

ANALYSIS ON NUCLEAR WEAPONS TESTS FROM 1945 TO 2019

INTRODUCTION

According to wikipedia Nuclear weapons tests are experiments carried out to determine nuclear weapons' effectiveness, yield, and explosive capability. Testing nuclear weapons offers practical information about how the weapons function, how detonations are affected by different conditions, and how personnel, structures, and equipment are affected when subjected to nuclear explosions. However, nuclear testing has often been used as an indicator of scientific and military strength. Many tests have been overtly political in their intention; most nuclear weapons states publicly declared their nuclear status through a nuclear test.

ABOUT THE DATA

This data was collected from <https://www.kaggle.com>. The data shows the year of each nuclear test by their country. The dataset is reliable, original and comprehensive. The source has their own licence over the dataset. Besides that, the dataset doesn't have any personal information. All the files have consistent columns and each column has the correct type of data. Finally, it would be good to have some updated information about the nuclear weapons tests.

```
In [1]: # Nuclear Weapons Tests Analysis with Pandas & Plotly
# We will be using the following Python Libraries:
# - Pandas
# - Matplotlib
# - Pandas Profiling Report
# - AutoViz
# - Plotly
```

```
In [2]: # We will cover the following chart types:
# - Histogram
# - Area Chart
# - Pie Chart
# - Bar Charts
```

CONTENTS

- [x] Libraries imports
- [x] Loading of dataset
- [x] Dataset exploration
- [x] Profiling report
- [x] AutoViz report
- [x] Data manipulation
- [x] Data visualization
- [x] Conclusion

IMPORT OUR LIBRARIES

```
In [3]: # Imports:
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
from pandas_profiling import ProfileReport
from autoviz.AutoViz_Class import AutoViz_Class
```



Imported v0.1.59. After importing, execute '%matplotlib inline' to display charts in Jupyter.
AV = AutoViz_Class()
dfte = AV.AutoViz(filename, sep=',', depVar='', dfte=None, header=0, verbose=1, lowess=False, chart_format='svg', max_rows_analyzed=150000, max_cols_analyzed=30, save_plot_dir=None)
Update: verbose=0 displays charts in your local Jupyter notebook.
verbose=1 additionally provides EDA data cleaning suggestions. It also displays charts.
verbose=2 does not display charts but saves them in AutoViz.Plots folder in local machine.
chart_format='boken' displays charts in your local Jupyter notebook.
chart_format='server' displays charts in your browser: one tab for each chart type
chart_format='html' silently saves interactive HTML files in your local machine

LOADING DATASET

```
In [4]: # Now we are going to load our dataset
df = pd.read_csv('nuclear.csv')
```

```
In [5]: df.head()
```

```
Out[5]:
```

| | country_name | year | nuclear_weapons_tests |
|---|--------------|------|-----------------------|
| 0 | China | 1945 | 0 |
| 1 | China | 1946 | 0 |
| 2 | China | 1947 | 0 |
| 3 | China | 1948 | 0 |
| 4 | China | 1949 | 0 |

DATASET EXPLORATION

In this part I will be exploring the dataset

```
In [6]: # Basic info about the DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 3 columns):
 #   Column              Non-Null Count  Dtype
---  --
 0   country_name         600 non-null    object
 1   year                 600 non-null    int64
 2   nuclear_weapons_tests 600 non-null    int64
dtypes: int64(2), object(1)
memory usage: 14.2+ KB
```

```
In [7]: # Describe method
df.describe()
```

```
Out[7]:
```

| | year | nuclear_weapons_tests |
|-------|-------------|-----------------------|
| count | 600.000000 | 600.000000 |
| mean | 1982.000000 | 3.431667 |
| std | 21.666774 | 9.806799 |
| min | 1945.000000 | 0.000000 |
| 25% | 1963.000000 | 0.000000 |
| 50% | 1982.000000 | 0.000000 |
| 75% | 2001.000000 | 1.000000 |
| max | 2019.000000 | 96.000000 |

```
In [8]: # Get a view of a unique value in column e.g. 'country_name'
df['country_name'].unique()
```

```
Out[8]: array(['China', 'France', 'India', 'North Korea', 'Pakistan', 'Russia', 'United Kingdom', 'United States'], dtype=object)
```

```
In [9]: # Check for null values
df.isnull()
```

```
Out[9]:
```

| | country_name | year | nuclear_weapons_tests |
|-----|--------------|-------|-----------------------|
| 0 | False | False | False |
| 1 | False | False | False |
| 2 | False | False | False |
| 3 | False | False | False |
| 4 | False | False | False |
| ... | ... | ... | ... |
| 595 | False | False | False |
| 596 | False | False | False |
| 597 | False | False | False |
| 598 | False | False | False |
| 599 | False | False | False |

600 rows x 3 columns

```
In [10]: # NoN count for each column
df.isnull().sum()
```

```
Out[10]:
```

| | |
|-----------------------|-------|
| country_name | 0 |
| year | 0 |
| nuclear_weapons_tests | 0 |
| dtype: | int64 |

PANDAS PROFILING REPORT

I will be using pandas-profiling to generate profile reports from my data [nuclear.csv]

```
In [11]: # AUTOMATED REPORTS
# Generate pandas profiling report
profile = ProfileReport(df, title = "nuclear.csv Profiling Report")
```

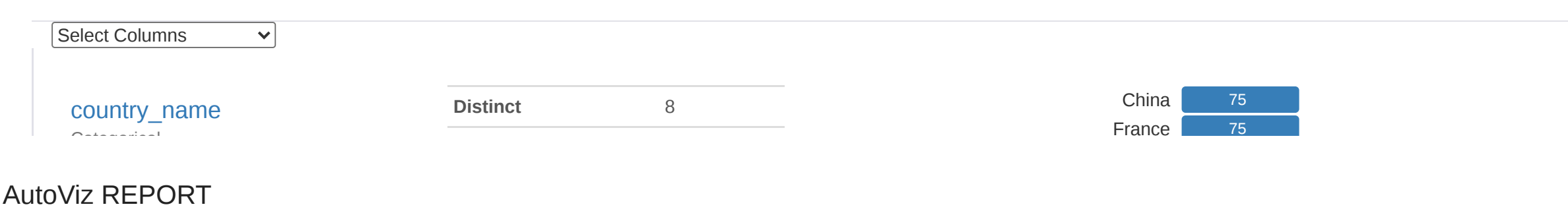
```
# To view in Notebook
profile.to_notebook_iframe()
```

```
Summarize dataset: 0% | 0/5 [00:00<7, ?it/s]
Generate report structure: 0% | 0/1 [00:00<7, ?it/s]
Render HTML: 0% | 0/1 [00:00<7, ?it/s]
```

nuclear.csv Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview



Variables



AutoViz REPORT

Using AutoViz to perform automatic visualization report on my dataset

```
In [12]: # AutoViz report
# Here we would show the AutoViz report
%matplotlib inline
plt.style.use('classic')
AV = AutoViz_Class()
df_autoviz = AV.AutoViz('nuclear.csv')
```

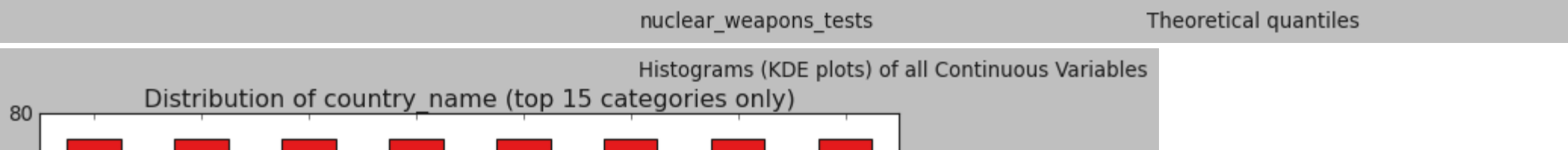
Shape of your Data Set loaded: (600, 3)

C L A S S I F Y I N G V A R I A B L E S #####

Classifying variables in data set...
Data cleaning improvement suggestions. Complete them before proceeding to ML modeling.

| | Nuimques | dtype | Nuils | NullPercent | NuimquePercent | Value counts Min | Data cleaning improvement suggestions |
|-----------------------|----------|--------|-------|-------------|----------------|------------------|---------------------------------------|
| year | 75 | int64 | 0 | 0.000000 | 12.500000 | 0 | |
| nuclear_weapons_tests | 1 | int64 | 0 | 0.000000 | 6.833333 | 0 | |
| country_name | 8 | object | 0 | 0.000000 | 1.333333 | 75 | |

3 Predictors classified...
No variables removed since no ID or low-information variables found in data set



All Plots done
Time to run AutoViz = 4 seconds

AUTO VISUALIZATION Completed

DATA MANIPULATION

Carrying out data manipulation on my dataset in order to gain more insights from my dataset.

First, we find country with the highest and lowest nuclear weapons tests in a year

```
In [13]: # Country with the highest Nuclear weapons tests in a year
df.nlargest(1, 'nuclear_weapons_tests')
```

```
Out[13]:
```

| | country_name | year | nuclear_weapons_tests |
|-----|---------------|------|-----------------------|
| 542 | United States | 1962 | 96 |

```
In [14]: # Country with the smallest Nuclear tests in a year
df.nsmallest(1, 'nuclear_weapons_tests')
```

```
Out[14]:
```

| | country_name | year | nuclear_weapons_tests |
|---|--------------|------|-----------------------|
| 0 | China | 1945 | 0 |

Finding total sum of nuclear weapons tests per country and grouping them by country names

```
In [15]: # Find the total Nuclear weapon tests per country since 1945 to 2019
total = df.groupby('country_name', as_index=False)['nuclear_weapons_tests'].sum()
total.head()
```

```
Out[15]:
```

| | country_name | nuclear_weapons_tests |
|---|--------------|-----------------------|
| 0 | China | 45 |
| 1 | France | 210 |
| 2 | India | 3 |
| 3 | North Korea | 9 |
| 4 | Pakistan | 2 |

```
In [16]: total.tail()
```

```
Out[16]:
```

| | country_name | nuclear_weapons_tests |
|---|----------------|-----------------------|
| 3 | North Korea | 9 |
| 4 | Pakistan | 2 |
| 5 | Russia | 715 |
| 6 | United Kingdom | 45 |
| 7 | United States | 1030 |

DATA VISUALIZATION

Visualizing my data

CHART 1: HISTOGRAM SHOWING TOTAL NUCLEAR WEAPONS TEST PER COUNTRY FROM 1945-2019

Creating a histogram chart to show the total nuclear weapons test per country from 1945 to 2019

```
In [17]: # Create Chart
template.style = 'plotly.white'
fig = px.histogram(total,
```

```
    x = 'country_name',
    y = 'nuclear_weapons_tests',
    color = 'country_name',
    template = template.style,
    title = '<b>Total Nuclear Weapons Test Per Country 1945-2019</b>',
    width=800, height=400)
```

```
# Plot chart
fig.show()
```



CHART 2: USING AREA CHART TO CHECK FOR TRENDS

Checking for trends in nuclear weapons tests from 1945-2019 using Area Chart

```
In [18]: # Create Chart
fig = px.area(df,
```

```
    x = 'year',
    y = 'nuclear_weapons_tests',
    color = 'country_name',
    template = template.style,
    title = '<b>Area Chart Showing Trends Nuclear Tests</b>',
    width=800, height=400)
```

```
# Display Plot
fig.show()
```

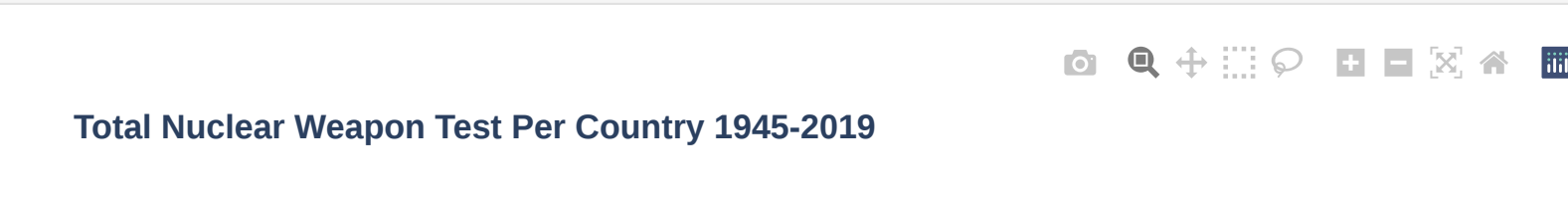


CHART 3: PIE CHART SHOWING PERCENTAGE OF WHOLE

I would create a pie chart showing the percentage each country has used in nuclear Weapons testing from 1945-2019

```
In [19]: # Create Chart
fig = px.pie(total, 'country_name',
    'nuclear_weapons_tests',
    color = 'country_name',
    title = '<b>Pie Chart Nuclear Tests</b>',
    width=800, height=400)
```

```
# Display Plot
fig.show()
```

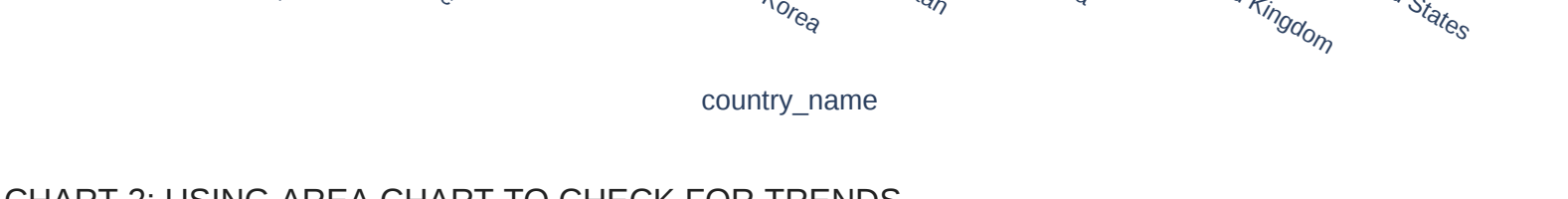


CHART 4: BAR CHART SHOWING AMOUNT OF YEARLY NUCLEAR WEAPONS TESTS PER EACH COUNTRY

```
In [20]: # Create a chart
template.style = 'plotly.white'
fig = px.bar(df,
```

```
    x = 'year',
    y = 'nuclear_weapons_tests',
    color = 'country_name',
    color_continuous_scale = ['green', 'yellow', 'red'],
    title = '<b>Yearly Nuclear Weapons Tests Per Country 1945-2019</b>',
    template = template.style,
    width=800, height=400)
```

```
# Display Plot
fig.show()
```



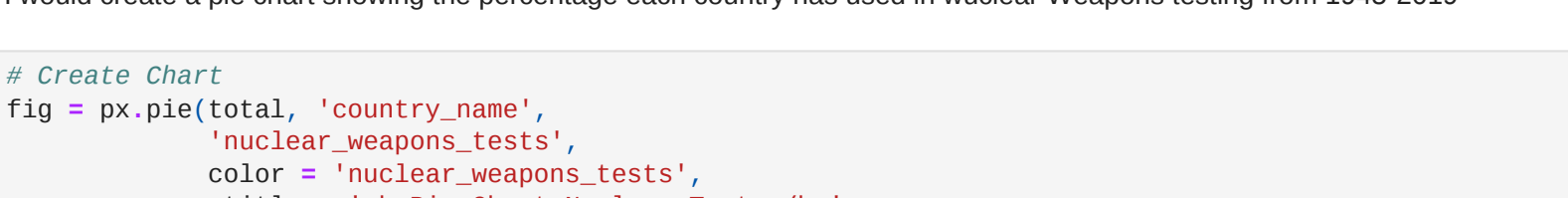
CHART 5: SCATTER PLOT SHOWING LINEAR CORRELATION BETWEEN YEAR/NUCLEAR WEAPONS TESTS

Is there any correlation between Year/Nuclear Weapons Tests & Tests [Scatter Plot]

```
In [21]: fig = px.scatter(df,
```

```
    x = 'year',
    y = 'nuclear_weapons_tests',
    color = 'country_name',
    width=800, height=400,
    title = '<b>Scatterplot Year/Nuclear Weapons Test</b>')
```

```
fig.show()
```



CONCLUSION

In this conclusion I would talk about the noticed trends in my insights

Since the end of world war 2, various nuclear weapons tests has been carried out by some countries, which is the basis of my analysis and insights. This analysis gave me some important insights on how countries has been testing nuclear weapons for years. What I found out was that:

- The United States carried out the highest nuclear weapons test from 1945 to 1949 with a total of 1030 tests.
- Russia had the second highest nuclear weapons tests from 1945 to 2019 with a total of 715 tests.
- Pakistan had the lowest nuclear weapons test with a total of 2 tests.
- The United States had the highest nuclear weapons tests in a calendar year with a total of 96 tests in 1962.
- China had no nuclear weapons tests in 1945 but had a total of 45 nuclear weapons tests since 1945 to 2019.
- The United States has tested about 50% of the world's nuclear weapons test from 1945 to 2019.
- Within 1951 to 1956 the United States has carried out about 36 nuclear weapons tests with each of the year having 18 tests each.
- The United States had a consistent nuclear weapons tests from 1971 to 1992, making it 21 years of consistent nuclear weapons testing.
- France also had a consistent nuclear weapons testing from 1960 to 1991 making it 21 years of consistent nuclear weapons testing.
- Russia's consistent nuclear weapons testing occurred between 1974 to 1985 making it also 21 years of consistent nuclear weapons testing.
- The U.S, Russia and France had 21 years each of consistent nuclear weapons testing.
- The United States, Russia and France had 94.9% of the world's nuclear weapons tests from 1945 to 2019.
- This report shows that the United States, Russia and France has carried out 94.9% of the world's nuclear weapons tests from 1945 to 2019.

```
In [ ]:
```