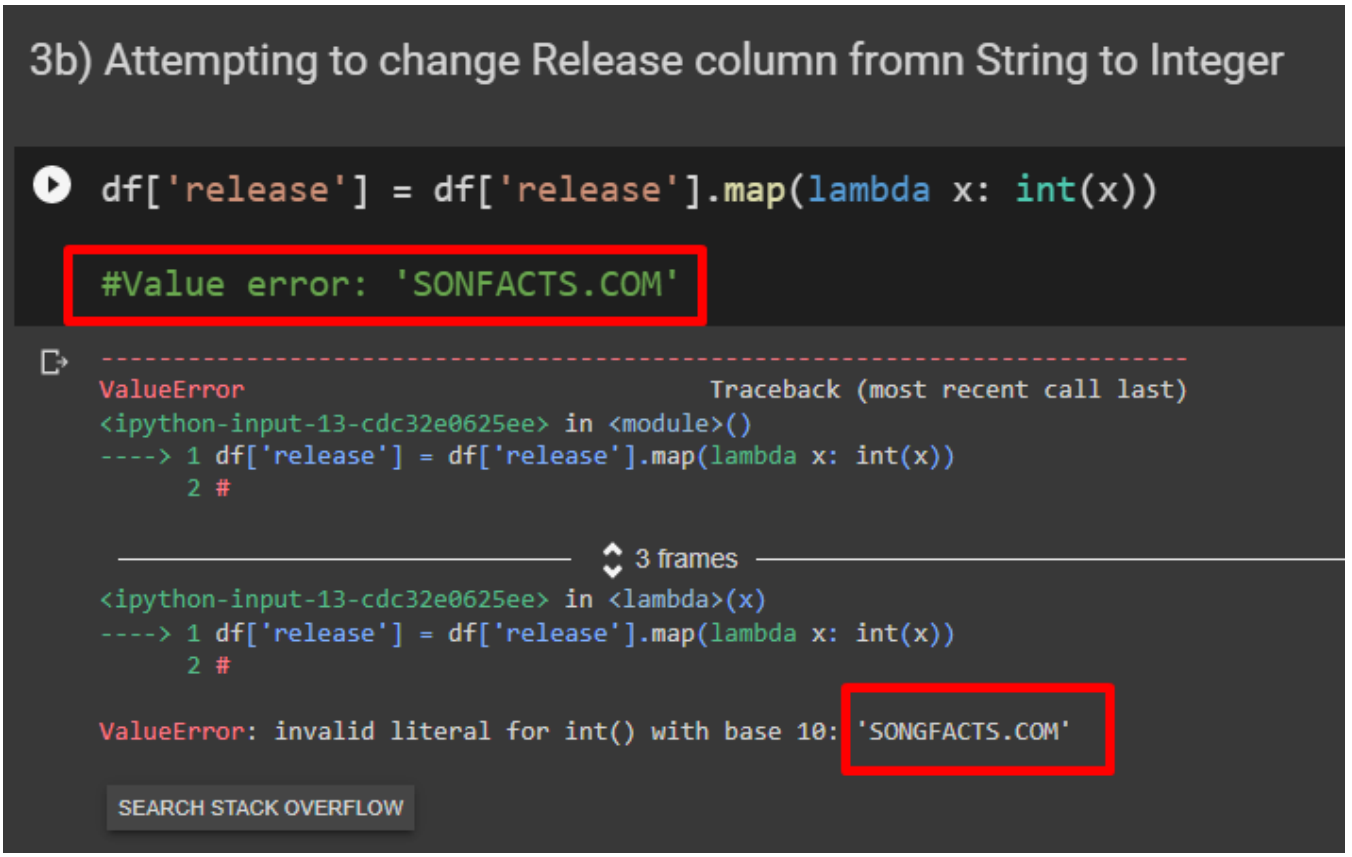
Step 3: Changing from String to Integer Type

3a) df.info()

```
df.info()
#we want to change the Release column to
#integer
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2230 entries, 0 to 2229
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   song         2230 non-null   object
 1   artist       2230 non-null   object
 2   release      2230 non-null   object
 3   song_artist  2230 non-null   object
 4   first        2230 non-null   int64
 5   year         2230 non-null   int64
 6   playcount    2230 non-null   int64
 7   fg           2230 non-null   int64
dtypes: int64(4), object(4)
memory usage: 139.5+ KB
```

B. ATTEMPTING TO CHANGE RELEASE COLUMN FROM STRING TO INTEGER



C. CHECK ALL UNIQUE VALUES IN RELEASE COLUMN

3c) Check All Unique Values in Release Column

```
[ ] df.release.unique()

    #there is Dirty Data 'SONGFACTS.COM'
```

```
array(['1982', 0, '1981', '1980', '1975', '2000', '2002', '1992', '1985',
       '1993', '1976', '1995', '1979', '1984', '1977', '1990', '1986',
       '1974', '2014', '1987', '1973', '2001', '1989', '1997', '1971',
       '1972', '1994', '1970', '1966', '1965', '1983', '1955', '1978',
       '1969', '1999', '1968', '1988', '1962', '2007', '1967', '1958',
       '1071', '1996', '1991', '2005', '2011', '2004', '2012', '2003',
       '1998', '2008', '1964', '2013', '2006', 'SONGFACTS.COM', '1963',
       '1961'], dtype=object)
```

D. CHANGE SONGFACTS TO 1972

3d) Change SONGFACTS to 1972

```
[ ] df[df['release'] == "SONGFACTS.COM"]
```

	song	artist	release	song_artist	first	year	playcount	fg
1504	Bullfrog Blues	Rory Gallagher	SONGFACTS.COM	Bullfrog Blues by Rory Gallagher	1	1	1	1

```
[ ] df.loc[
        df['release'] == "SONGFACTS.COM",
        'release']\
        =1972
```

```
[ ] df.iloc[1504:1505]
    #Row 1504 Release has been changed to 1972
```

	song	artist	release	song_artist	first	year	playcount	fg
1504	Bullfrog Blues	Rory Gallagher	1972	Bullfrog Blues by Rory Gallagher	1	1	1	1

3e) Retrying Conversion

```
[ ] df['release'] = df['release'].\
                    map(lambda x: int(x))
```

```
[ ] df.info()

    #Release has been changed to Integer!
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2230 entries, 0 to 2229
Data columns (total 8 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   song         2230 non-null   object
 1   artist       2230 non-null   object
 2   release      2230 non-null   int64
 3   song_artist  2230 non-null   object
 4   first        2230 non-null   int64
 5   year         2230 non-null   int64
 6   playcount    2230 non-null   int64
 7   fg           2230 non-null   int64
dtypes: int64(5), object(3)
memory usage: 139.5+ KB
```

A. DF.DESCRIBE()

Step 4: Adjusting the Minimum Release Year

4a) df.describe()

```
df['release'].describe()

#maximum year of release = 2014
#but Minimum year of release = 0?
#something is wrong???
```

```
count    2230.000000
mean     1466.215695
std       866.706564
min         0.000000
25%         0.000000
50%      1973.000000
75%      1981.000000
max      2014.000000
Name: release, dtype: float64
```

B. ADJUSTING THE MIN RELEASE YEAR

4b) Adjusting the Min Release Year

```
[ ] min_release_year = df[df['release']>0]

    #minimum release year allowed must be > 0
```

```
[ ] min_release_year['release'].describe()

    #note that the earliest release year = 1071
    #isn't it strange for songs to be releaseed 1071???
```

```
count    1653.000000
mean     1978.016334
std        24.184378
min      1071.000000
25%      1971.000000
50%      1977.000000
75%      1984.000000
max      2014.000000
Name: release, dtype: float64
```

C. IMPOSING A CUT OFF FOR RELEASE YEAR

4c) Imposing a Cut Off for Release Year

```
[ ] min_release_year[df.release == 1071]

    #row 547 has release year = 1071
    #obviously this is wrong, its corrupted and should be replaced with something else
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: Boolean Series key will be reindexed to m
  """Entry point for launching an IPython kernel.
```

	song	artist	release	song_artist	first	year	playcount	fg
547	Levon	Elton John	1071	Levon by Elton John	1	1	8	8

```
[ ] above_1071 = df[df['release']>1071]
```

```
above_1071.release.describe()

#now minimum year of release is 1955
```

```
count    1652.000000
mean     1978.565375
std         9.308364
min      1955.000000
25%      1971.000000
50%      1977.000000
75%      1984.000000
max      2014.000000
Name: release, dtype: float64
```

```
[ ] above_1071.iloc[400:403]

    #now Row 547 has been removed!
```

	song	artist	release	song_artist	first	year	playcount	fg
546	Honky Cat	Elton John	1972	Honky Cat by Elton John	1	1	2	2
548	Madman Across The Water	Elton John	1971	Madman Across The Water by Elton John	1	1	1	1
549	Rocket Man	Elton John	1972	Rocket Man by Elton John	1	1	68	68

V. STEP 5: CREATING A FUNCTION TO VALIDATE IF RELEASE YEAR < 1970

A. CREATE THE FUNCTION

Step 5: Creating a Function to Validate if Release Year < 1970

5a) Create the Function

```
def release_inspector(row):
    print('-------------------------------------------------------')
    print('Title:', row['song'],'|',
          'Artiste:', row['artist'], '|',
          'Release:', row['release'], '|',
          'Is it < 1970?:',
          row['release'] < 1970)
    return None
```

B. APPLY THE FUNCTION

5b) Apply the Function

```
df.sample(5).apply(release_inspector, axis = 1)
```

```
-------------------------------------------------------
Title: Those Shoes | Artiste: Eagles | Release: 1979 | Is it < 1970?: False
-------------------------------------------------------
Title: Atomic Punk | Artiste: Van Halen | Release: 0 | Is it < 1970?: True
-------------------------------------------------------
Title: Mista Bone | Artiste: Great White | Release: 0 | Is it < 1970?: True
-------------------------------------------------------
Title: You Better Run | Artiste: Pat Benatar | Release: 1980 | Is it < 1970?: False
-------------------------------------------------------
Title: Funk #49 | Artiste: Joe Walsh | Release: 2012 | Is it < 1970?: False
513     None
2117    None
730     None
1239    None
904     None
dtype: object
```

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.