Software Engineering For Data Science (SEDS)

Class: 2nd Year 2nd Cycle
Branch: AIDS

Dr. Belkacem KHALDI| ESI-SBA

Lecture 07:

Data Processing & Cleaning for Data Science: Exploratory Data Analysis and Visualization– Going Deeper

Data Processing & Cleaning for Data Science

Part III: Exploratory Data Analysis and Visualization –Going deeper

1. Performing EDA with Seaborn and pandas
2. Using EDA Python packages
3. Using visualization best practices
4. Making Spatial plots with Plotly

Exploratory Data Analysis & Visualization

Exploratory Data Analysis

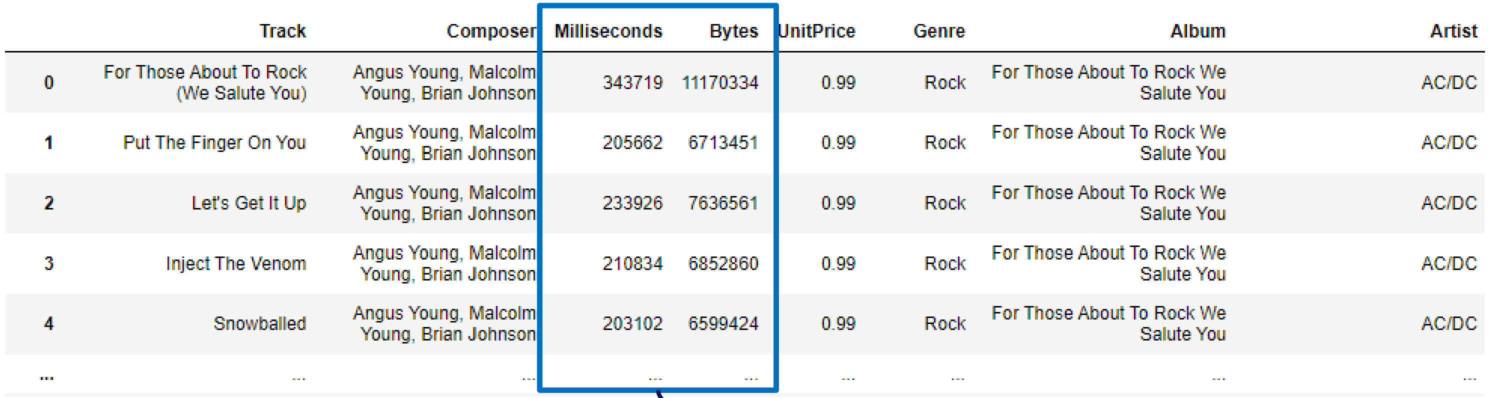
- ❑ **EDA**: A crucial step in any data science project
  - ○ A tool to better understand your data to properly use it.
  - ○ **EDA** is iterative and happens continually throughout a project.
  - ○ We also need to incorporate more advanced **EDA** to deepen our understanding.
- ❑ **Visualization** goes hand in hand with **EDA**.

Exploratory Data Analysis & Visualization

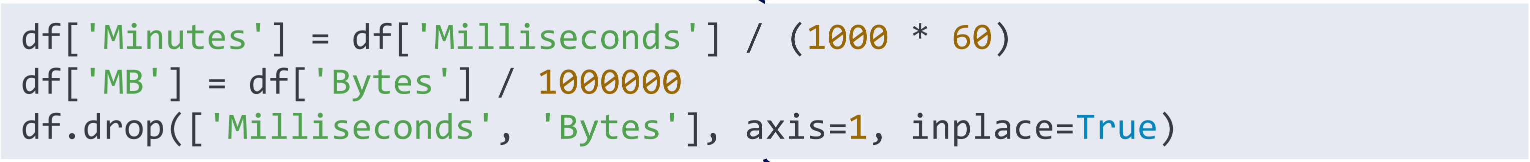
Performing EDA with Seaborn and Pandas

Dimensional Analysis (DA)

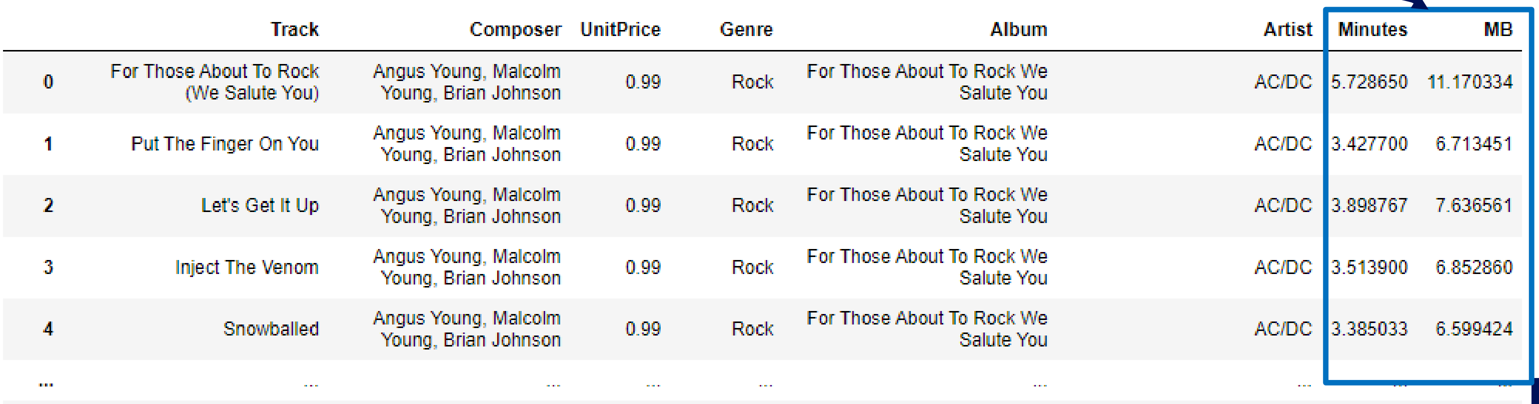
- ❑ **DA** ➔ Technique of analyzing the relationships between different physical quantities by identifying their base quantities (such as **length**, **mass**, **time**, ...) and common **units of measure**.
- ❑ Example of the Itune dataset:
  - ○ 'Milliseconds' ➔ 'Minutes'
  - ○ 'Bytes' ➔ 'MB'



	Track	Composer	Milliseconds	Bytes	UnitPrice	Genre	Album	Artist
0	For Those About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	343719	11170334	0.99	Rock	For Those About To Rock We Salute You	AC/DC
1	Put The Finger On You	Angus Young, Malcolm Young, Brian Johnson	205662	6713451	0.99	Rock	For Those About To Rock We Salute You	AC/DC
2	Let's Get It Up	Angus Young, Malcolm Young, Brian Johnson	233926	7636561	0.99	Rock	For Those About To Rock We Salute You	AC/DC
3	Inject The Venom	Angus Young, Malcolm Young, Brian Johnson	210834	6852860	0.99	Rock	For Those About To Rock We Salute You	AC/DC
4	Snowballed	Angus Young, Malcolm Young, Brian Johnson	203102	6599424	0.99	Rock	For Those About To Rock We Salute You	AC/DC
...	...	...	...	...	...	...	...	...



```
df['Minutes'] = df['Milliseconds'] / (1000 * 60)
df['MB'] = df['Bytes'] / 1000000
df.drop(['Milliseconds', 'Bytes'], axis=1, inplace=True)
```



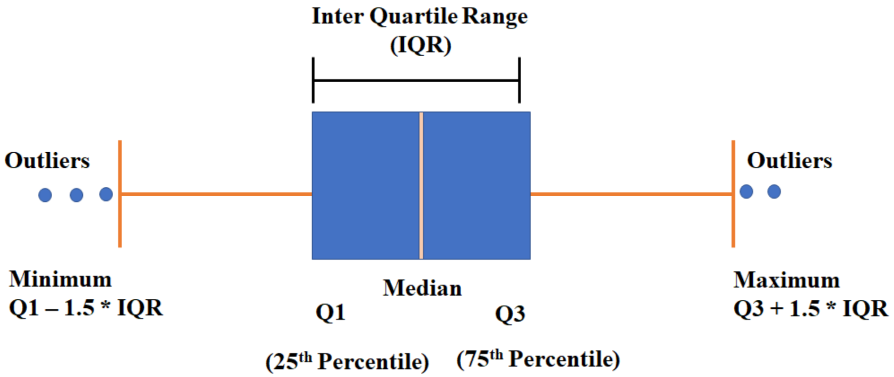
	Track	Composer	UnitPrice	Genre	Album	Artist	Minutes	MB
0	For Those About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	For Those About To Rock We Salute You	AC/DC	5.728650	11.170334
1	Put The Finger On You	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	For Those About To Rock We Salute You	AC/DC	3.427700	6.713451
2	Let's Get It Up	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	For Those About To Rock We Salute You	AC/DC	3.898767	7.636561
3	Inject The Venom	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	For Those About To Rock We Salute You	AC/DC	3.513900	6.852860
4	Snowballed	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	For Those About To Rock We Salute You	AC/DC	3.385033	6.599424
...	...	...	...	...	...	...	...	...

Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Boxplots and Letter-Value plots

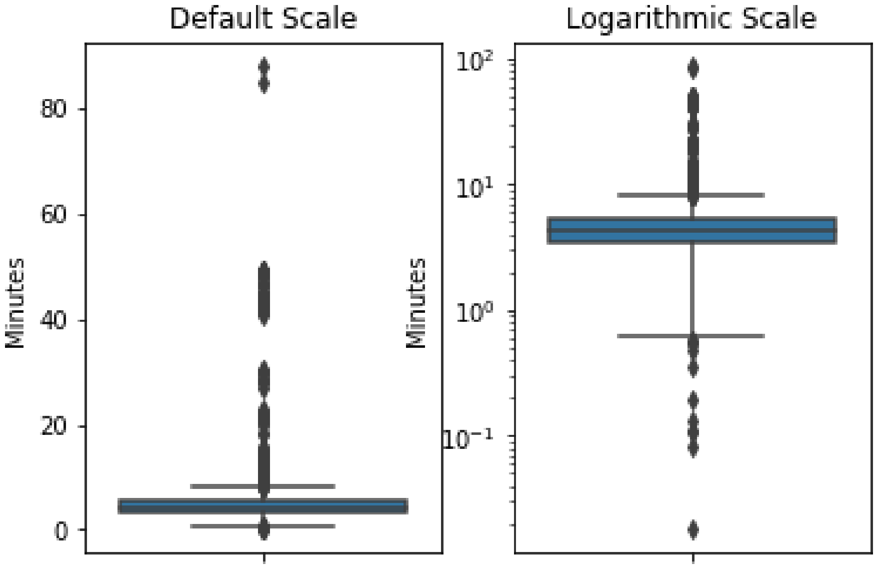
- **Boxplots:**
  - Invented in **1970** by **John Tukey**.
  - Helps to quickly see information about the distribution of a dataset and enables comparing subsets of data.
  - Data are plotted according to the **IQR** formula:



```
import seaborn as sns

fig, axes = plt.subplots(nrows=1, ncols=2)
sns.boxplot(y=df['Minutes'],ax=axes[0])
sns.boxplot(y=df['Minutes'],ax=axes[1])
plt.yscale('log')

axes[0].set_title("Default Scale")
axes[1].set_title("Logarithmic Scale")
```



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

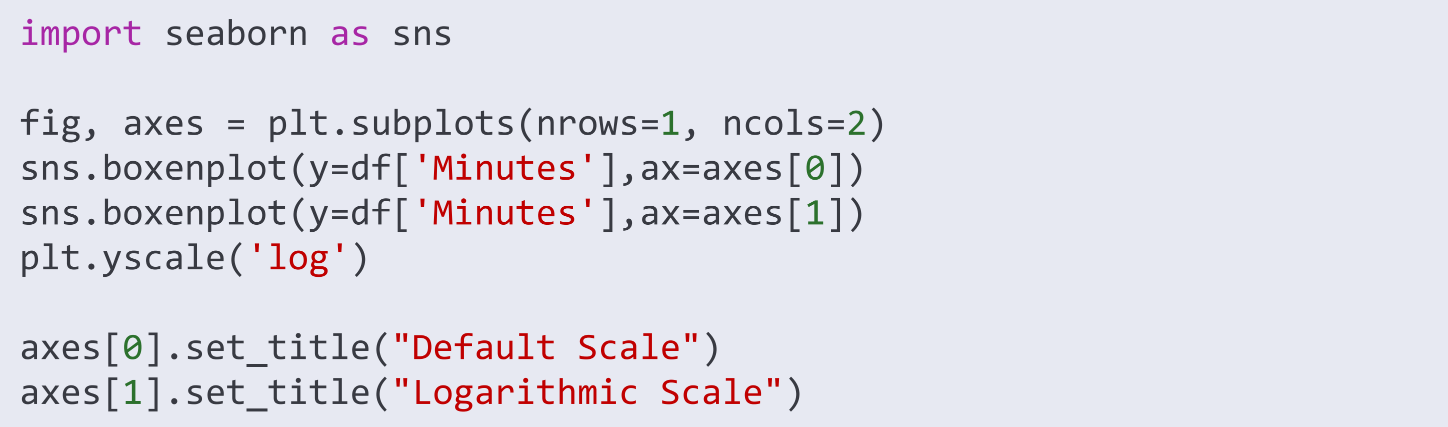
```
count    3503.000000
mean        6.559987
std         8.916757
min         0.017850
25%         3.454683
50%         4.260567
75%         5.360750
max        88.115883
Name: Minutes, dtype: float64
```

Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Boxplots and Letter-Value plots

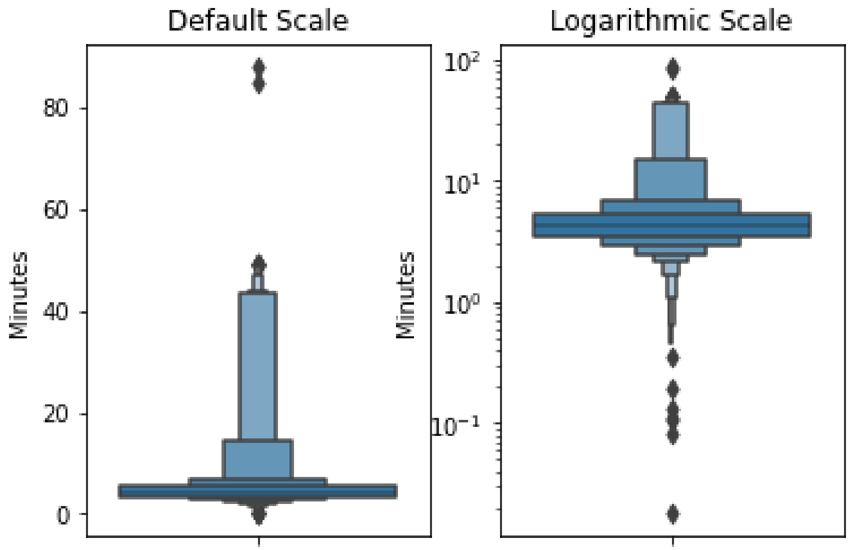
- **Letter-Value Plots:**
  - Invented around **2011** by **Hadley Wickham.**
  - Helps improving the boxplot's shortcomings.
  - Instead of showing outliers outside the **IQR** ➔ Plotting outliers with a letter-value plot results in 5 to 8 outliers on the upper and lower extremes.
  - Shows the distribution better by grouping data into more quantiles.



```
import seaborn as sns

fig, axes = plt.subplots(nrows=1, ncols=2)
sns.boxenplot(y=df['Minutes'],ax=axes[0])
sns.boxenplot(y=df['Minutes'],ax=axes[1])
plt.yscale('log')

axes[0].set_title("Default Scale")
axes[1].set_title("Logarithmic Scale")
```



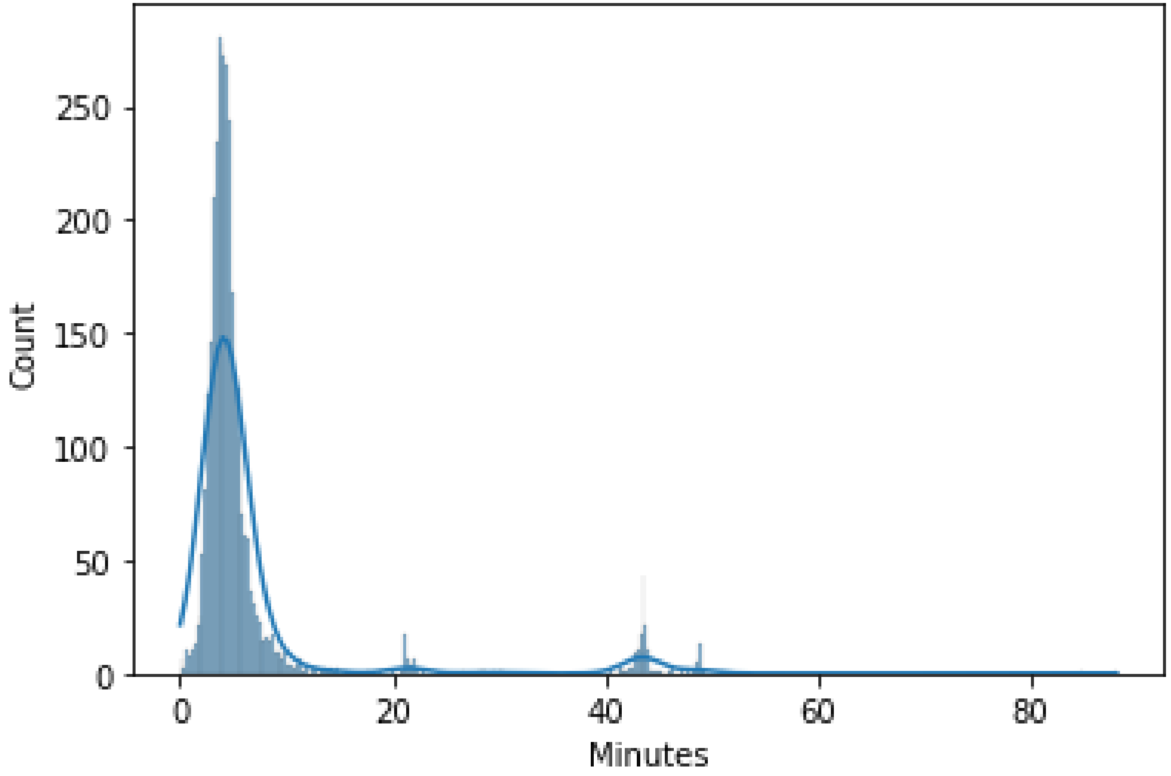
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making histograms and violin plots

```
import seaborn as sns
sns.histplot(x=df['Minutes'], kde=True)
```

- **Histograms Plots:**
  - Another way to see the distribution of data is using **histograms** and **Kernel Density Estimation (KDE)**.
  - **KDE** fits a line to the distribution of data and produces a smoothed histogram.
  - The resulting plot shows bars that represent the density of the data – bigger bars mean more points. The line is the **KDE** fit to the data.



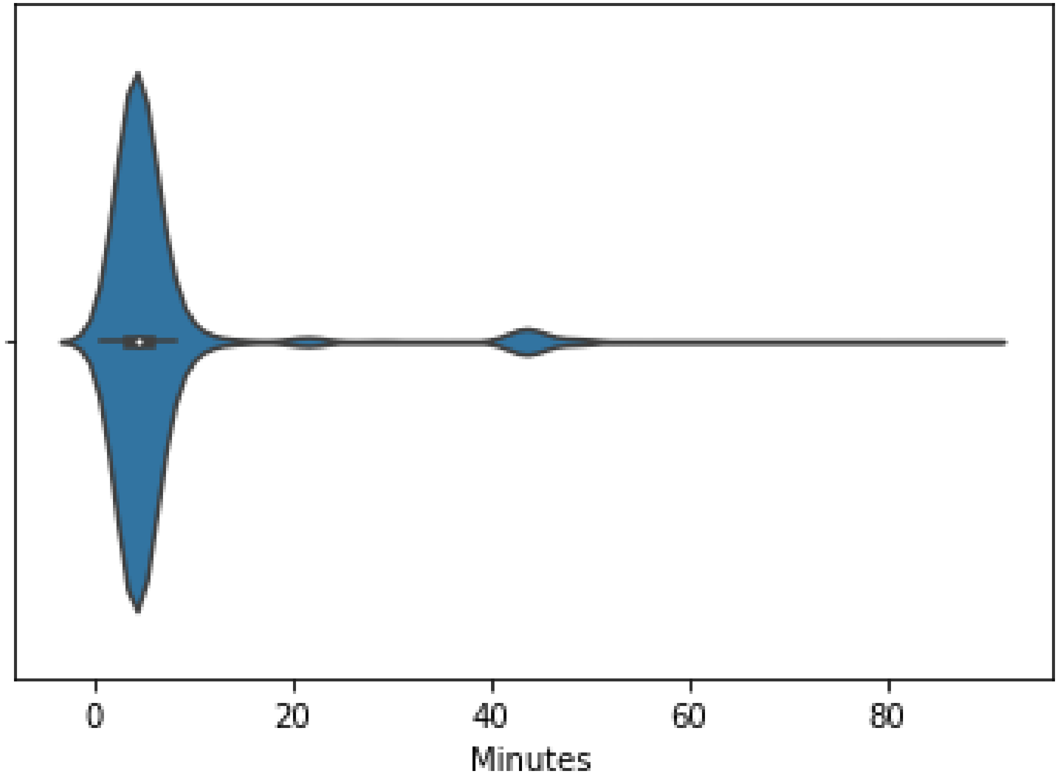
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making histograms and violin plots

```python
import seaborn as sns
sns.violinplot(data=df, x='Minutes')
```

- ❑ **Violin Plots:**
  - ○ A **violin plot** is similar, but shows the **KDE** and a boxplot.
  - ○ The **KDE** is the main feature of the plot, and it is **mirrored** on the **x axis**.
  - ○ A small **boxplot** in the middle of the **mirrored** KDE distribution is also shown.



Exploratory Data Analysis & Visualization

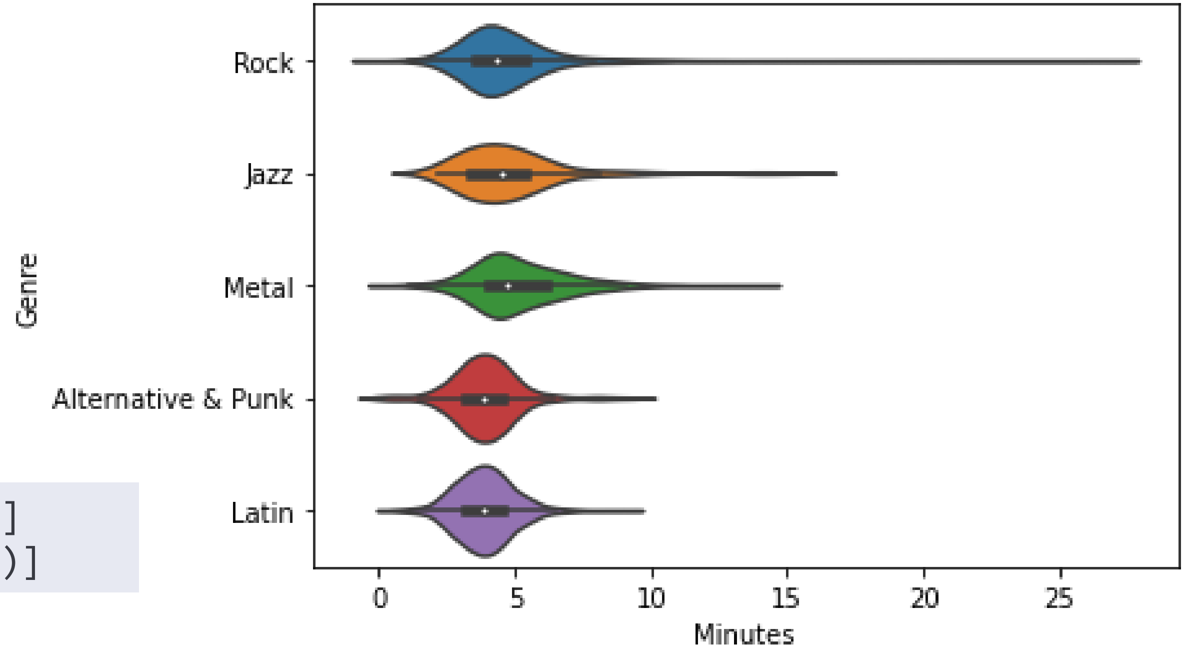
Performing EDA with Seaborn and Pandas

Making histograms and violin plots

```
import seaborn as sns
sns.violinplot(data=top_5_data, x='Minutes', y='Genre')
```

- ❑ **Violin Plots:**
  - ○ Possibility to plot by a few groups of data at once with a violin plot.
  - ○ **Example:**
    - Plotting the top 5 genres by Minutes of songs length.

```
top_5_genres = df['Genre'].value_counts().index[:5]
top_5_data = data=df[df['Genre'].isin(top_5_genres)]
```

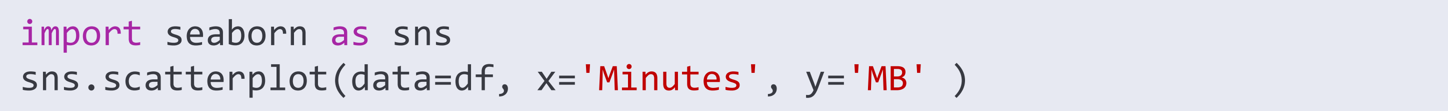


Exploratory Data Analysis & Visualization

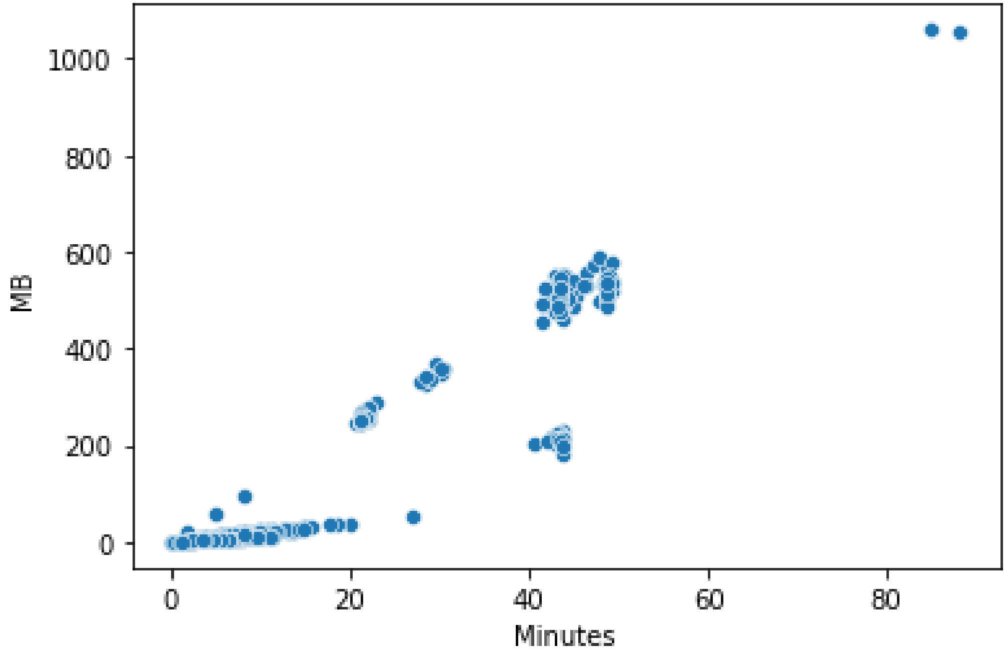
Performing EDA with Seaborn and Pandas

Making Scatter Plots

- ❑ **Scatter** plots ➔ Essential **EDA** plot for **continuous**, **numeric** data.
  - ○ **Continuous data**: data that can take any **value** between two **bounds**, such as **length**, or **temperature**.
  - ○ **Example: Let's take a look at song length versus size in MB.**



```
import seaborn as sns
sns.scatterplot(data=df, x='Minutes', y='MB' )
```



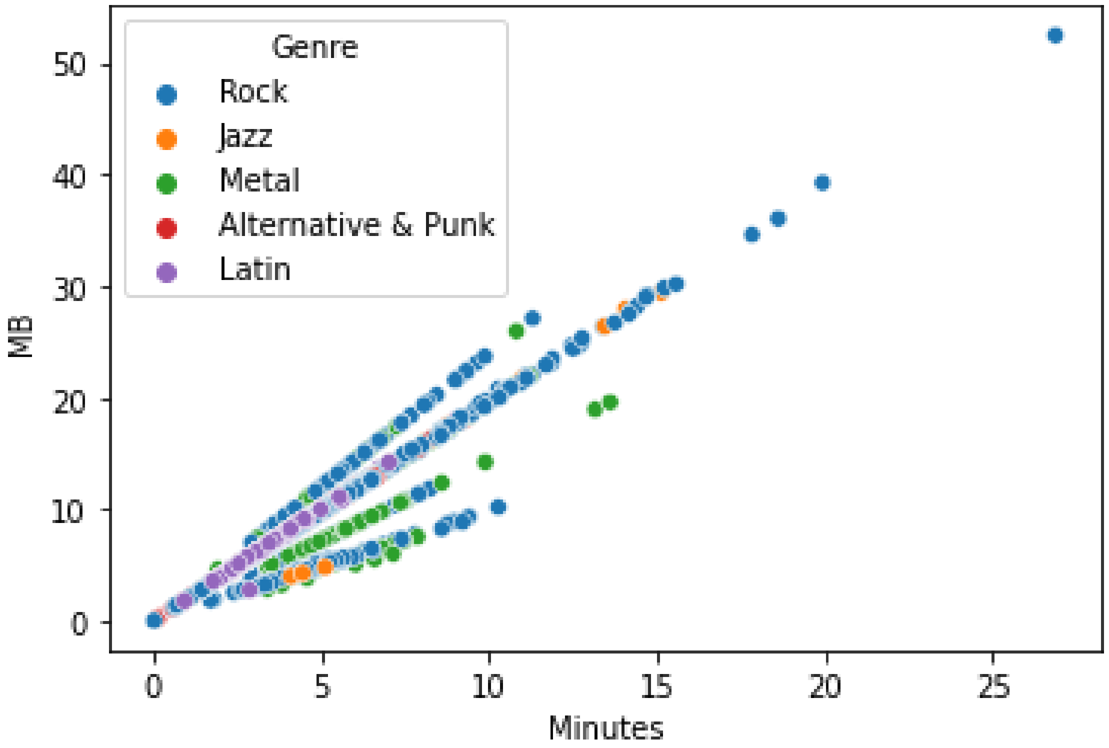
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Scatter Plots

- ❑ **Scatter** plots ➔ Essential **EDA** plot for **continuous**, **numeric** data.
  - ○ Possibility to group by a column using the **hue** argument**.**
  - ○ **Example:**
    - ▪ Grouping by 'Genre' of the top five genres by song munites,

```python
import seaborn as sns
sns.scatterplot(data=top_5_data, x='Minutes', y='MB', hue='Genre')
```



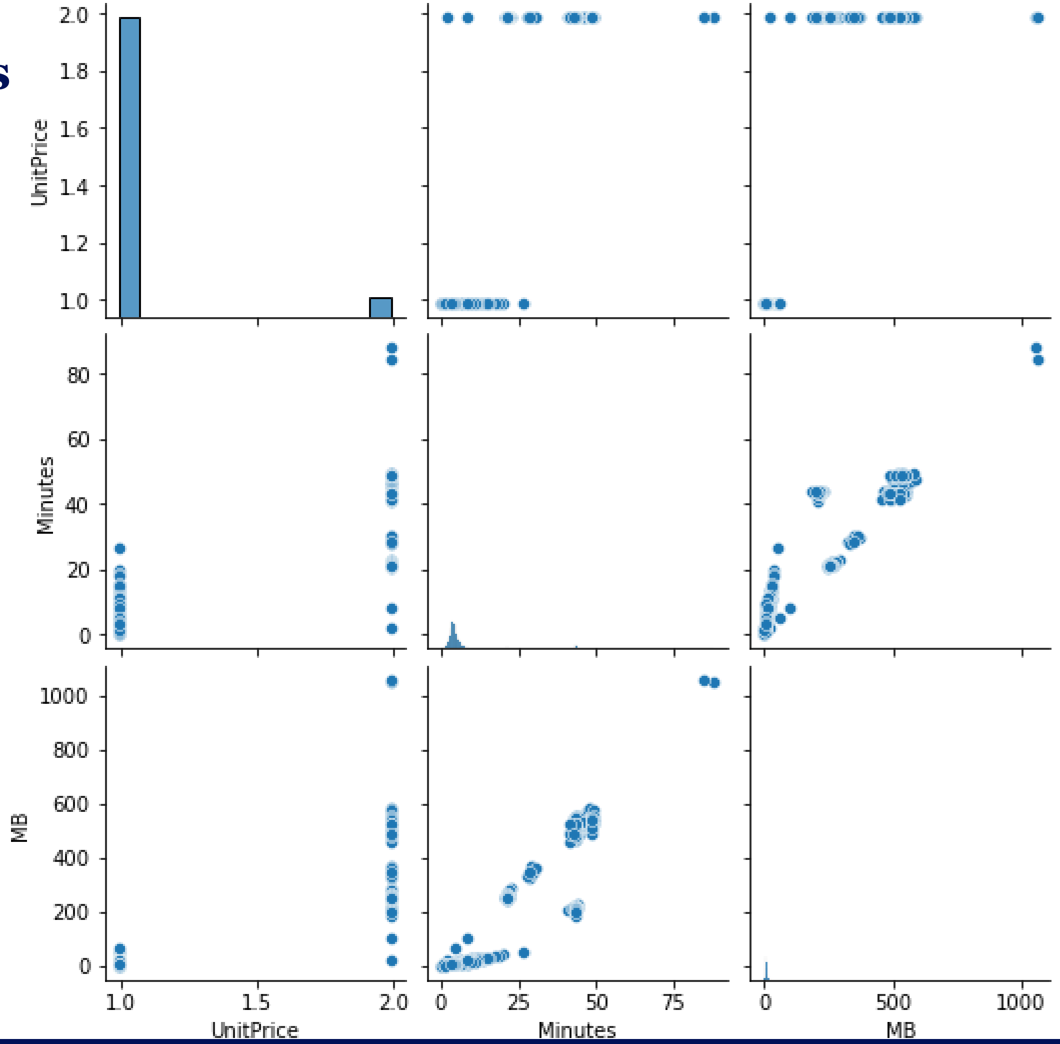
Exploratory Data Analysis & Visualization

Performing EDA with Seborn and Pandas

Making Correlograms & Examining Correlations

- ❑ A **Correlogram** ➔ allows to analyze the relationship between each pair of numeric variables of a dataset.
- ❑ Can be gotten using one line of code using the seaborn **pairplot** built-function

```
sns.pairplot(data=df)
```



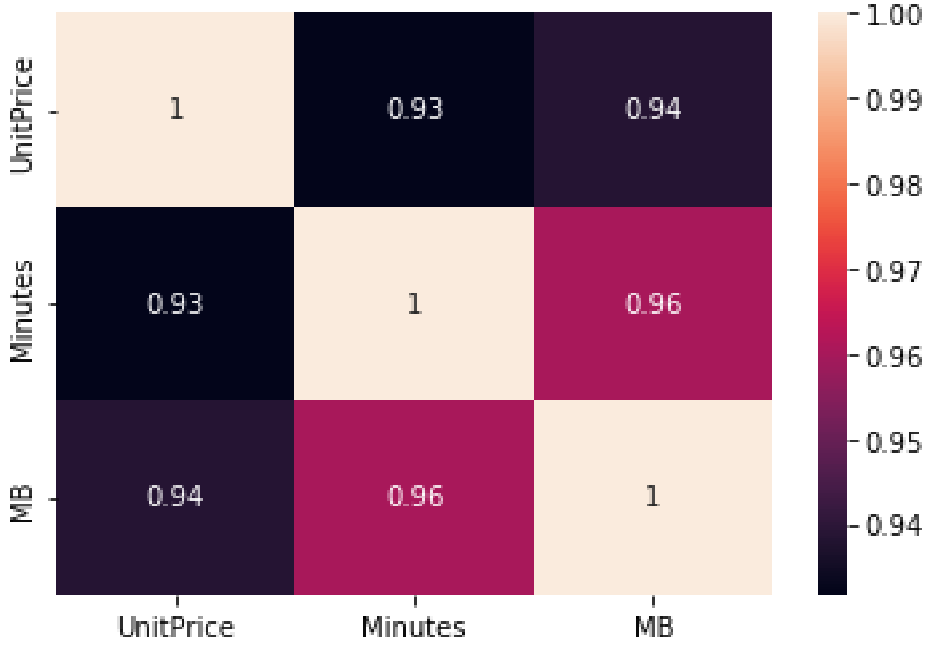
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Correlograms & Examining Correlations

- We often want to see how strongly correlated different numeric columns are.
- Correlation Matrix can be gotten using **DataFrame.corr() pandas built-in function**. We can simply plot it as follows:

```
sns.heatmap(df.corr(), annot=True)
```



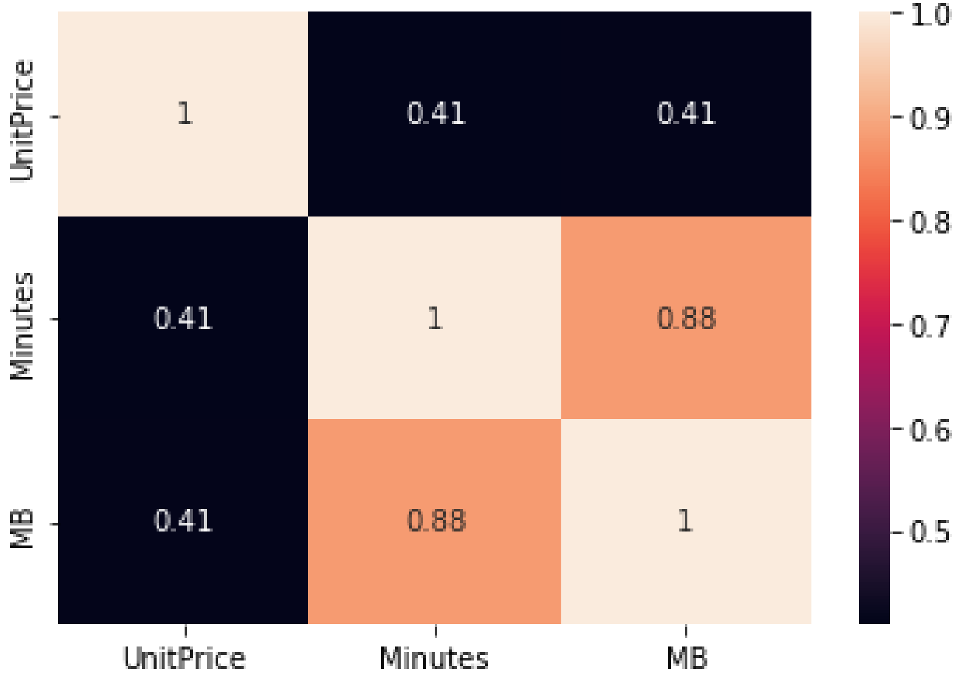
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Correlograms & Examining Correlations

- ❑ Other types of correlations are available such as **Spearman Correlation**.
- ❑ Better suited for **non-linear relationships.**
- ❑ Spearman Correlation Matrix can be simply plotted with seaborn's heatmap as follows:

```
sns.heatmap(df.corr(method='spearman'), annot=True)
```



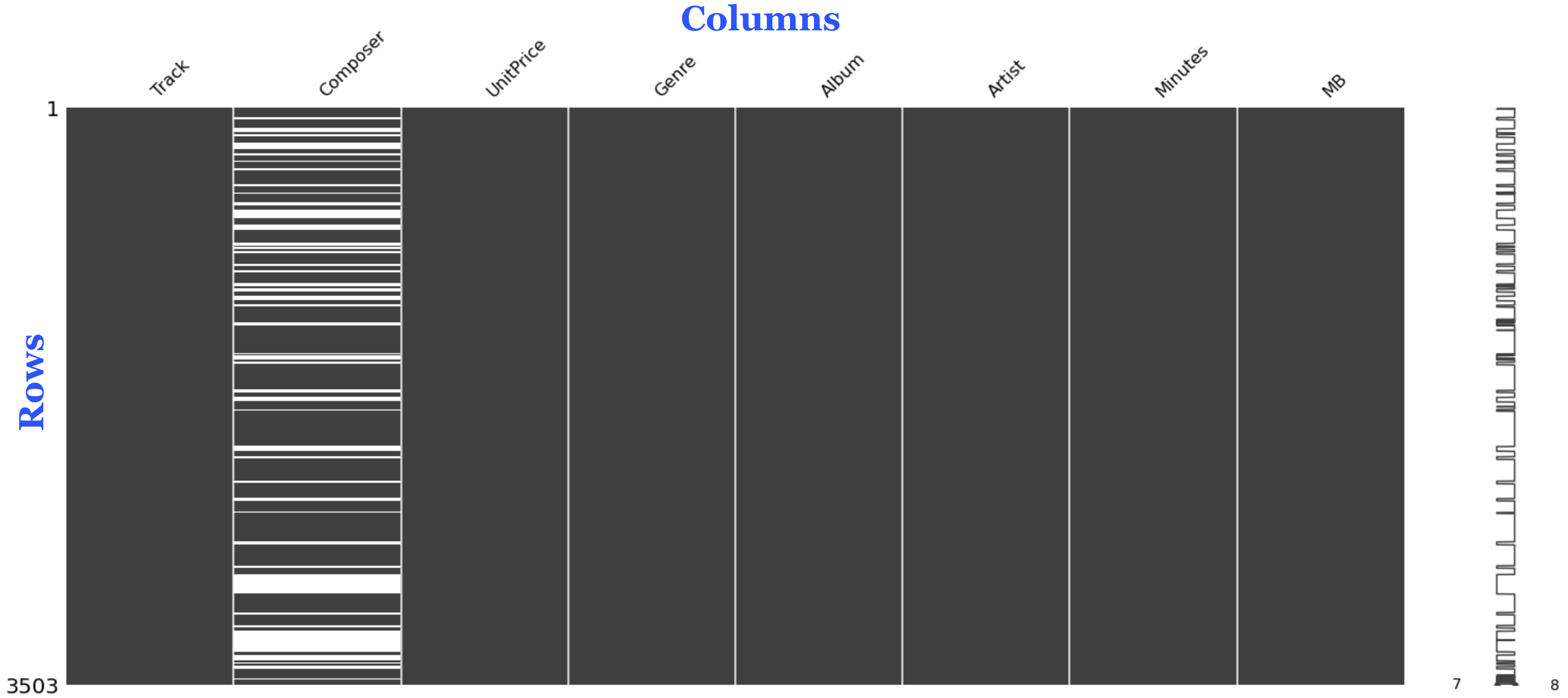
Exploratory Data Analysis & Visualization

Performing EDA with Seaborn and Pandas

Making Missing Value Plots

- ❑ Missing values can be examined with pandas built-in functions:
  - ○ **DataFrame.isna().sum()**
  - ○ **DataFrame.info()**
- ○ But, it can be easier to look at a visualization with the help of the **missingno** package.
- ○ It shows a matrix of **non-missing values** in **gray** and missing values in **white**.

```python
import missingno as msno
msno.matrix(df)
```



Columns

Rows

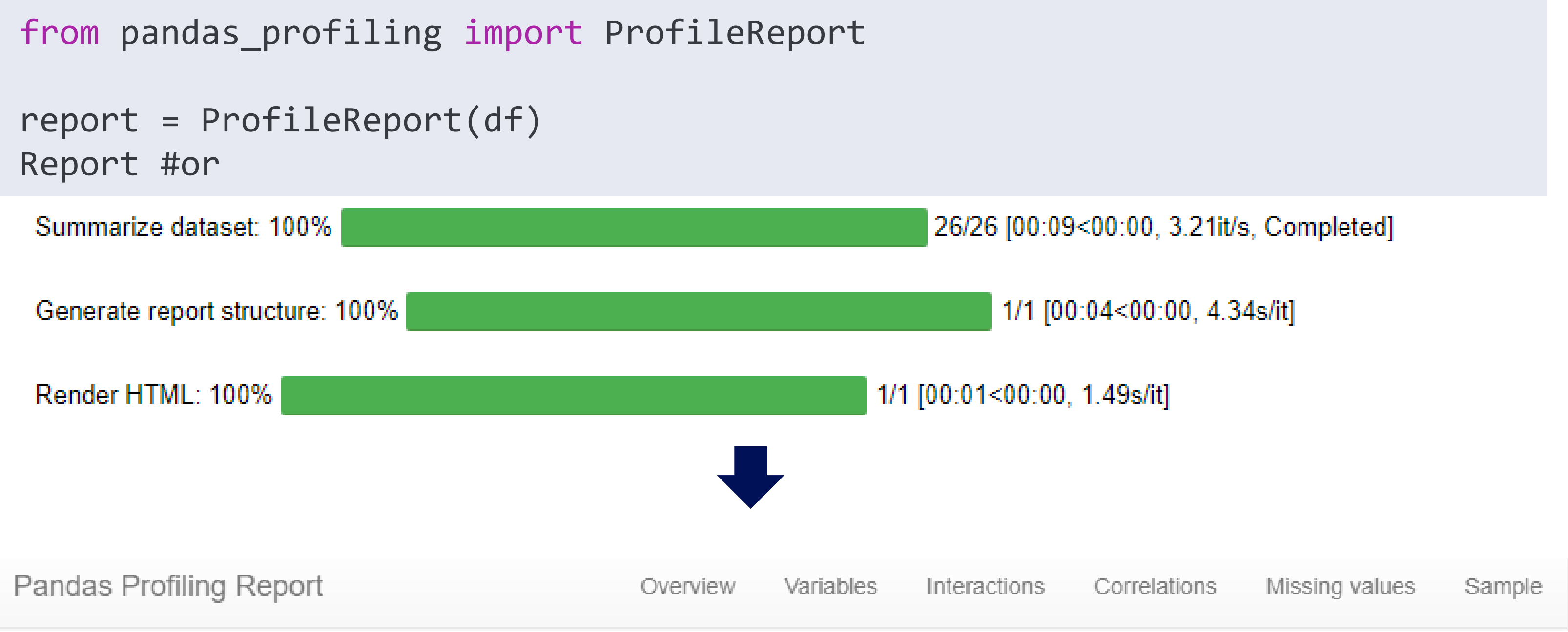
Min, Max # of no-missing Values per Rows

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



```
from pandas_profiling import ProfileReport

report = ProfileReport(df)
Report #or
```

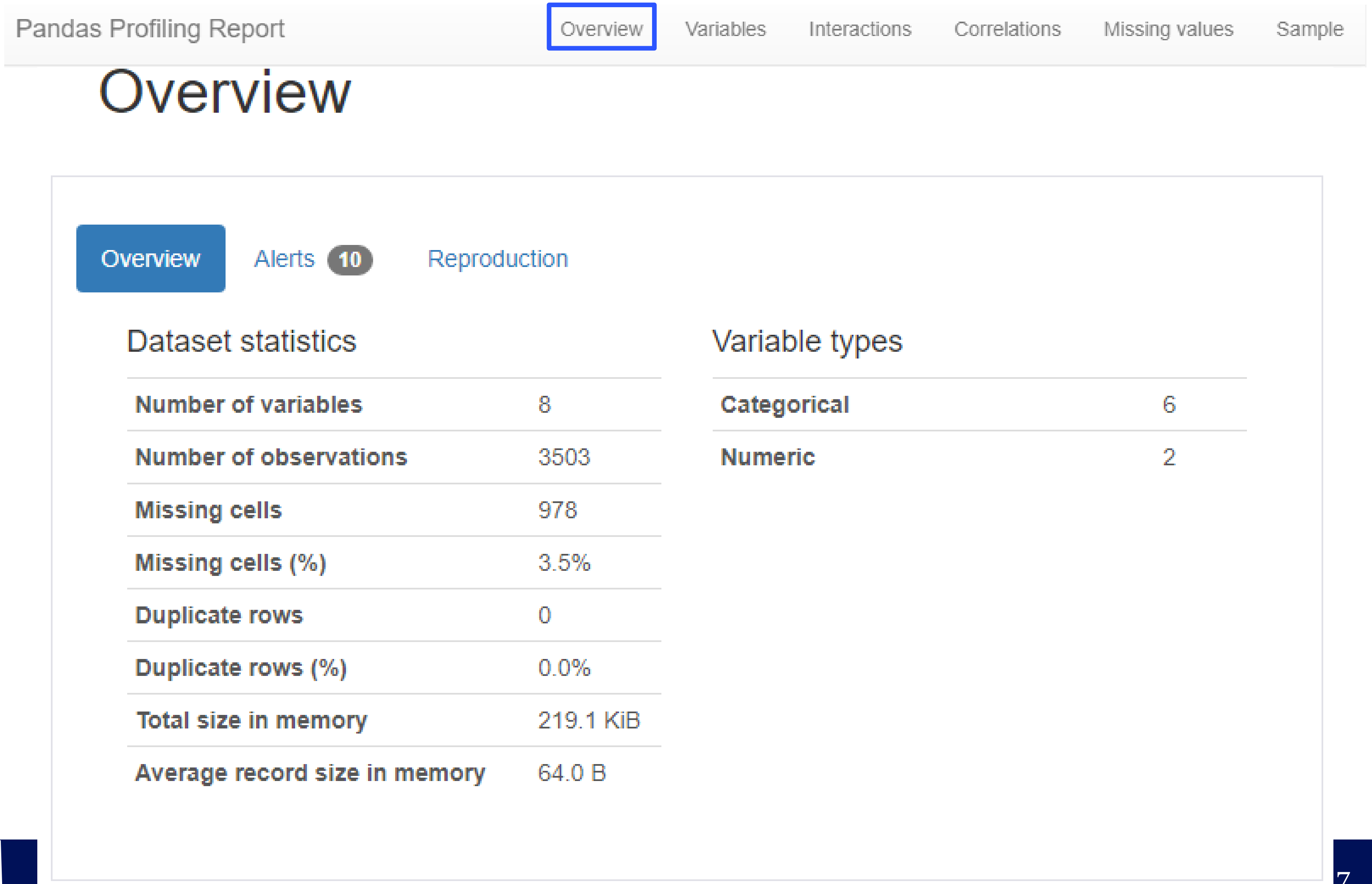
```
Summarize dataset: 100%  26/26 [00:09<00:00, 3.21it/s, Completed]
Generate report structure: 100%  1/1 [00:04<00:00, 4.34s/it]
Render HTML: 100%  1/1 [00:01<00:00, 1.49s/it]
```

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Overview

Dataset statistics

Number of variables	8
Number of observations	3503
Missing cells	978
Missing cells (%)	3.5%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	219.1 KiB
Average record size in memory	64.0 B

Variable types

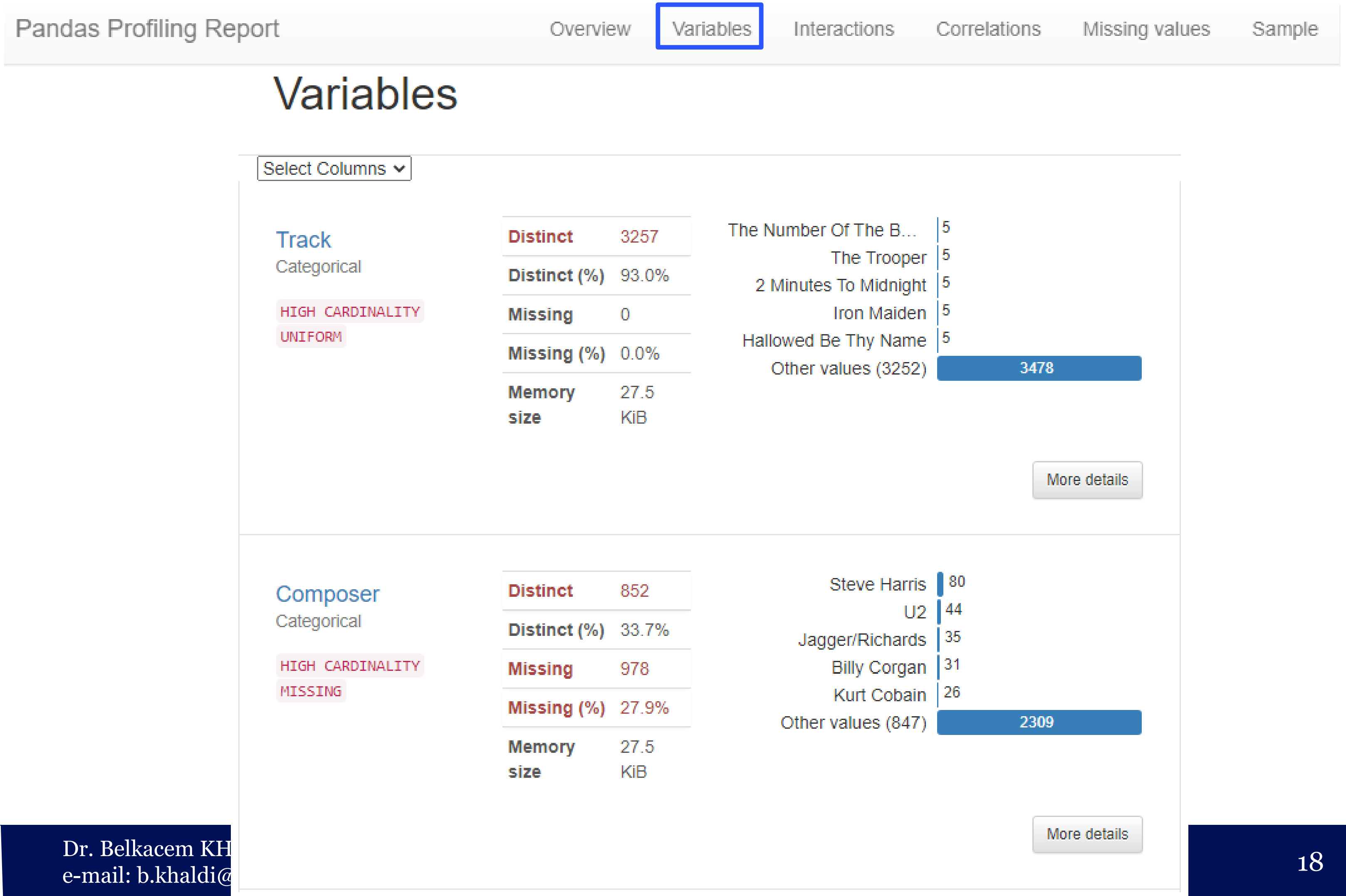
Categorical	6
Numeric	2

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
    - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Variables

Track
Categorical
HIGH CARDINALITY
UNIFORM

Distinct	3257
Distinct (%)	93.0%
Missing	0
Missing (%)	0.0%
Memory size	27.5 KiB

The Number Of The B...	5
The Trooper	5
2 Minutes To Midnight	5
Iron Maiden	5
Hallowed Be Thy Name	5
Other values (3252)	3478

Composer
Categorical
HIGH CARDINALITY
MISSING

Distinct	852
Distinct (%)	33.7%
Missing	978
Missing (%)	27.9%
Memory size	27.5 KiB

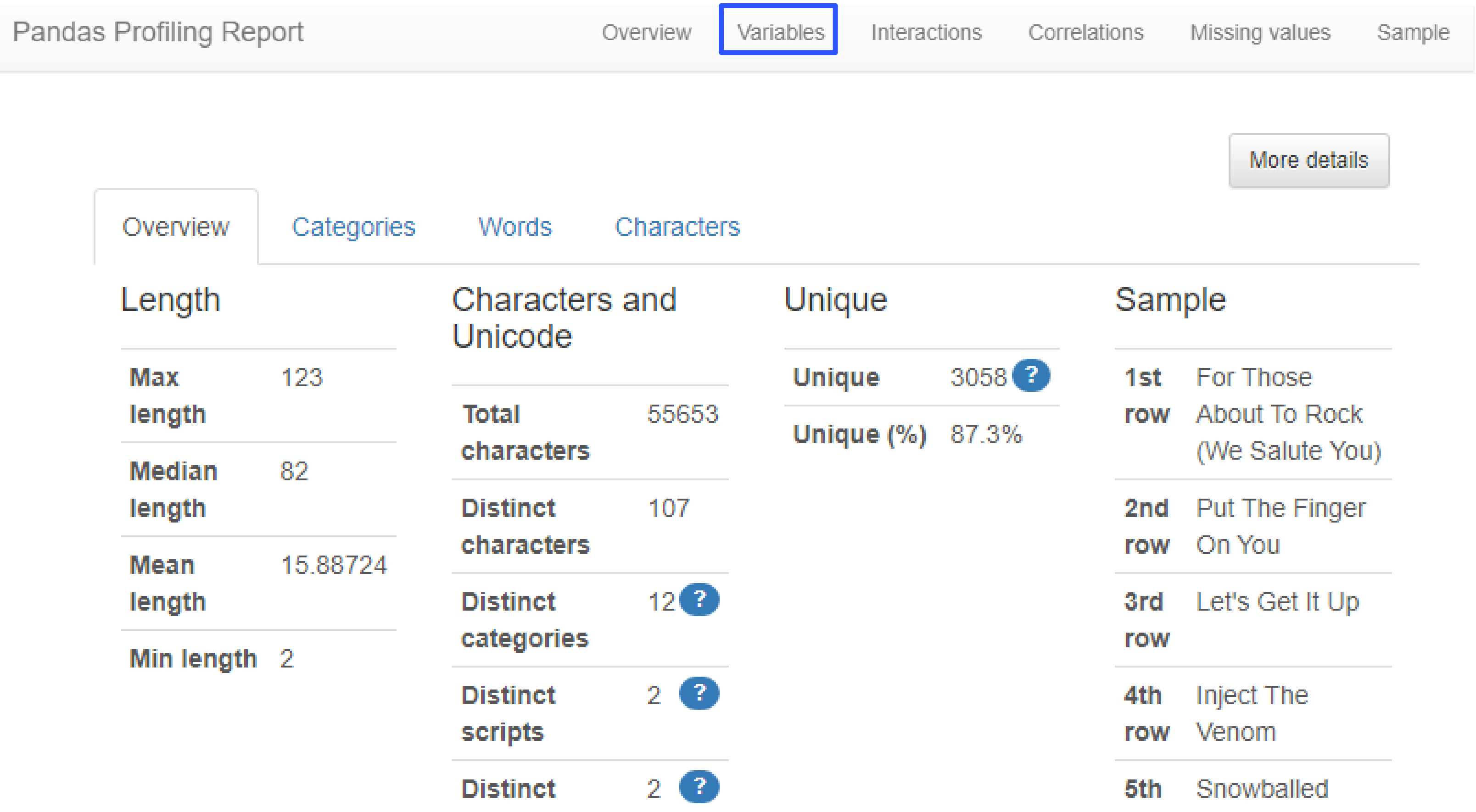
Steve Harris	80
U2	44
Jagger/Richards	35
Billy Corgan	31
Kurt Cobain	26
Other values (847)	2309

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- Sometimes it's helpful to run an auto-EDA package on the dataset.
  - We will cover the **pandas-profiling EDA** package.
    - A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Length

Max length	123
Median length	82
Mean length	15.88724
Min length	2

Characters and Unicode

Total characters	55653
Distinct characters	107
Distinct categories	12 ?
Distinct scripts	2 ?
Distinct	2 ?

Unique

Unique	3058 ?
Unique (%)	87.3%

Sample

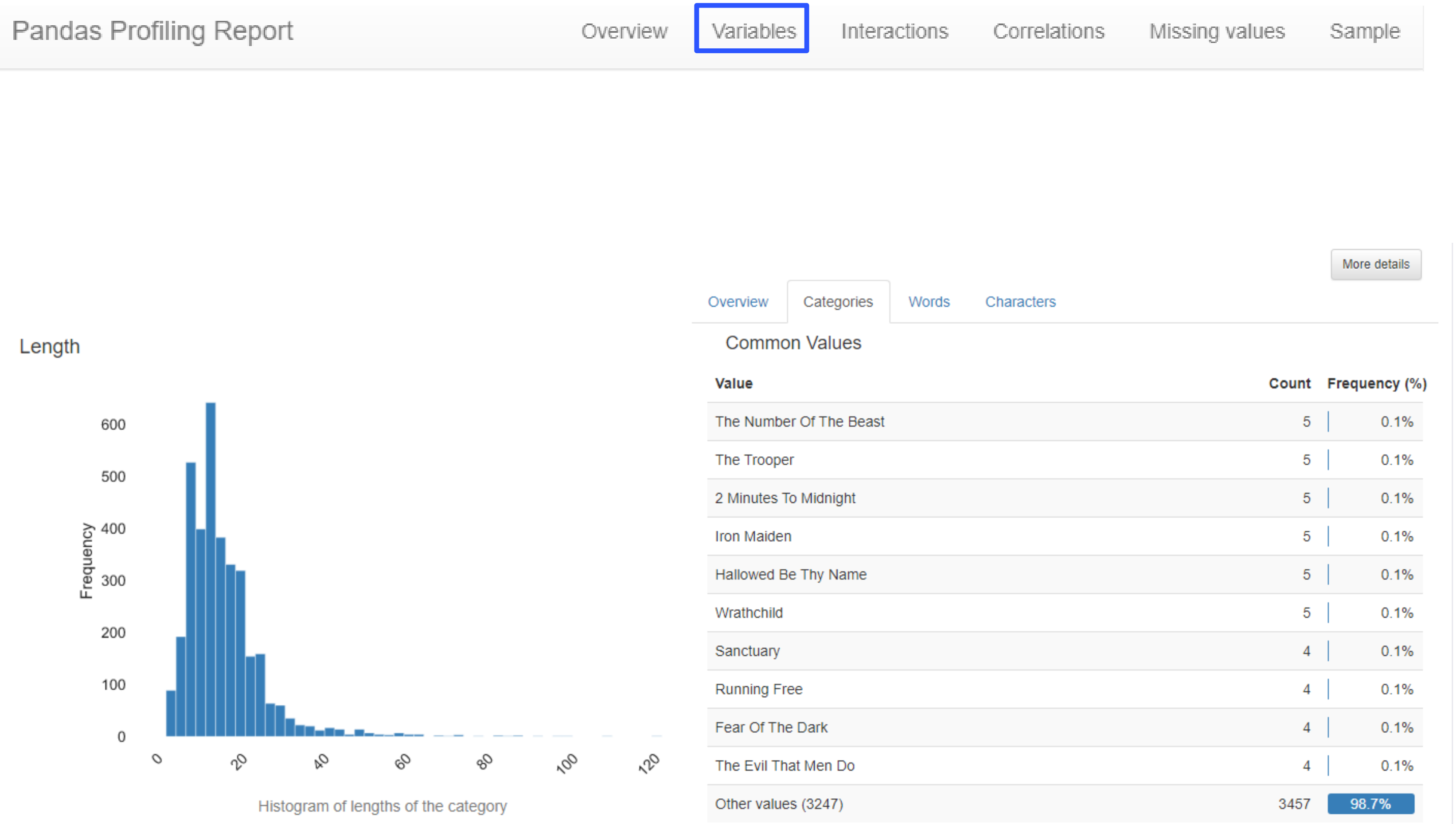
1st row	For Those About To Rock (We Salute You)
2nd row	Put The Finger On You
3rd row	Let's Get It Up
4th row	Inject The Venom
5th	Snowballed

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
  - ○ We will cover the **pandas-profiling EDA** package.
    - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Common Values

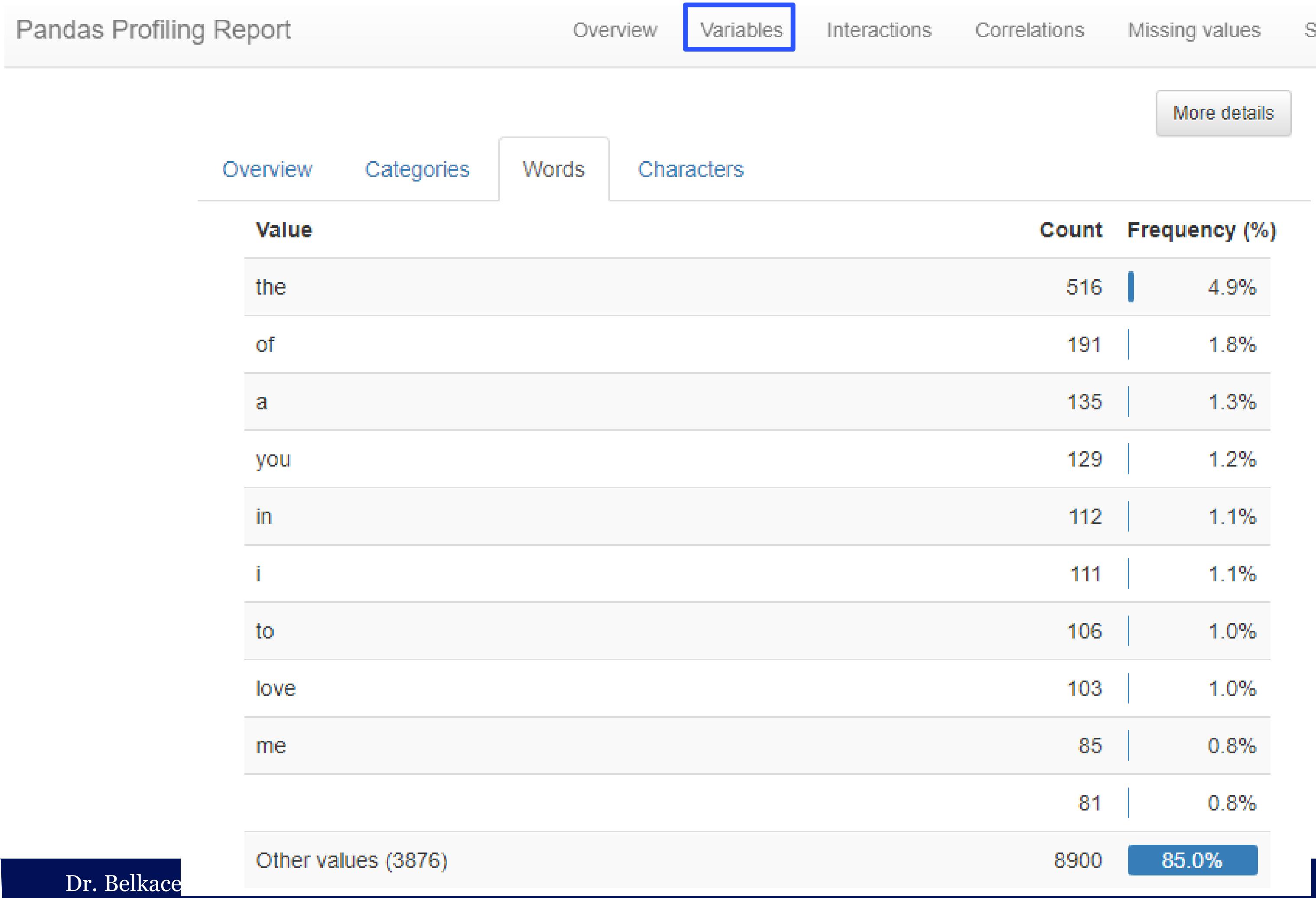
Value	Count	Frequency (%)
The Number Of The Beast	5	0.1%
The Trooper	5	0.1%
2 Minutes To Midnight	5	0.1%
Iron Maiden	5	0.1%
Hallowed Be Thy Name	5	0.1%
Wrathchild	5	0.1%
Sanctuary	4	0.1%
Running Free	4	0.1%
Fear Of The Dark	4	0.1%
The Evil That Men Do	4	0.1%
Other values (3247)	3457	98.7%

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



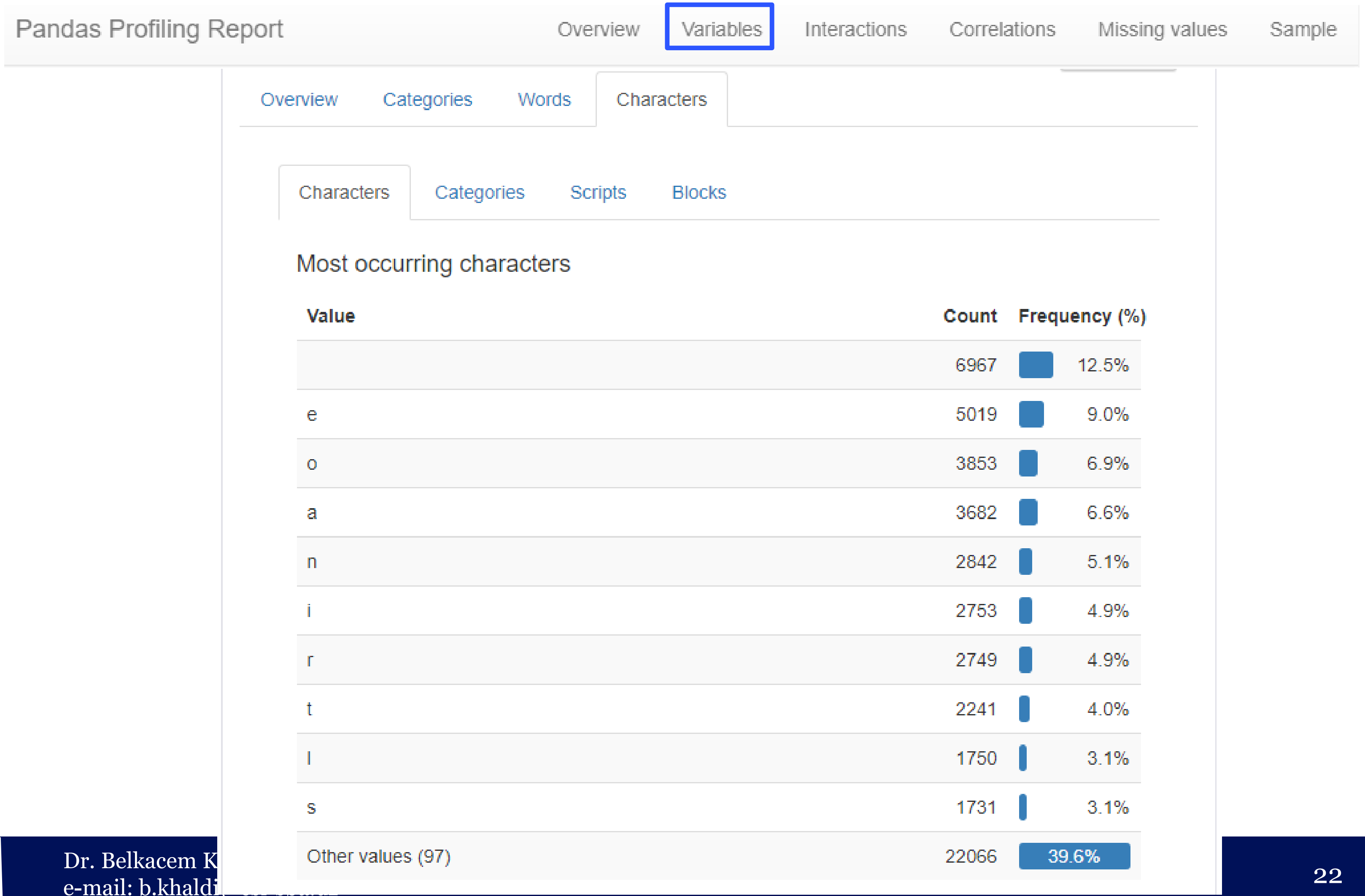
Value	Count	Frequency (%)
the	516	4.9%
of	191	1.8%
a	135	1.3%
you	129	1.2%
in	112	1.1%
i	111	1.1%
to	106	1.0%
love	103	1.0%
me	85	0.8%
	81	0.8%
Other values (3876)	8900	85.0%

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
  - ○ We will cover the **pandas-profiling EDA** package.
    - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Most occurring characters

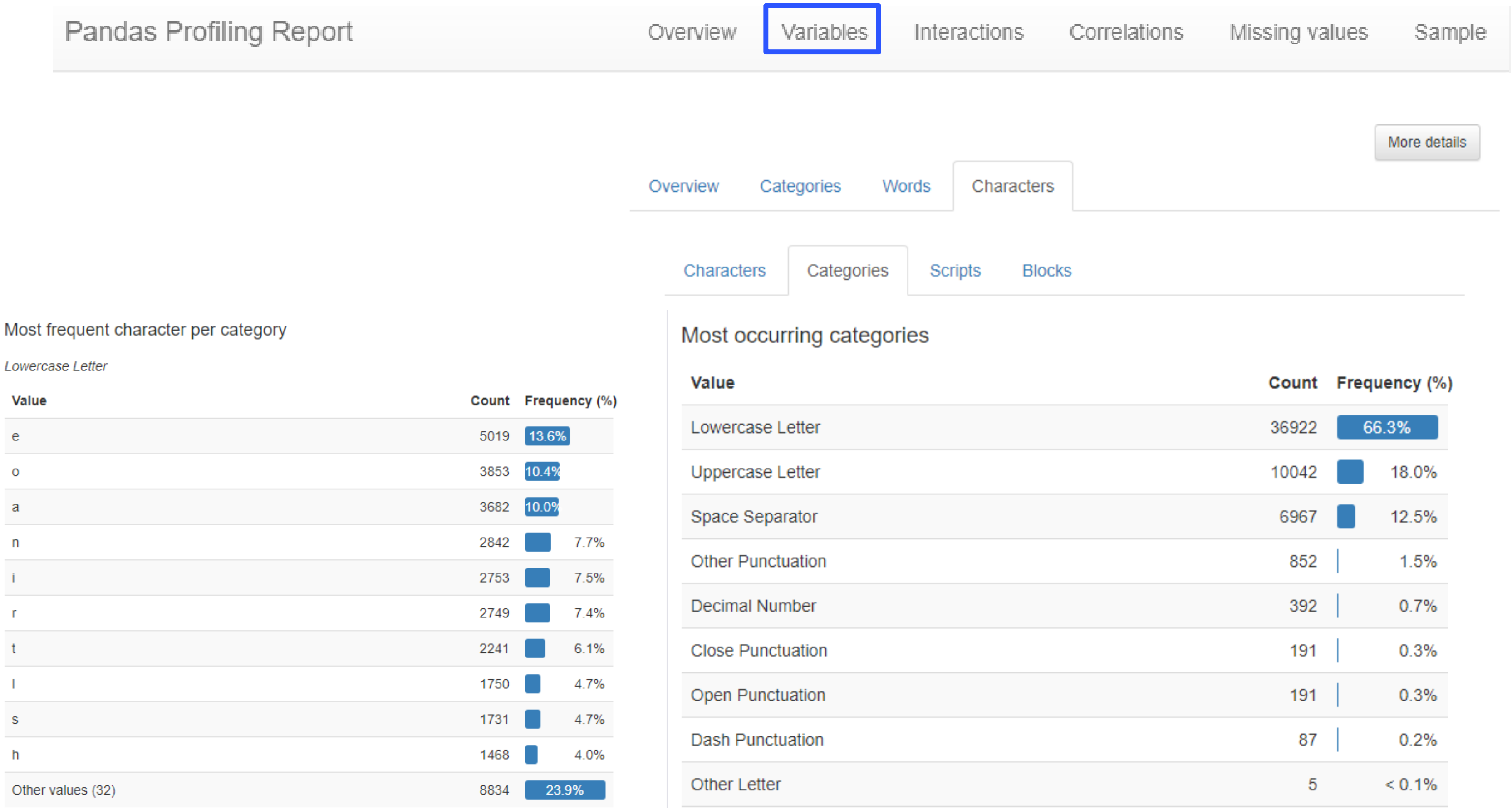
Value	Count	Frequency (%)
	6967	12.5%
e	5019	9.0%
o	3853	6.9%
a	3682	6.6%
n	2842	5.1%
i	2753	4.9%
r	2749	4.9%
t	2241	4.0%
l	1750	3.1%
s	1731	3.1%
Other values (97)	22066	39.6%

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Most frequent character per category

Lowercase Letter

Value	Count	Frequency (%)
e	5019	13.6%
o	3853	10.4%
a	3682	10.0%
n	2842	7.7%
i	2753	7.5%
r	2749	7.4%
t	2241	6.1%
l	1750	4.7%
s	1731	4.7%
h	1468	4.0%
Other values (32)	8834	23.9%

Most occurring categories

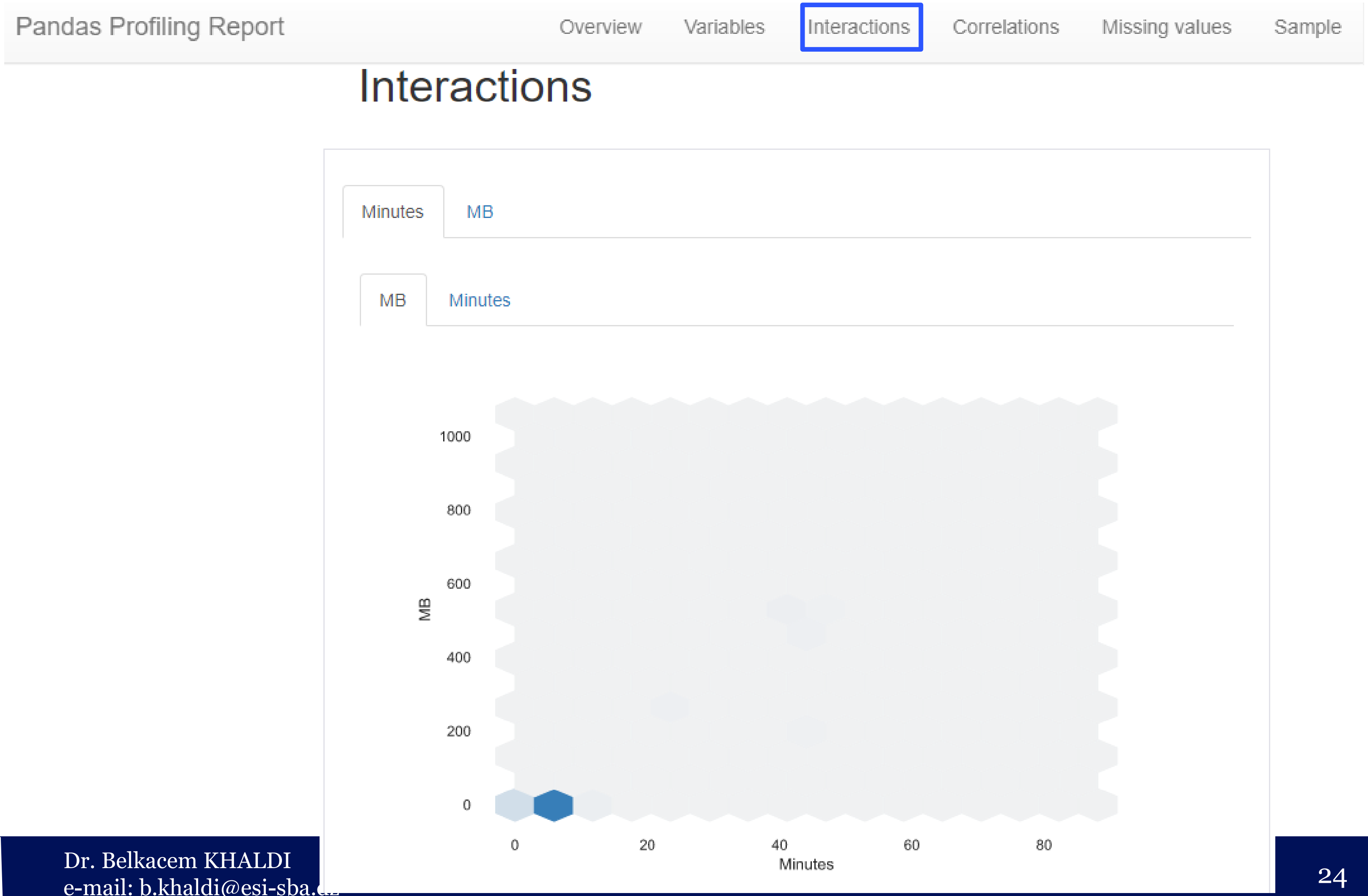
Value	Count	Frequency (%)
Lowercase Letter	36922	66.3%
Uppercase Letter	10042	18.0%
Space Separator	6967	12.5%
Other Punctuation	852	1.5%
Decimal Number	392	0.7%
Close Punctuation	191	0.3%
Open Punctuation	191	0.3%
Dash Punctuation	87	0.2%
Other Letter	5	< 0.1%

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**

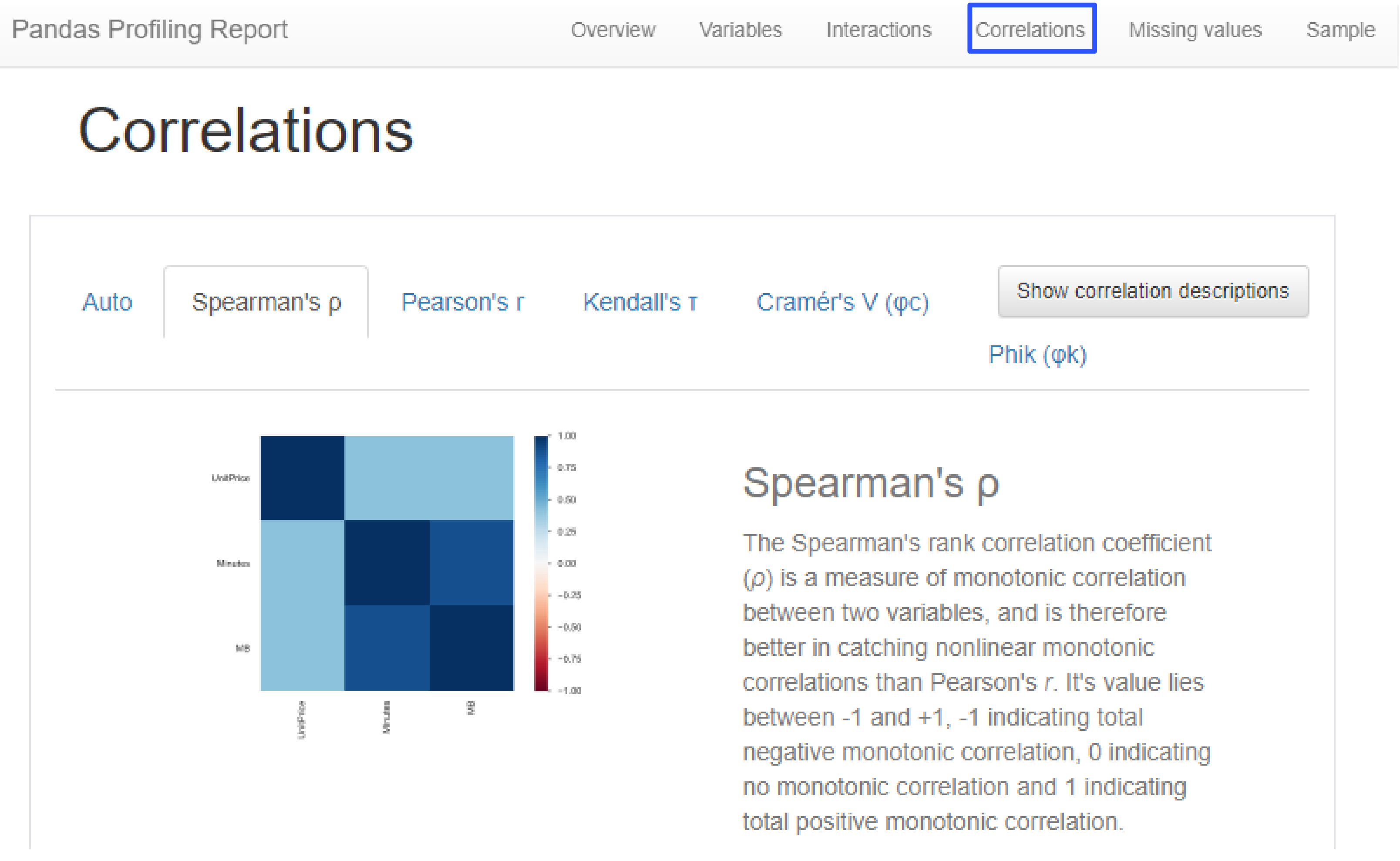


Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
    - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Correlations

Spearman's ρ

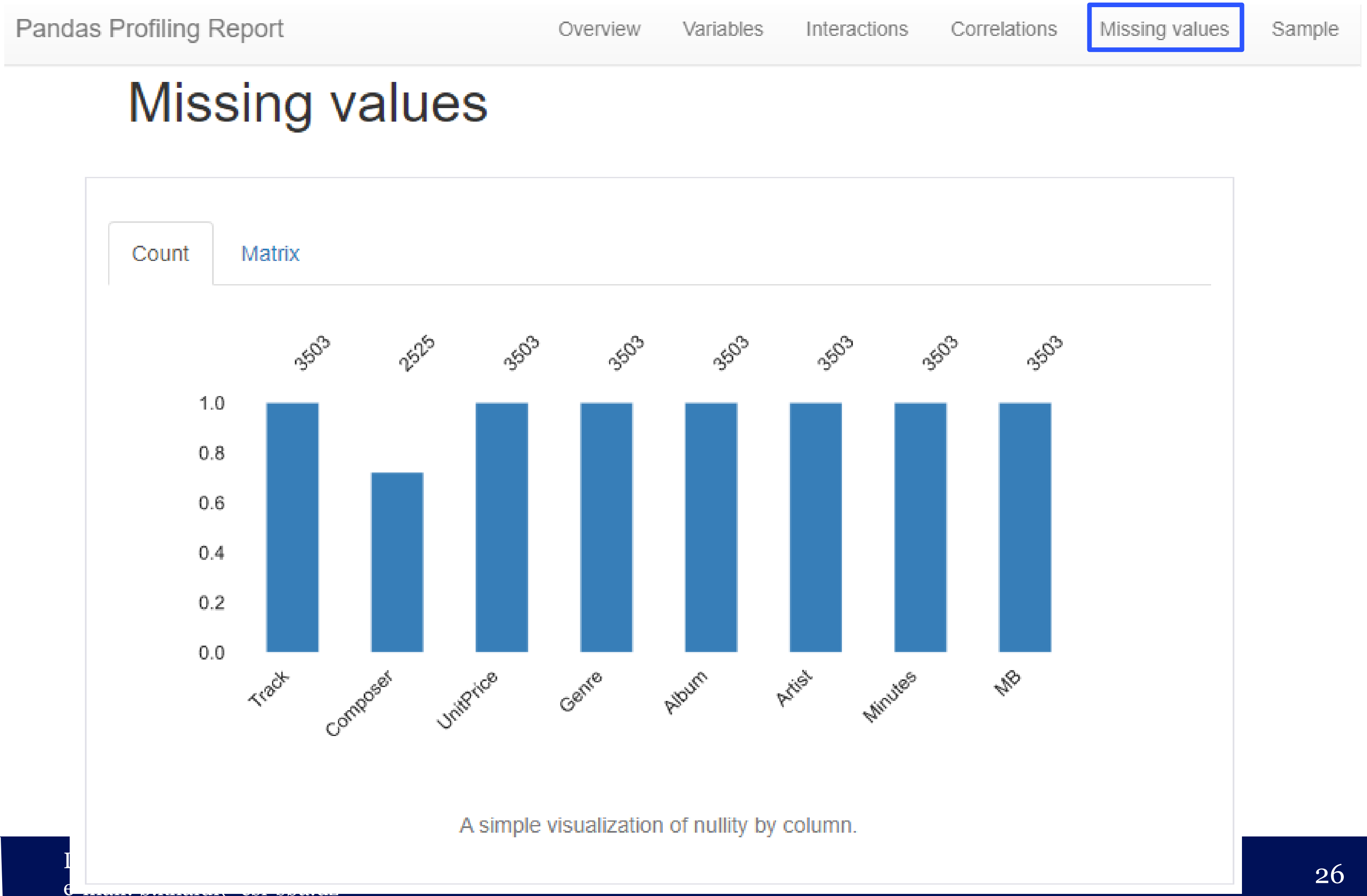
The Spearman's rank correlation coefficient (*ρ*) is a measure of monotonic correlation between two variables, and is therefore better in catching nonlinear monotonic correlations than Pearson's *r*. It's value lies between -1 and +1, -1 indicating total negative monotonic correlation, 0 indicating no monotonic correlation and 1 indicating total positive monotonic correlation.

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
    - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Missing values

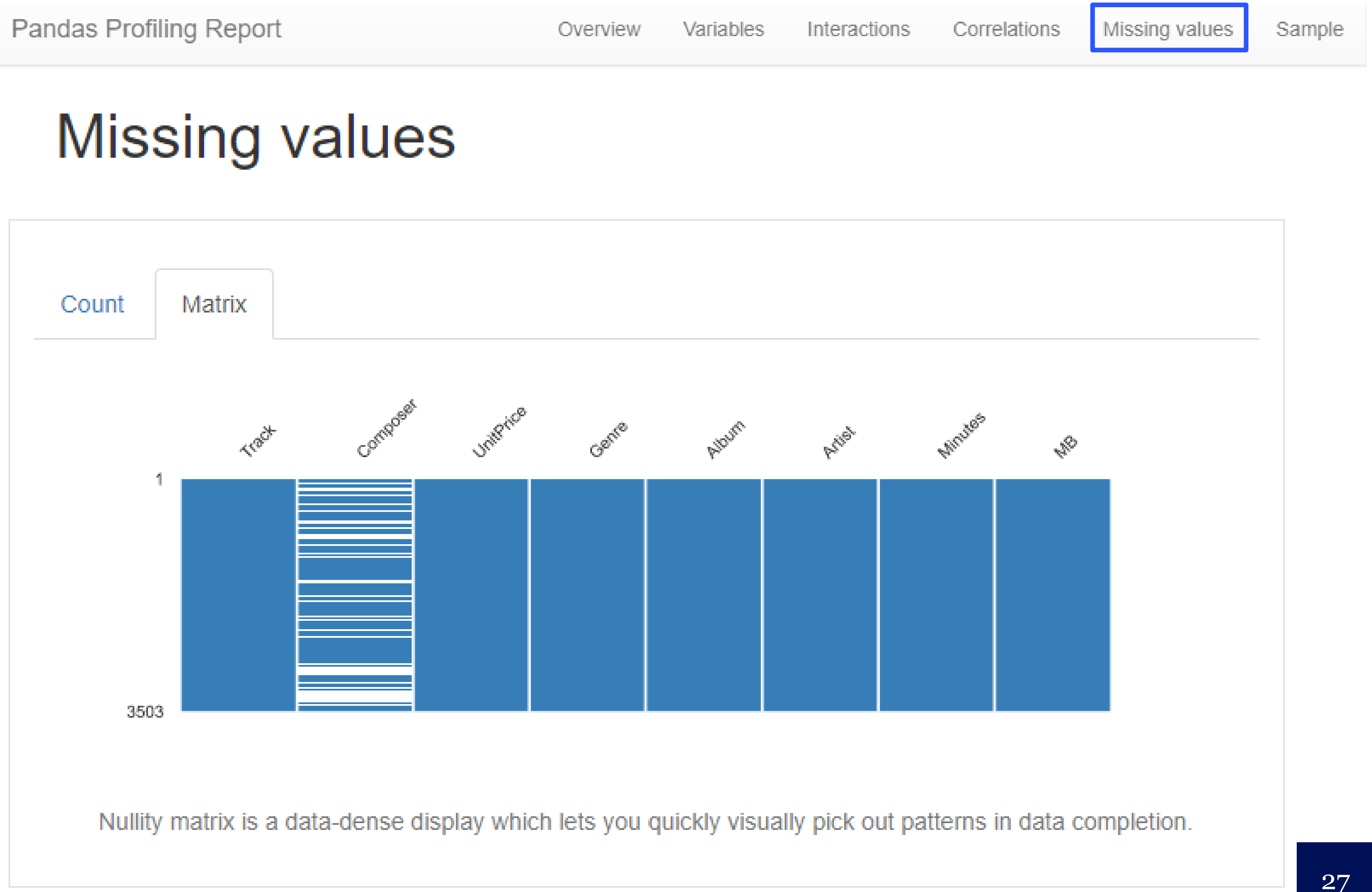
A simple visualization of nullity by column.

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
- ○ We will cover the **pandas-profiling EDA** package.
  - ○ A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Missing values

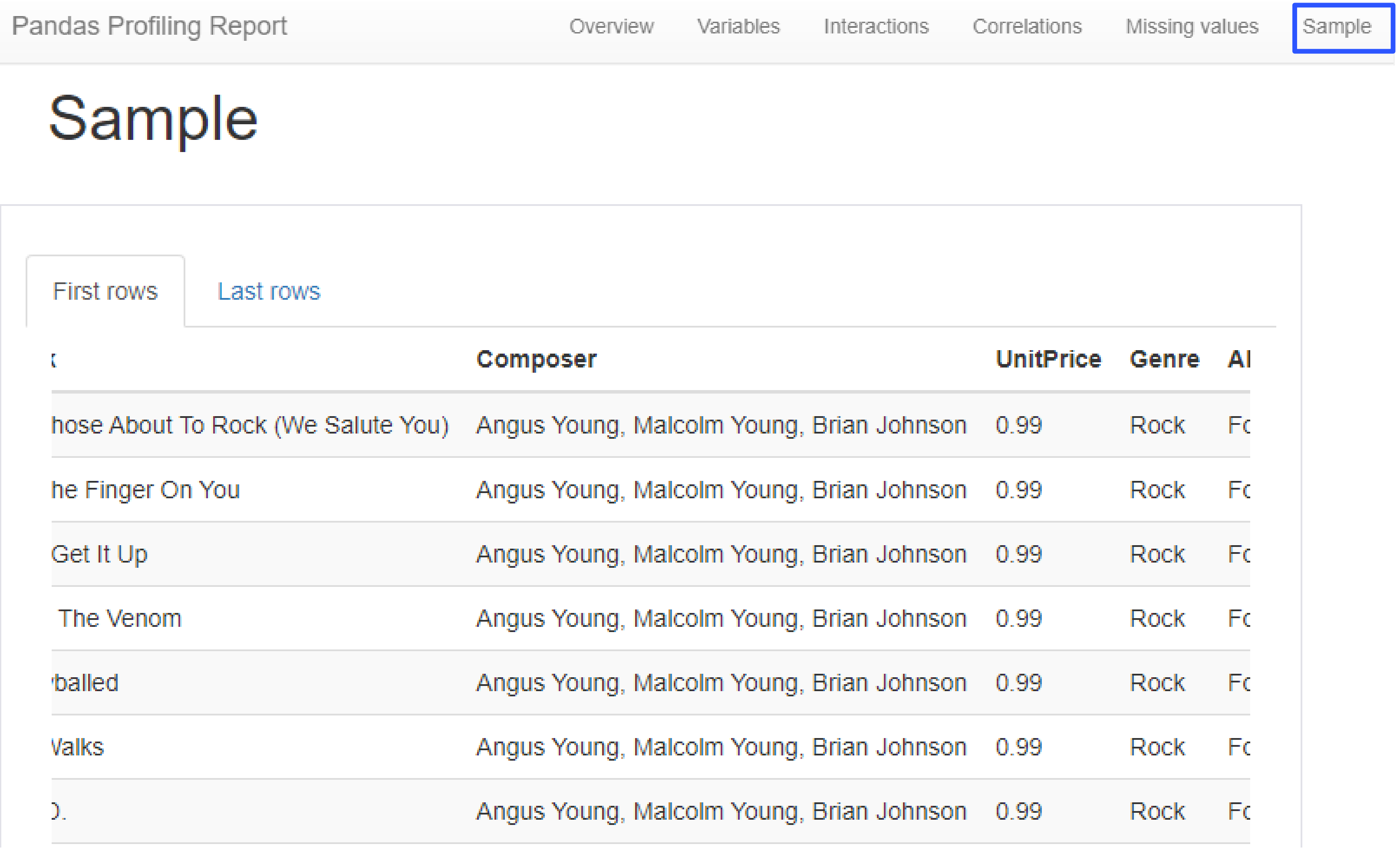
Nullity matrix is a data-dense display which lets you quickly visually pick out patterns in data completion.

Exploratory Data Analysis & Visualization

Using EDA Python packages

Making Missing Value Plots

- ❑ Sometimes it's helpful to run an auto-EDA package on the dataset.
  - We will cover the **pandas-profiling EDA** package.
    - A convenient package that creates an **EDA summary** with only a few lines of code from a pandas **DataFrame**



Sample

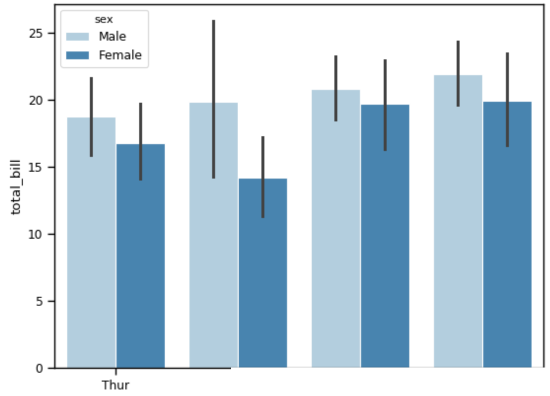
ɩ	Composer	UnitPrice	Genre	Al
hose About To Rock (We Salute You)	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
he Finger On You	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
Get It Up	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
The Venom	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
'balled	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
Valks	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc
).	Angus Young, Malcolm Young, Brian Johnson	0.99	Rock	Fc

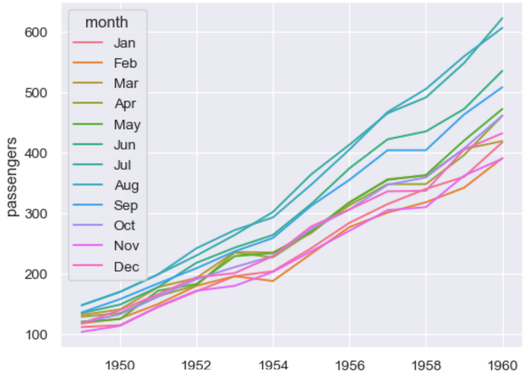
Exploratory Data Analysis & Visualization

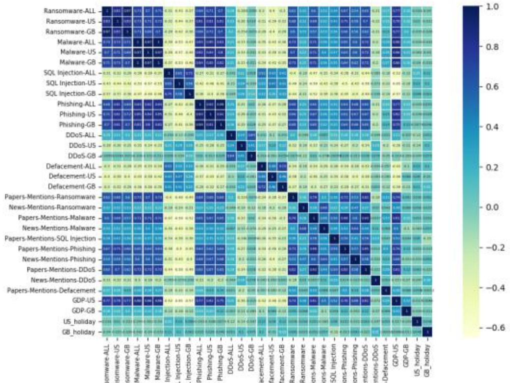
Using visualization best practices

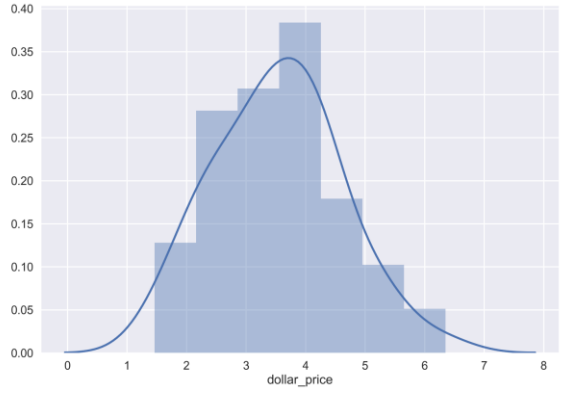
Useful tips on creating visualization

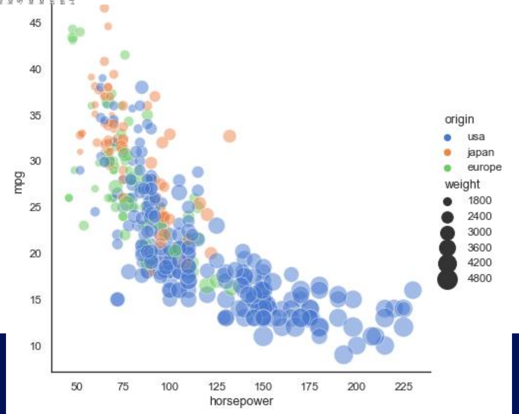
- ❑ **Bar plots** – for categorical plots
- ❑ **Histograms** – for the distribution of continuous values
- ❑ **Line charts** – for time series
- ❑ **Scatter plots** – for relationships between two continuous variables
- ❑ **Heatmaps** – for relationships between two continuous variables and correlations









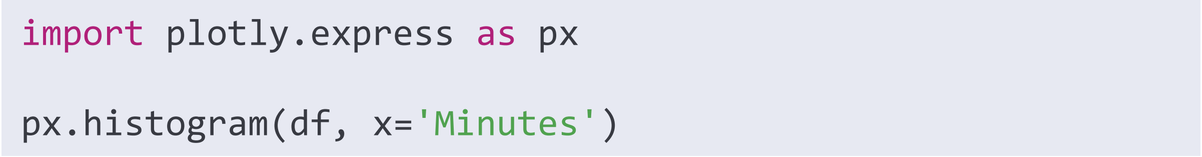


Exploratory Data Analysis & Visualization

Visualization with Plotly

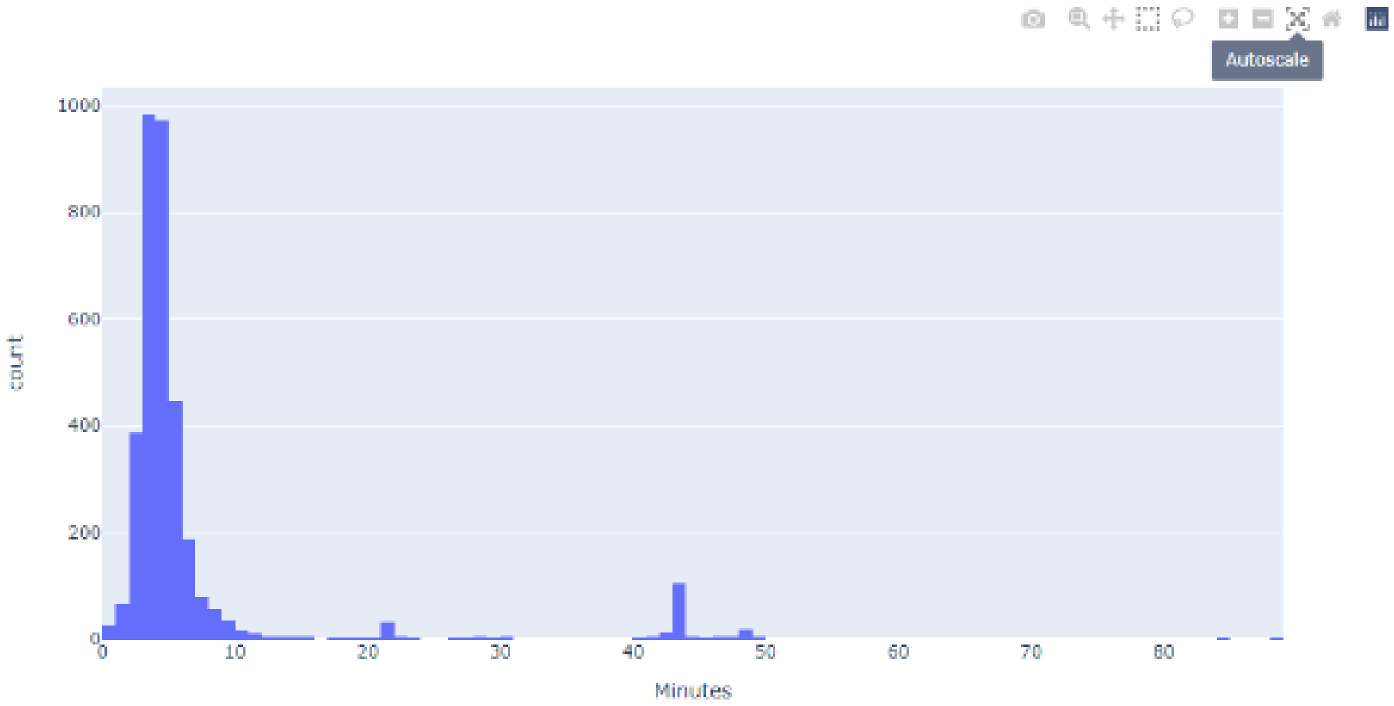
Making Histograms Plots

- ❑ **Plotly** is another visualization libraries in Python. An advantage of Plotly
- ❑ **Advantage**:
  - ○ Visualization with extra toolbar
  - ○ Visualizations can be automatically published and saved to Plotly's cloud.



```
import plotly.express as px

px.histogram(df, x='Minutes')
```



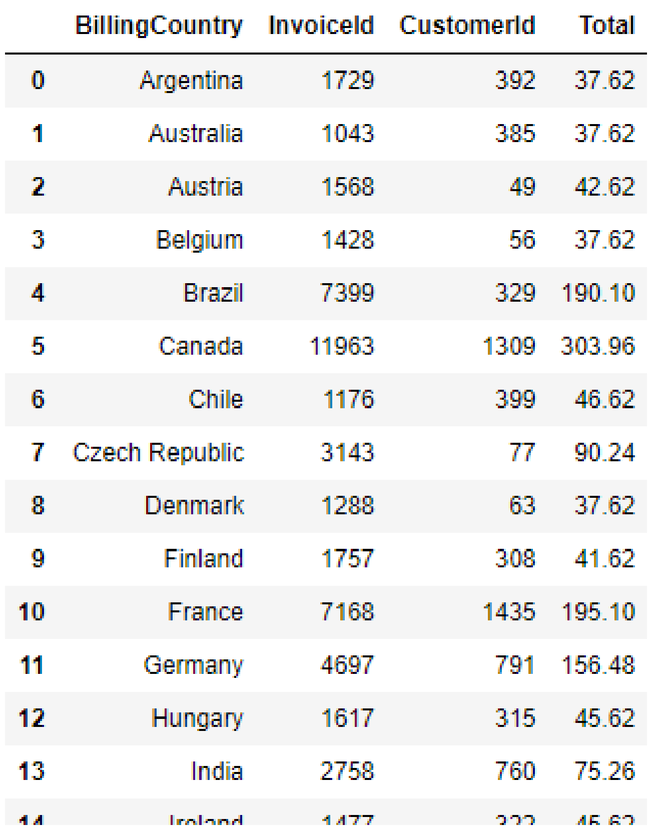
Exploratory Data Analysis & Visualization

Visualization with Plotly

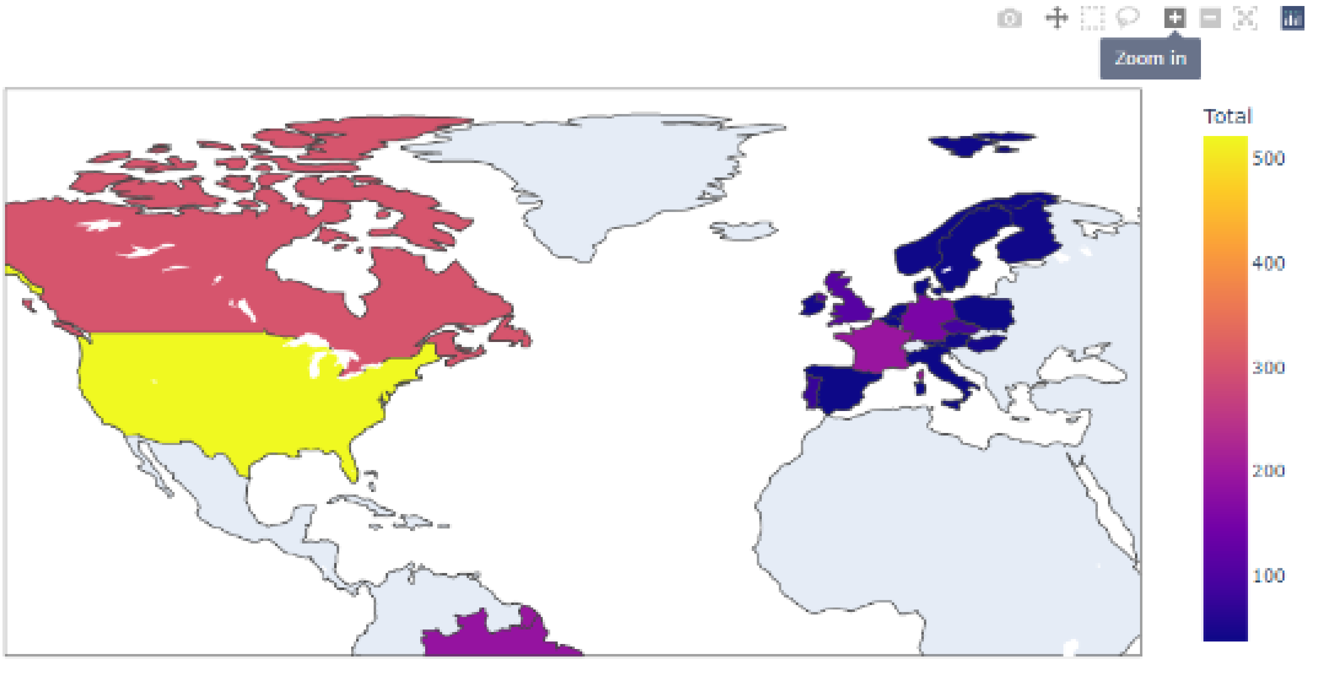
Choropleth Maps Visualization

```
px.choropleth(df_countries, locations='BillingCountry',
                 locationmode='country names', color='Total')
```

- ❑ Advanced Geographic Maps plots can be plottd with **Choropleth**
  - ○ A representation of spatial variations of a quantity,
- ❑ More interactive dashbords can be developed (See: **https://plotly.com/**



	BillingCountry	InvoiceId	CustomerId	Total
0	Argentina	1729	392	37.62
1	Australia	1043	385	37.62
2	Austria	1568	49	42.62
3	Belgium	1428	56	37.62
4	Brazil	7399	329	190.10
5	Canada	11963	1309	303.96
6	Chile	1176	399	46.62
7	Czech Republic	3143	77	90.24
8	Denmark	1288	63	37.62
9	Finland	1757	308	41.62
10	France	7168	1435	195.10
11	Germany	4697	791	156.48
12	Hungary	1617	315	45.62
13	India	2758	760	75.26
14	Ireland	1477	322	45.62



Thanks for your Listening

Ecole Supérieure en Informatique
ESI SBA
Sidi-Bel-Abbès
المدرسة العليا للإعلام الآلي
سيدي بلعباس