David Griesel

Data Analyst Portfolio

Hamburg, Germany

+49 157 534 08870

davidgrieselde@icloud.com

About Me

I'm a data analyst with a background in accounting and auditing. I started out supporting small businesses with financial reporting and analysis before moving into public sector audit, where I worked with large, complex datasets across a range of government clients.

Over time, I began enhancing audit processes through the expanded use of data analytics and computer-assisted audit techniques (CAATs), which led to a dedicated role in data analytics. There, I introduced custom scripts to streamline workflows, reduce repetitive tasks, and improve efficiency.

Today, I apply the same principles to data analytics projects — uncovering insights, solving challenges, and continuing to explore new tools and approaches.

Links

https://www.linkedin.com/in/davidgriesel

https://github.com/davidgriesel

https://public.tableau.com/app/profile/david.griesel

https://davidgriesel.com

Tools

Projects

Medical Staffing Plan

Analysed historical influenza and demographic data to develop a staffing plan for a medical staffing agency by identifying high-risk states and forecasting seasonal flu trends.

Online Grocery Store

Conducted an exploratory analysis of an online grocery store's data to gain insights into their customer base and purchasing behaviour, with the goal of optimising their marketing strategy.

Coffee Quality

Performed statistical analysis on scoring and production data to investigate the factors influencing coffee quality scores


MEDICAL STAFFING PLAN

Medical Staffing Plan

Goal

Inform the timing and spatial distribution of a medical agency's staffing plan by analysing historical influenza and demographic data

Process Flow

Defining the Business Requirements – Project Questions | Privacy & Ethics Considerations

Designing a Data Research Project – Research Hypothesis | Data Requirements

Describing the Data – Source | Collection Method | Contents | Limitations | Relevance

Data Profiling and Integrity Assessment – Data Types | Accuracy & Consistency | Cleaning | Summary Stats

Data Quality Assessment – Completeness | Uniqueness | Timeliness | Cleaning

Data Transformation & Integration

Statistical Analysis – Variance | Standard Deviation | Relationships | Correlation

Statistical Hypothesis Testing

Compiling an Interim Report

Data Visualisation & Spatial Analysis

Writing a Narrative & Creating an Interactive Storyboard

Tools

– Excel

– Tableau

Data

Number of influenza-related deaths – Number of deaths by location, time, and age. (CDC)

Influenza vaccination rates in children – Number of vaccinations by location, time, and age. (CDC)

Population – Number of people by location, time, age, and gender. (US Census Bureau)

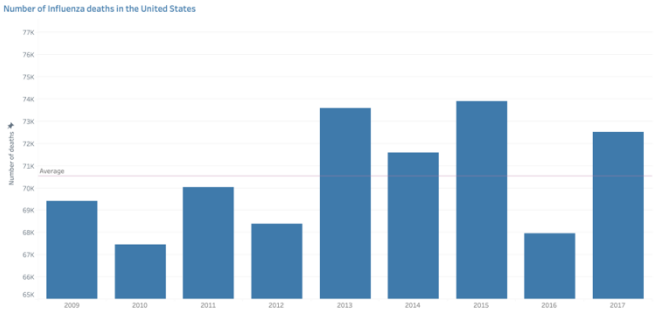
Links

GitHub Repository

Tableau Storyboard

Analysis & Insights

Spatial Distribution & Risk Classification



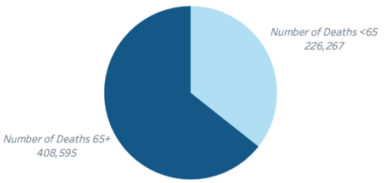
Dataset	Integrated Data Set	
Sample or Population?	Sample	> 30 - Central Limit Theorem applies

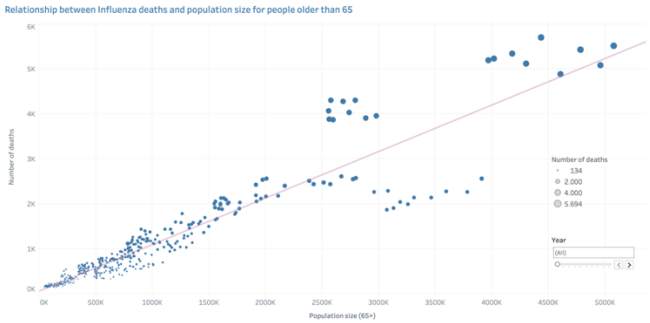
Variable	Population 65+	Influenza Deaths 65+
Number of Records	459	459
Variance	786799847780,66	651954,62
Standard Deviation	887017,39	807,44
Mean	806989,68	1263,72
-2SD	-967045,10	-351,15
+2SD	2581024,46	2878,60
Outliers below -2SD	0	0
Outliers above +2SD	29	18
Outlier Percentage (Empirical Rule)	6%	4%

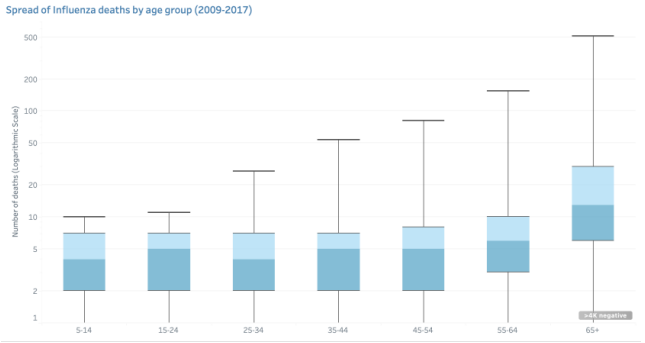
Pearson's Correlation Coefficient	0,84
Relationship	Strong Relationship
Interpretation	The more people there are aged 65 and older, the higher the number of influenza-related deaths will be

Statistical Hypothesis Testing	
Research hypothesis	Individuals aged 65 and older have a higher influenza-related death rate than
Independent Variable	Population (65+ vs. <65)
Dependent Variable	Death Rate
H_0 (Null)	Death rate for people 65+ is ≤ death rate for people under 65
H_1 (Alternative)	Death rate for people 65+ is > death rate for people under 65
Type of Test	One-Tailed Test (Null hypothesis - > equal to, or lower than / one direction of
Significance Level (α)	0,0500
p-value	0,0000
Assessment	The p-value of 0.00 is below the significance level of 0.05, indicating a statistically

t-Test: Two-Sample Assuming Unequal Variances	Death Rate 0-64	Death Rate 65+
Mean	0,00059714382	0,00395811359
Variance	0,00000040031	0,00002543436
Observations	459	459
Hypothesized Mean Difference	0	
df	472	
t Stat	-14,16672411596	
P(T<=t) one-tail	0,00000000000	
t Critical one-tail	1,64808833557	
P(T<=t) two-tail	0,00000000000	
t Critical two-tail	1,96500267646	

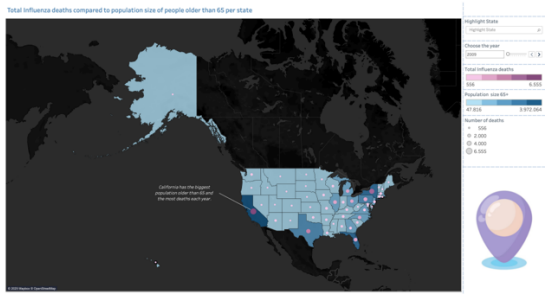


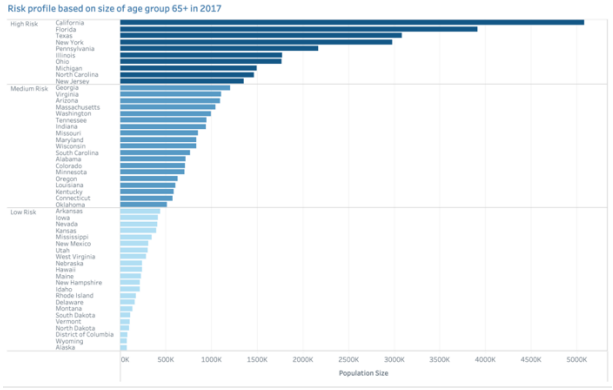


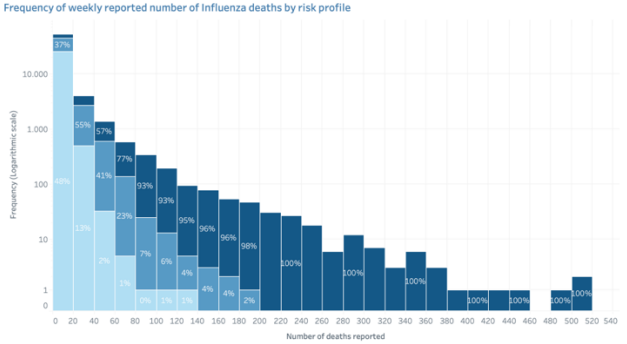


Analysis & Insights

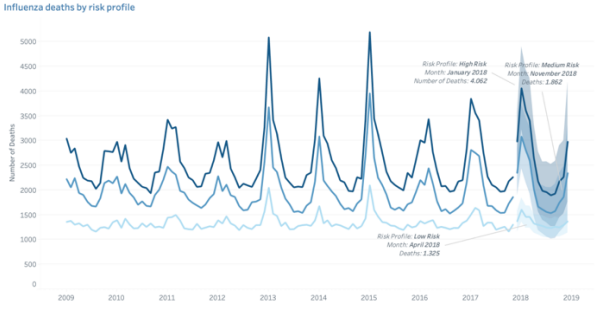
Spatial Distribution & Risk Classification

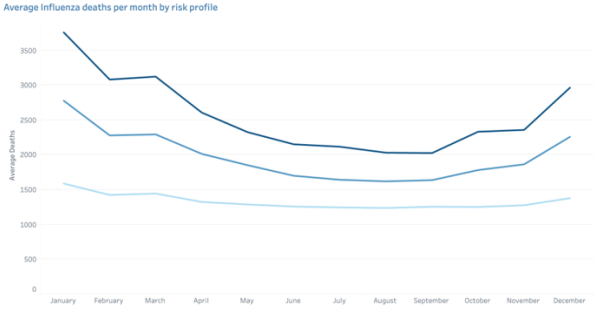


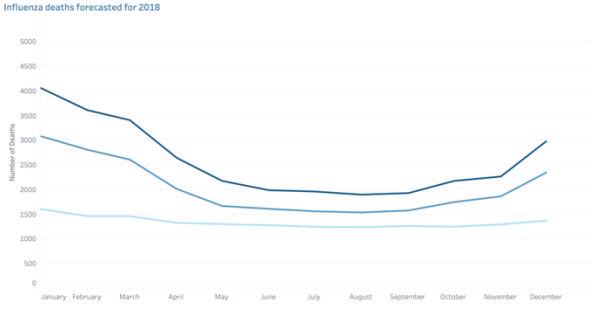




Seasonality & Forecasting







Recommendations

It is recommended that staffing resources be prioritised for high-risk states, with allocations scaled according to the size of their elderly populations. Additionally, deployment should begin ahead of the seasonal peak — ideally starting in November — to ensure adequate coverage throughout the critical period from December through April. This approach supports a targeted and data-informed strategy to minimise influenza-related mortality and optimise medical staffing efficiency.

Reflection

Successes

The project provided actionable insights into the spatial distribution of vulnerable populations, enabling the development of a risk profile to guide regiona staff allocations across states. Additionally, it offered valuable understanding of influenza seasonality, which supported the strategic timing of staff deployments to meet anticipated healthcare demands.

Challenges

The available data lacked sufficient detail to account for all known factors in the analysis. The suppression of all records for children under 5 limited the ability to assess risk in this age group, highlighting how data privacy lawscan restrict public health analyses when key demographic groups are excluded.

Moving Forward

Monitor the impact of the 2018 deployment strategy to evaluate its effectiveness and inform the next planning cycle, ensuring continuous improvement in staff allocation strategies.


Online
Shopping

Online Grocery Store

Goal

Perform an exploratory analysis of customer demographics and transactional data to gain insights into ordering behaviours, thereby enhancing customer profiling and optimising marketing strategies through targeted product placement.

Process Flow

Conducted Descriptive Exploratory Data Analysis

Data Wrangling – Data Types I Accuracy I Drop & Rename Columns

Data Consistency Checks – Missing Values I Duplicates I Mixed-Type variables

Merging Dataframes

Subsetting & Deriving New Variables

Grouping & Aggregating Data

Data Visualisations & Analysis

Compiling and Interim Report

Tools

– Python

– Excel

Data

The Instacart Online Grocery Shopping Dataset – Product, department, and order details. (Instacart)

Supplementary Customer dataset – Customer details and demographic information. (CareerFoundry)

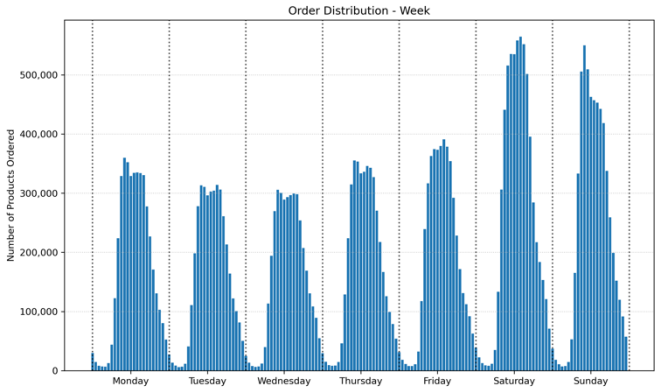
Links

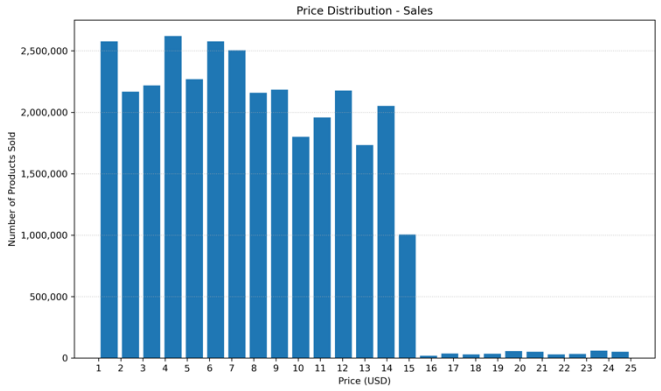
Kaggle

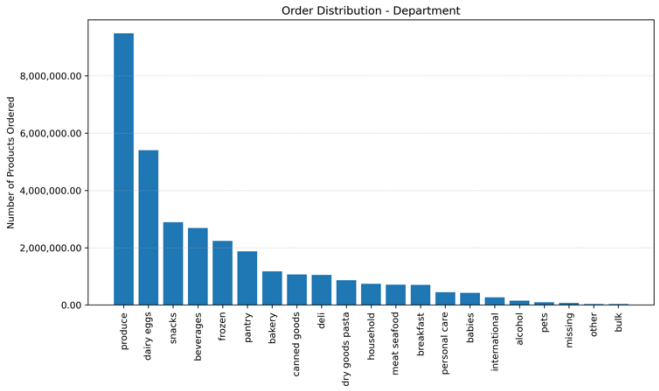
GitHub Repository

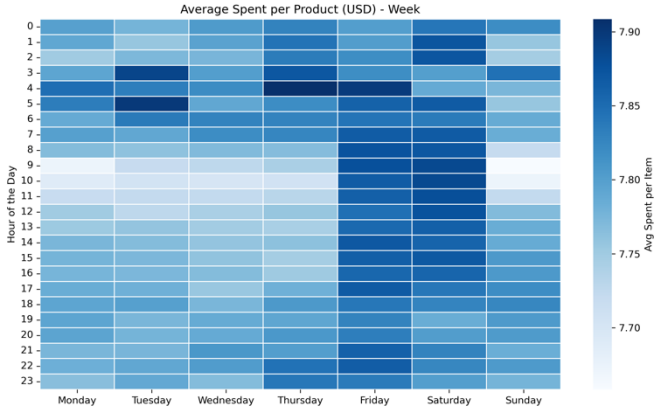
Analysis & Insights

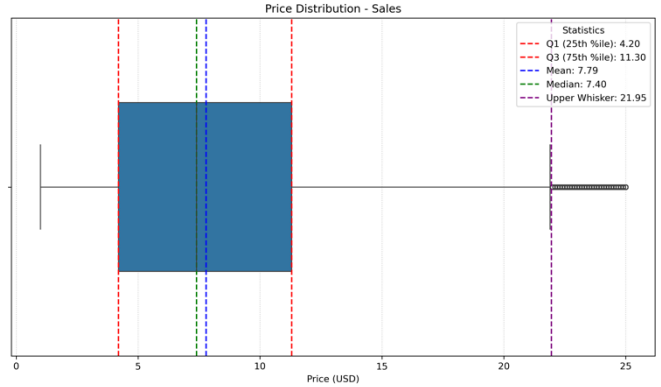
Ordering Trends

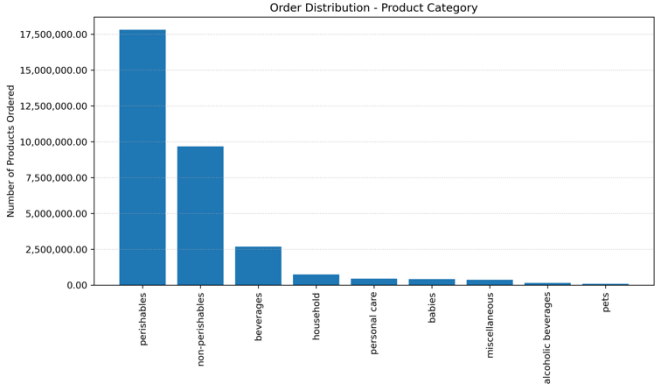






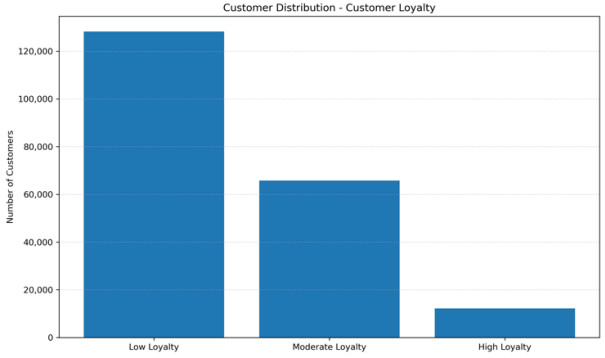


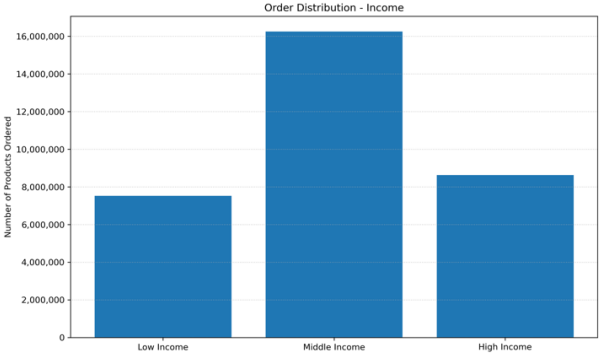


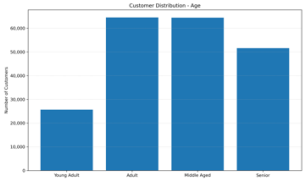


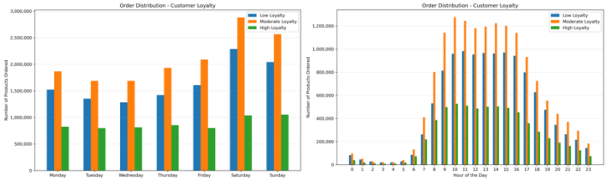
Analysis & Insights

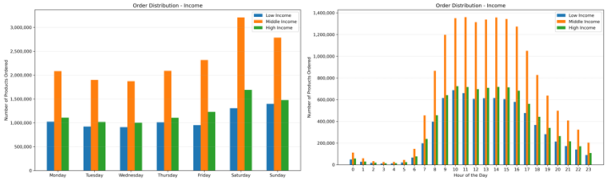
Ordering Trends

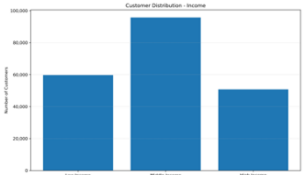


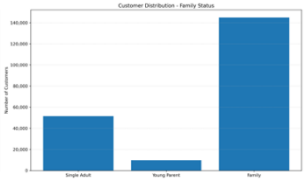


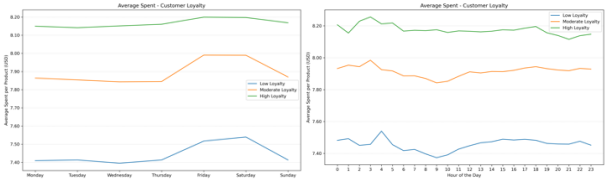


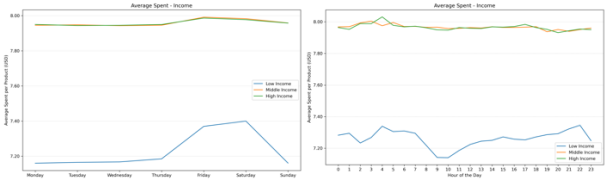


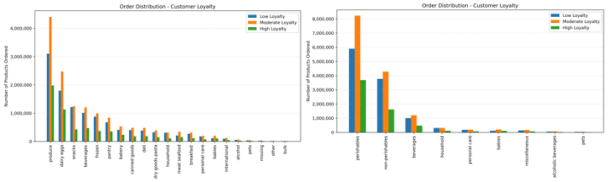


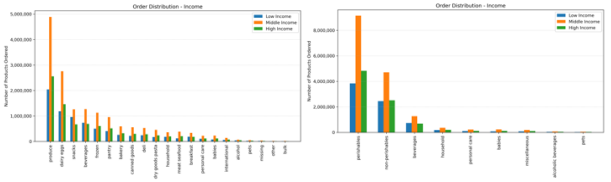












Recommendations

To optimise engagement and revenue, ads should run during low-order periods — specifically between 4 p.m. and 7 a.m. Product promotions should align with spending behaviour: mid-priced items in the evening, high-priced products in the early morning and late on Thursday and Friday evenings, and premium items between 3–6 a.m., when spending peaks. Focus should be placed on food and beverage categories, especially fresh produce, dairy & bakery items.

Targeting should also consider customer profiles:

- Moderate loyalty customers respond well to a mix of staple and premium products.
- High loyalty customers prefer healthier, premium options.
- Low loyalty customers are more price sensitive and respond to budget-friendly goods.

Income segmentation can further refine targeting, with premium products appealing to high-income shoppers and essentials resonating more with low-income groups.

Reflection

Successes

The project provided actionable insights into ordering behaviour across customer profiles, enabling a more targeted marketing strategy through better product placement.

Challenges

Generating visualisations in Python is cumbersome and creating clear and intuitive charts reguired significant customisation.

Moving Forward

Create an interactive Tableau storyboard to visually present key findings, enabling stakeholders to explore ordering trends, customer profiles, and spending behaviours with filters, dahsboards, and drill-down insights.

Coffee Quality

Goal

Analyse coffee quality across countries, explore predictive relationships between quality measures, and examine the link between quality scores and demand.

Process Flow

Define Project Requirements – Define Project Deliverables & Data Requirements

Data Sourcing & Preparation – Source Prepare Open Source Data

Conduct Exploratory Data Analysis – Visualising Relationships & Developing a Hypothesis

Geospatial Analysis – Integrate & Analyse Spatial Data with Choropleth Maps using Shapefiles

Supervised Machine Learning – Apply Regression Techniques for Predictive Analysis

Unsupervised Machine Learning – Apply k-means Clustering using Elbow Technique for pattern discovery

Time Series Analysis – Analysing Temporal Data through Decomposition, Stationary Tests, and Transformations

Data Visualisation & Storytelling – Communicate Insights throug Interactive Dashboard.

Tools

– Python

– Tableau

Data

Coffee Quality with Locations of Origin – Quality scores of coffee beans by country. (Coffee Quality Institute)

Average Production Per Year – Consumption, import, export, and production metrics over time. (Coffee Quality Institute)

Links

Kaggle

GitHub Repository

Tableau Storyboard

Analysis & Insights

Ordering Trends

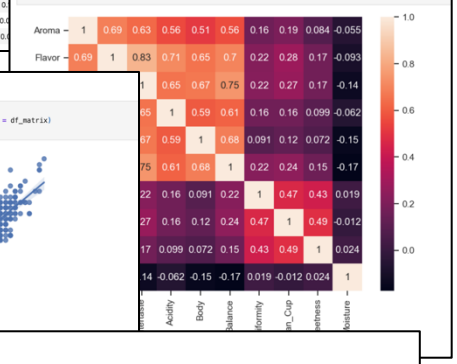
Correlation Matrix

```
# Drop categorical columns
df_matrix = df_quality.drop(columns = ['Species', 'Continent_of_Origin', 'Country_of_Origin', 'Harvest_Year',
       'Expiration', 'Variety', 'Color', 'Processing_Method','Quakers', 'Category_One_Defects',
       'Category_Two_Defects'])

# Correlation matrix
df_matrix.corr()
```

	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	Uniformity	Clean_Cup	Sweetness	Moisture
Aroma	1.000000	0.687843	0.631839	0.558277	0.511892	0.563619	0.157327	0.193493	0.083830	-0.055065
Flavor	0.687843	1.000000	0.831670	0.713368	0.654850	0.700998	0.218916	0.222753	0.106326	-0.093089
Aftertaste	0.631839	0.831670	1.000000	0.654850	0.668807	0.746396	0.222753	0.274055	0.106326	-0.136153
Acidity	0.558277	0.713368	0.654850	1.000000	0.652239	0.746396	0.222753	0.274055	0.106326	-0.136153
Body	0.511892	0.653870	0.668807	0.652239	1.000000	0.746396	0.222753	0.274055	0.106326	-0.136153
Balance	0.563619	0.700998	0.746396	0.746396	0.746396	1.000000	0.222753	0.274055	0.106326	-0.136153
Uniformity	0.157327	0.218916	0.222753	0.222753	0.222753	0.222753	1.000000	0.274055	0.106326	-0.136153
Clean_Cup	0.193493	0.282430	0.274055	0.274055	0.274055	0.274055	0.274055	1.000000	0.106326	-0.136153
Sweetness	0.083830	0.166396	0.166022	0.106326	0.106326	0.106326	0.106326	0.106326	1.000000	-0.136153
Moisture	-0.055065	-0.093089	-0.136153	-0.136153	-0.136153	-0.136153	-0.136153	-0.136153	-0.136153	1.000000

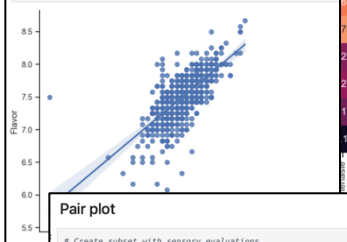
HeatMap

```
# Subplot
f,ax = plt.subplots(figsize = (8,6))

# Correlation heatmap
corr = sns.heatmap(df_matrix.corr(), annot = True, ax = ax)
plt.show()
```



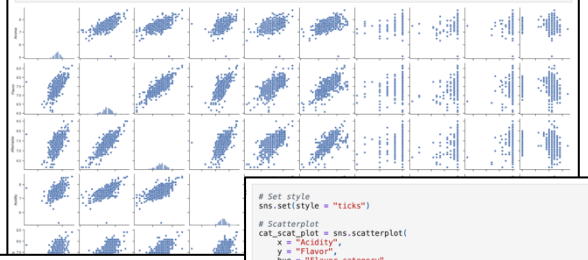
Scatterplots

```
# Acidity & Flavor - 0.71
sns.lmplot(x = 'Acidity', y = 'Flavor', data = df_matrix)
plt.show()
```



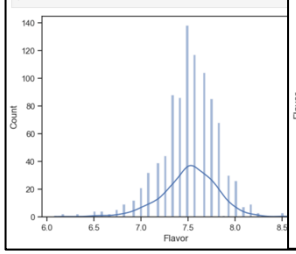
Pair plot

```
# Create subset with sensory evaluations
df_pair_plot = df_quality[['Aroma', 'Flavor', 'Aftertaste', 'Acidity', 'Body',
                           'Balance','Uniformity', 'Clean_Cup', 'Sweetness', 'Moisture']]

# Pair plot
pair_plot = sns.pairplot(df_pair_plot)
plt.show()
```

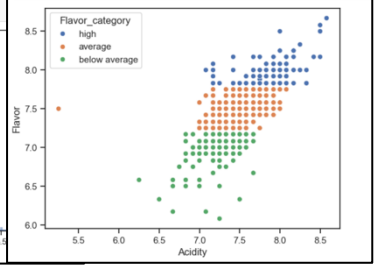


Categorical plots

```
# Histogram
sns.histplot(df_quality['Flavor'], bins = 100, kde = True)
plt.show()
```




```
# Set style
sns.set(style = "ticks")

# Scatterplot
cat_scat_plot = sns.scatterplot(
    x = "Acidity",
    y = "Flavor",
    hue = "Flavor_category",
    data = df_categorical
)

plt.show()
```



GeoJSON file

```
# Load the JSON file
with open(country_geo, 'r') as file: #Consulted ChatGPT
    data = json.load(file)

# View the data
print(data) #Consulted ChatGPT
```


{'type': 'FeatureCollection', 'features': [{'type': 'Feature', 'properties': {'name': 'Afghanistan'}, 'geometry':
{'type': 'Polygon', 'coordinates': [[[61.210817, 35.650072], [62.230651, 35.270664], [62.984662, 35.404041], [63.1
93538, 35.857166], [63.982896, 36.007957], [64.546479, 36.312073], [64.746105, 37.111818], [65.588948, 37.305217],
[65.745631, 37.661164], [66.217385, 37.39
[68.135562, 37.023115], [68.859446, 37.344
223], [70.270574, 37.735165], [70.376304,
37.953265], [71.541918, 37.905774], [71.44
609, 37.047558], [73.260056, 37.495257], [
4.575893, 37.020841], [74.067552, 36.836176
8], [71.498768, 35.650563], [71.613076, 35
3.988856], [69.930543, 34.020122], [70.3235
64, 31.901412], [68.926677, 31.620189], [6
938891, 31.304911], [66.381458, 30.738899]
11, [64.148002, 29.340819], [63.550261, 29
0.73585], [61.699314, 31.379506], [60.9419
33.528832], [60.52843, 33.676446], [60.803
ure', 'properties': {'name': 'Angola'}, 'g
71, [16.57318, -6.622645], [16.860191, -7.
87678], [18.464176, -7.847014], [19.016752
3, -7.116361], [20.091622, -6.943091], [20.
746456, -7.920085], [21.949131, -8.305901]
61, [22.155268, -11.084001], [22.402798, -
5, -10.926826], [24.017894, -11.239298], [
8], [24.016137, -12.911046], [21.933886, -
-17.523116], [21.377176, -17.930636], [18.
14.058501, -17.423381], [13.462362, -16.97
7.301889], [11.640096, -16.673142], [11.77
2.500095, -13.5477], [12.738479, -13.13798
863], [13.686379, -10.731076], [13.387328,
-8.959091], [13.236433, -8.562629], [12.93
432, -6.100092], [12.735171, -5.965682], [
[[12.436688, -5.684304], [12.182337, -5.7
923], [12.995517, -4.781103], [12.631612,

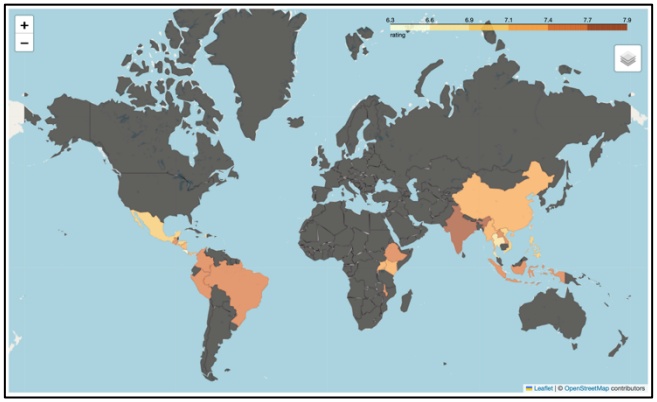
Choropleth

Aroma

```
# Dataframe for plot
plot_data_aroma = df_quality_v1[['Country_of_Origin','Aroma']]

# Preview
plot_data_aroma.head()
```

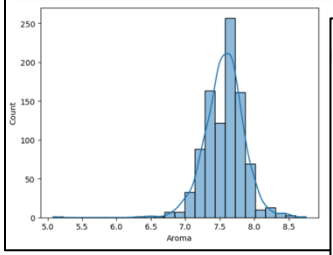
	Country_of_Origin	Aroma
1	Ethiopia	8.75
4	Ethiopia	8.25
6	Peru	8.42
9	Ethiopia	8.08
18	China	8.42

```
# Setup a folium map at a high-level zoom
map = folium.Map(location = [100, 0], zoom_start = 1.5)

# Choropleth maps bind Pandas Data Frames and json geometries.This allows us to quickly visualize data combinati
folium.Choropleth(
    geo_data = country_geo,
    data = plot_data_aroma,
    columns = ['Country_of_Origin', 'Aroma'],
    key_on = 'feature.properties.name', # this part is very important - check your json file to see where the KE
    fill_color = 'YlOrBr', fill_opacity=0.6, line_opacity=0.1,
    legend_name = "rating").add_to(map)
folium.LayerControl().add_to(map)
map
```



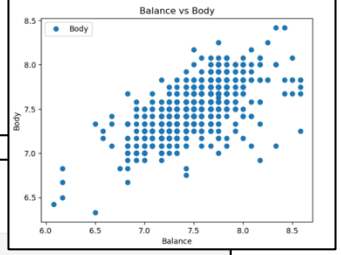
Extreme value checks - Aroma

```
# Histogram - Aroma
sns.histplot(df_quality['Aroma'], bins = 25, kde = True)
plt.show()
```



Balance vs Body

```
# Scatterplot
df_quality.plot(x = 'Balance', y = 'Body',style = 'o')
plt.title('Balance vs Body')
plt.xlabel('Balance')
plt.ylabel('Body')
plt.show()
```



Hypothesis #1

Reshape variables

```
# Reshape variables
X1 = df_quality['Aftertaste'].values.reshape(-1,1)
y1 = df_quality['Flavor'].values.reshape(-1,1)
```

Split data

```
# Split data
X1_train, X1_test, y1_train, y1_test =
```

Linear regression

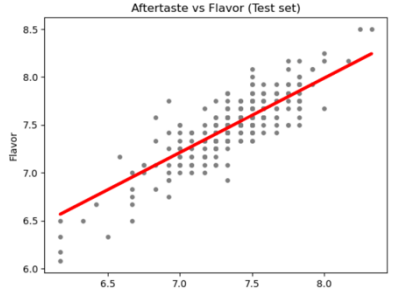
```
# Create regression object
regression_1 = LinearRegression()

# Fit the regression object onto the t
regression_1.fit(X1_train, y1_train)
```

```
LinearRegression()
```

```
# Predict the values of y1 using X1
y1_predicted = regression_1.predict(X1
```

Regression plot

```
# Plot the modelled regression - test set
plot_test_1 = plt
plot_test_1.scatter(X1_test, y1_test, color = 'gray', s = 15)
plot_test_1.plot(X1_test, y1_predicted, color = 'red', linewidth = 3)
plot_test_1.title('Aftertaste vs Flavor (Test set)')
plot_test_1.xlabel('Aftertaste')
plot_test_1.ylabel('Flavor')
plot_test_1.show()
```



Hypotheses

The following variables appear most frequ

- Aftertaste
- Flavor
- Balance

Hypothesis #1

- The higher the score for Aftertaste, the

Model performance statistics (Test set)

```
# Create objects that contain the model summary stat
rmse_1 = mean_squared_error(y1_test, y1_predicted)
r2_1 = r2_score(y1_test, y1_predicted)

# Print the model summary statistics
print('Slope:' ,regression_1.coef_)
print('Mean squared error: ', rmse_1)
print('R2 score: ', r2_1)
```

```
Slope: [[0.77654044]]
Mean squared error:  0.031563293173207344
R2 score:  0.7305230539312092
```

Model performance statistics (Training set)

```
# Predict
y1_predicted_train = regression_1.predict(X1_train)

# Create objects that contain the model summary statistics
rmse_1 = mean_squared_error(y1_train, y1_predicted_train)
r2_1 = r2_score(y1_train, y1_predicted_train)

# Print the model summary statistics
print('Slope:' ,regression_1.coef_)
print('Mean squared error: ', rmse_1)
print('R2 score: ', r2_1)
```

```
Slope: [[0.77654044]]
Mean squared error:  0.029287772376638082
R2 score:  0.6660234980676127
```

Analysis & Insights

Ordering Trends

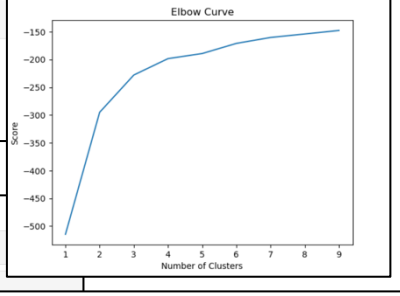
Elbow technique

```
# Defines the range
num_cl = range(1, 10)

# Defines k-means clusters
kmeans = [KMeans(n_clusters = i) for i in num_cl]

# Create scores
score = [kmeans[i].fit(df_matrix).score(df_matrix) for i in range(len(kme

# View scores
score
```

```
[-514.7984129180053,
 -295.17284742456613,
 -228.00142912404007,
 -198.50689513746968,
 -189.33859850871 7,
 -171.09110326131645,
 -160.4211309152332,
 -154.13947772845893,
 -147.78222725919875]
```

Elbow Curve

```
# Plot the elbow curve
pl.plot(num_cl,score)
pl.xlabel('Number of Clusters')
pl.ylabel('Score')
pl.title('Elbow Curve')
pl.show()
```



k-means clustering

```
# Create the k-means object
kmeans = KMeans(n_clusters = 3)

# Fit the object
kmeans.fit(df_matrix)
```

```
KMeans(n_clusters=3)
In a Jupyter environment, please rerun this cell to show the HTML r
On GitHub, the HTML representation is unable to render, please try
```

New column

```
# Create 'clusters' variable
df_matrix['clusters'] = kmeans.fit_predict(df_matrix)

# Preview
df_matrix.head()
```

	Aroma	Flavor	Aftertaste	Acidity	Body	Balance	clusters
1	8.75	8.67	8.50	8.58	8.42	8.42	1
4	8.25	8.50	8.25	8.50	8.42	8.33	1
6	8.42	8.50	8.33	8.50	8.25	8.25	1
9	8.08	8.58	8.50	8.50	7.67	8.42	1
18	8.42	8.25	8.08	8.17	7.92	8.00	1

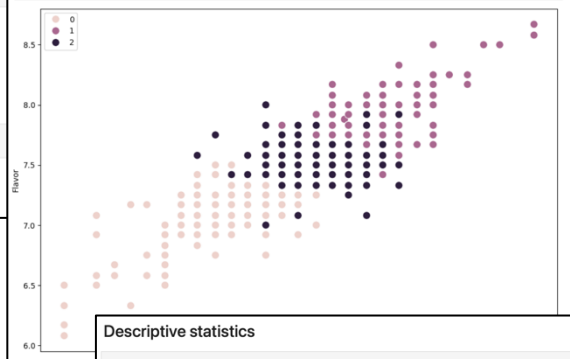
```
# Frequency of 'clusters' variable
df_matrix['clusters'].value_counts()
```

```
clusters
2    468
1    200
0    197
Name: count, dtype: int64
```

Hypothesis #1

- The higher the score for Aftertaste, the higher the score for Flavor would be.

```
# Cluster plot - "Aftertaste" and "Flavor"
plt.figure(figsize = (12,8))
ax = sns.scatterplot(x = df_matrix['Aftertaste'], y = df_matrix['Flavor'], hue = kmeans.labels_, s = 100)

ax.grid(False)
plt.xlabel('Aftertaste')
plt.ylabel('Flavor')
plt.show()
```



Descriptive statistics

```
# Assign corresponding colour to the clusters
df_matrix.loc[df_matrix['clusters'] == 2, 'cluster_color'] = 'dark purple'
df_matrix.loc[df_matrix['clusters'] == 1, 'cluster_color'] = 'purple'
df_matrix.loc[df_matrix['clusters'] == 0, 'cluster_color'] = 'pink'

# Calculate statistics
df_matrix.groupby('cluster_color').agg({'Aroma':['mean', 'median'],
    'Flavor':['mean', 'median'],
    'Aftertaste':['mean', 'median'],
    'Acidity':['mean', 'median'],
    'Body':['mean', 'median'],
    'Balance':['mean', 'median']})
```

	Aroma		Flavor		Aftertaste		Acidity		Body		Balance	
	mean	median	mean	median	mean	median	mean	median	mean	median	mean	median
cluster_color												
dark purple	7.546880	7.58	7.516453	7.50	7.393654	7.42	7.501688	7.50	7.480769	7.50	7.498333	7.50
pink	7.257665	7.25	7.097411	7.17	6.970457	7.00	7.210406	7.25	7.228071	7.25	7.097817	7.17
purple	7.811464	7.75	7.802107	7.75	7.688571	7.67	7.804143	7.75	7.751464	7.75	7.817679	7.75

Group and aggregate

```
# Group by 'Year' and calculate the mean production
df_production_4 = df_production_3.groupby('Year')['Production'].mean().reset_index()

# Rename column
df_production_4.rename(columns = {'Production': 'Average Production'}, inplace = True)

# Display the result
df_production_4
```

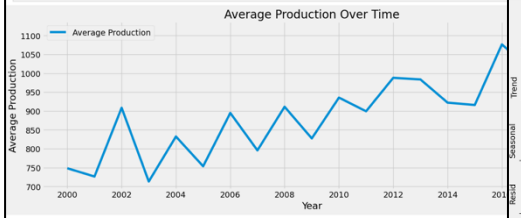
Indexing

```
# Set 'Date' as datetime index

# Copy dataframe
df_production_5 = df_production_4.copy()

from datetime import datetime

df_production_5['datetime'] = pd.to_datetime(df_production_5['Year'], format='%Y')
df_production_5 = df_production_5.set_index('datetime')
df_production_5.drop(['Year'], axis = 1, inplace = True)
```

Analysis

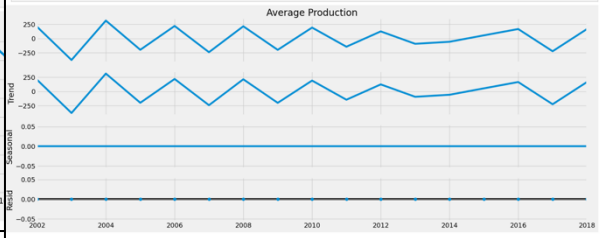
Line chart

```
plt.figure(figsize = (15, 5), dpi = 100)

# Specify the column to plot
plt.plot(df_production_5.index, df_production_5['Average Production'], label = 'Average Productio

# Add labels and title
plt.title('Average Production Over Time')
plt.xlabel('Year')
plt.ylabel('Average Production')
plt.legend()

# Show the plot
plt.show()
```



Decomposition - Round 2

```
from pylab import rcParams

# Decompose time series - additive model
decomposition = sm.tsa.seasonal_decompose(df_production_7['Average Production'], model = 'additive')

# Set the figure size
rcParams['figure.figsize'] = 18, 7

# Plot the decomposition
decomposition.plot()
plt.show()
```



Comments

- The data does not exhibit seasonality or significant noise, and the upward trend over time has been smoothed out.

Dicky-Fuller test - Round 2

```
from statsmodels.tsa.stattools import adfuller

# Define the function
def dickey_fuller(timeseries):

# Perform the Dickey-Fuller test:
    print ('Dickey-Fuller Stationarity test:')
    test = adfuller(timeseries, autolag = 'AIC')
    result = pd.Series(test[0:4], index = ['Test Statistic','p-value','Number of Lags Used','Number of Obse
    for key,value in test[4].items():
        result['Critical Value (%s)'%key] = value
    print (result)

# Apply the test
dickey_fuller(df_production_7['Average Production'])
```

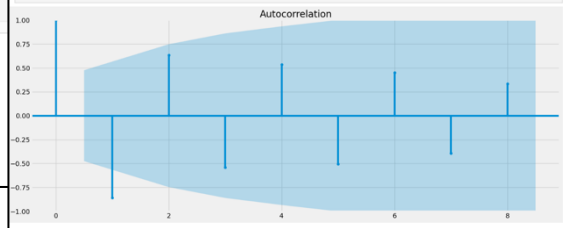
```
Dickey-Fuller Stationarity test:
Test Statistic                 -2.157692
p-value                         0.221958
Number of Lags Used             6.000000
Number of Observations Used    10.000000
Critical Value (1%)            -4.331573
Critical Value (5%)            -3.232950
Critical Value (10%)           -2.748700
dtype: float64
```

Comments

- At a significance level of 5%, the test statistic is smaller than the critical value and the null hypothesis can be rejected.
- That there is not a unit root in the data, and the data is stationary.

Autocorrelation - Round 2

```
# Plot of autocorrelations
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

plot_acf(df_production_7)
plt.show()
```

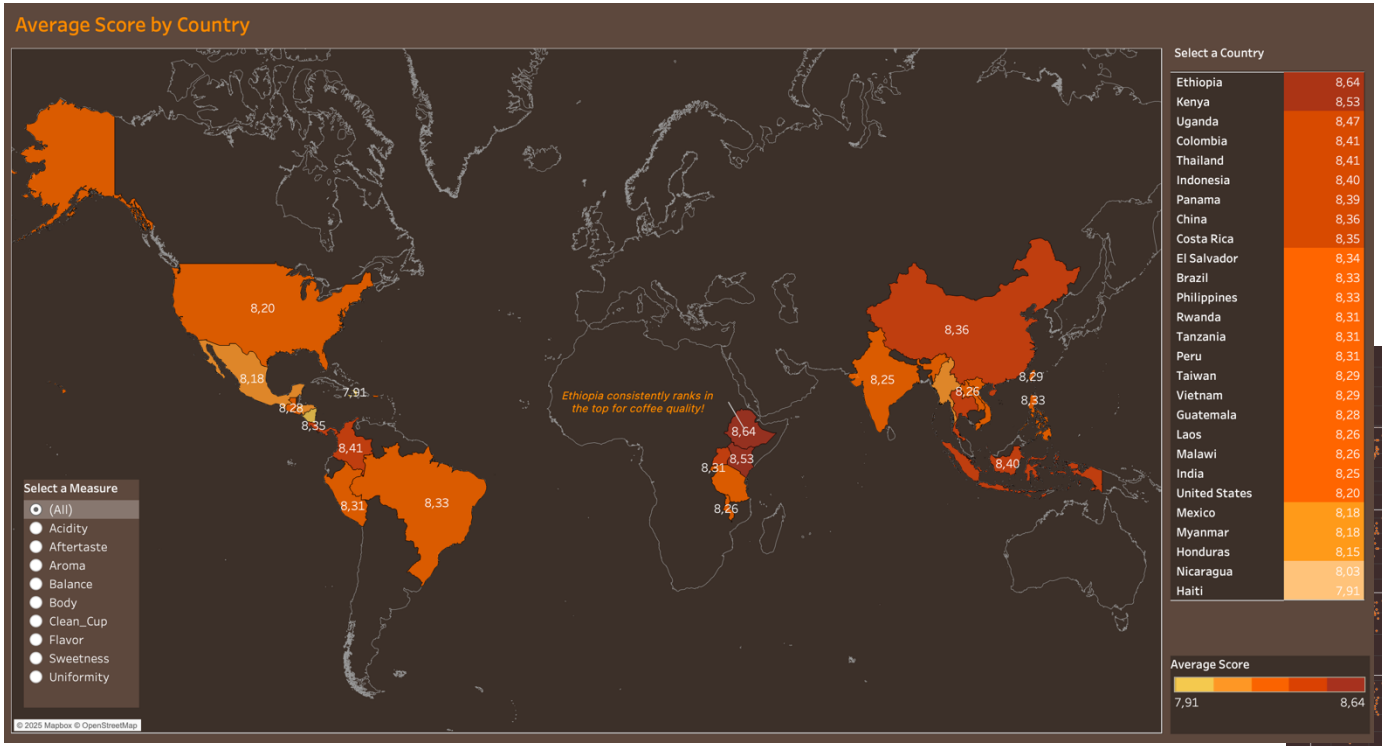


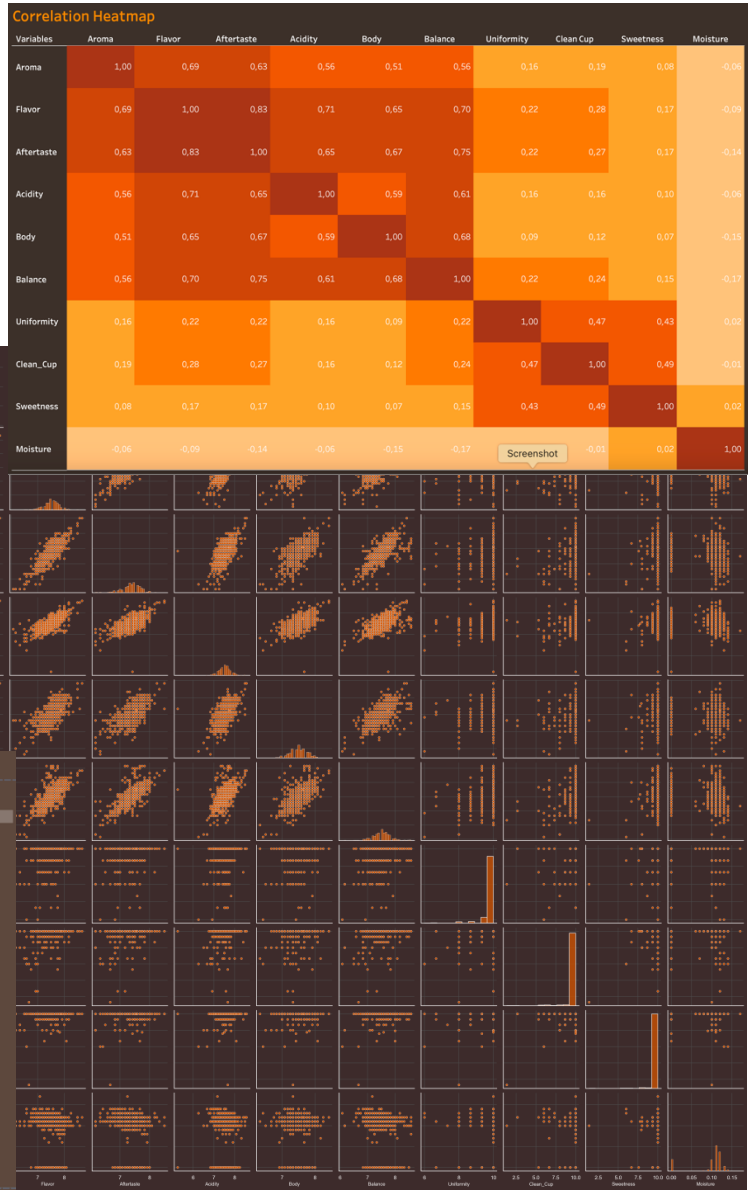
Comments

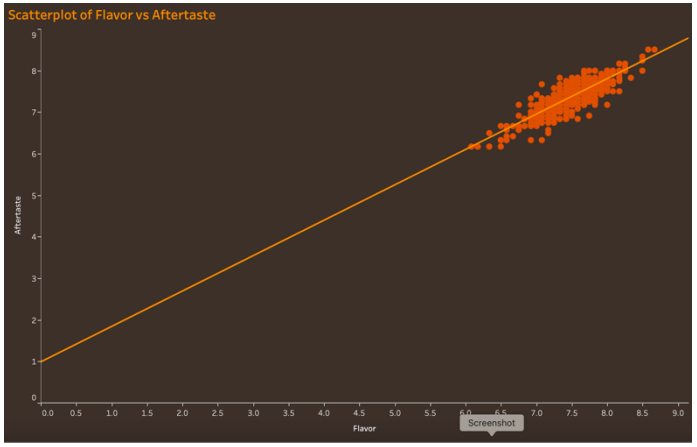
- The plot suggests reduced autocorrelation.

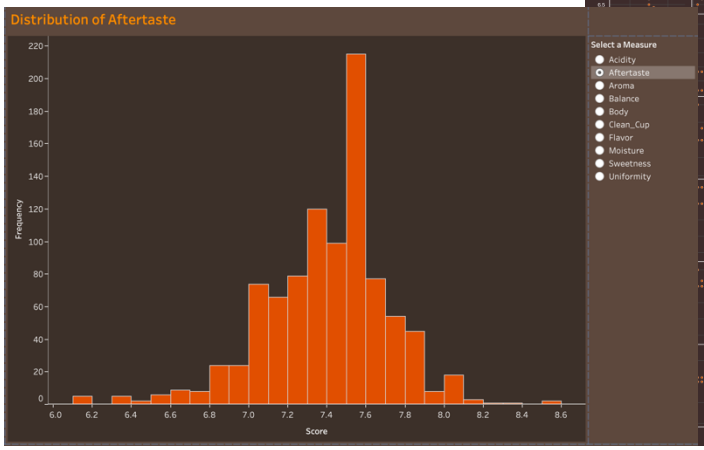
Analysis & Insights

Ordering Trends









Recommendations

The analysis achieved its goals of identifying top coffee-producing countries and exploring score predictability across quality measures. Ethiopia emerged as a standout, with consistently high coffee ratings, while Central American and Southeast Asian countries scored lower overall.

Strong correlations were found among Flavour, Aftertaste, and Balance, with Aftertaste being a reliable predictor of Flavour, though less so for Balance.

Clustering the data offered no added insights beyond the observed linear relationships.

Due to data limitations, the analysis could not determine whether higher-quality coffee is in greater demand.

Reflection

Successes

Overall, the project provided meaningful insights into coffee quality metrics and their interactions, highlighting Ethiopia's dominance in producing high-quality coffee and that it is possible to predict scores of certain measures using the scores of others.

Challenges

It was not possible to obtain a dataset with sufficient granularity to assess whether there is a higher demand for better quality coffee, underscoring the importance of clearly defined data requirements at the outset of a project.

Moving Forward

Reach out to the Coffee Quality Institute to inquire about access to a more suitable dataset.

Investigate alternative modelling approaches for predicting the less correlated measures.