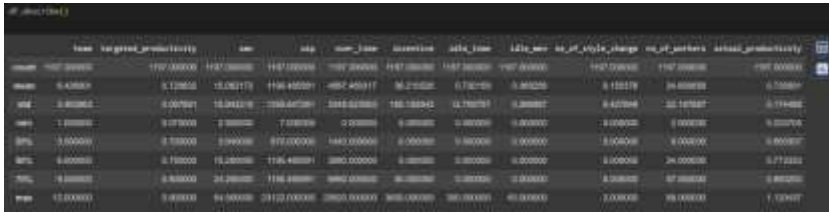


Data Collection and Preprocessing Phase

Date	15 July 2024
Team ID	SWTID1720108643
Project Title	Garment Worker Productivity Prediction
Maximum Marks	6 Marks

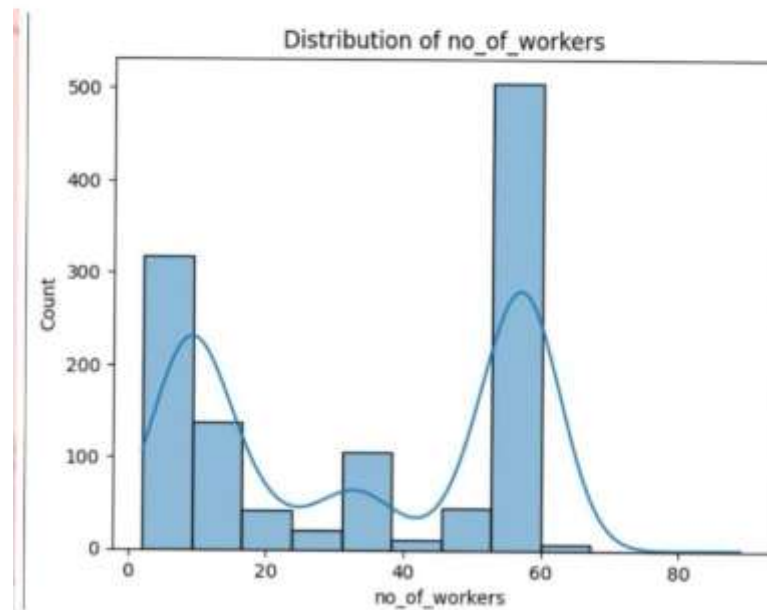
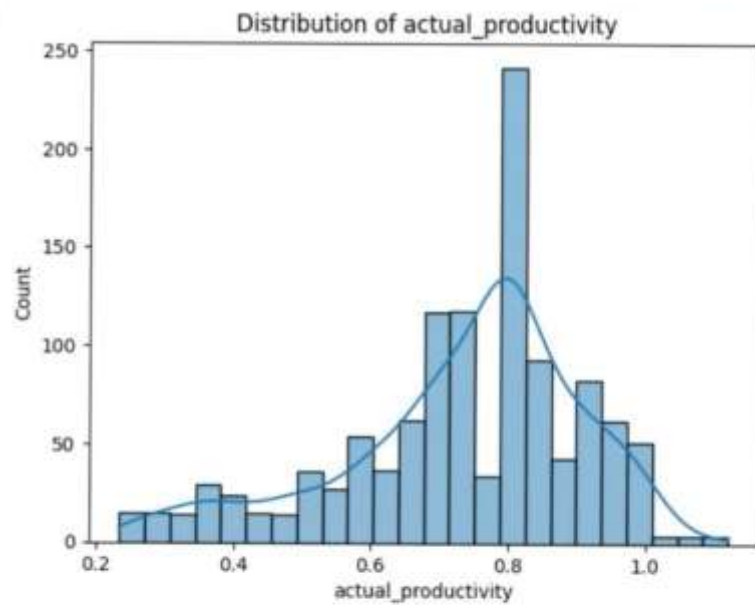
Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

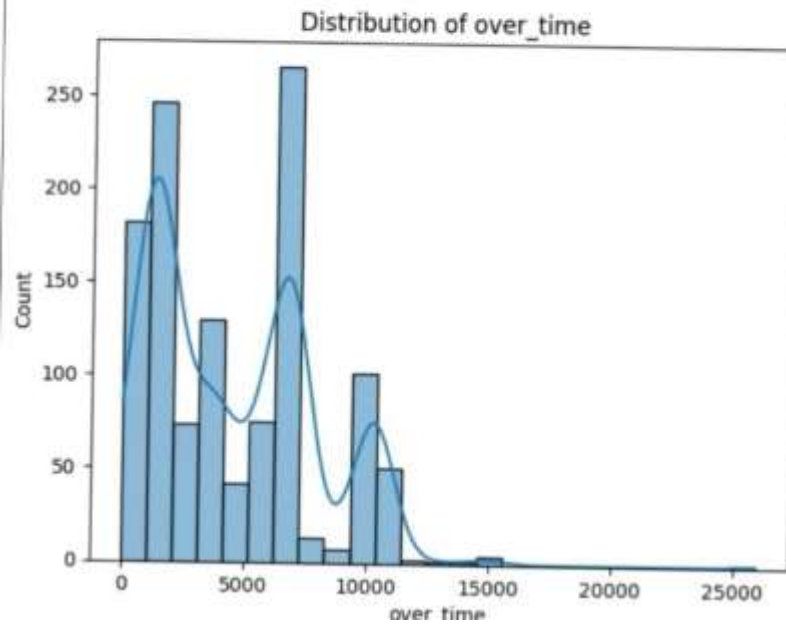
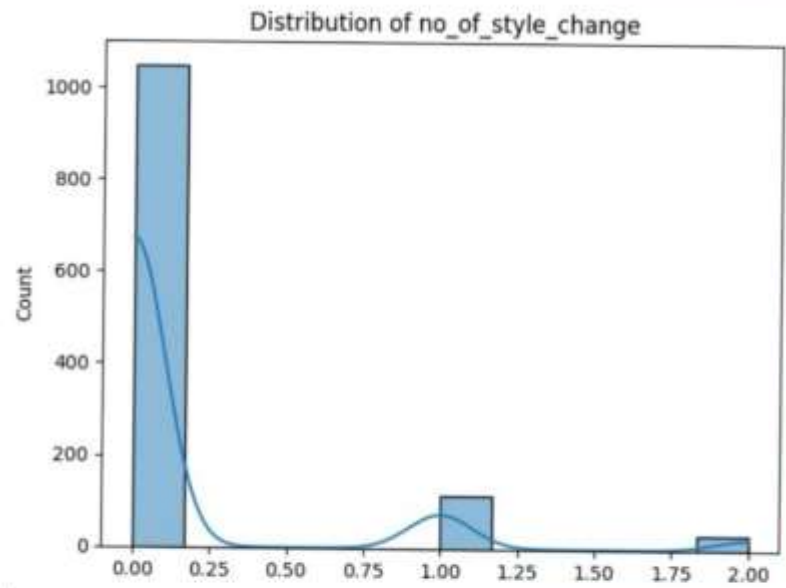
Section	Description
Data Overview	<p><u>The Shape of the Data Frame: (1197, 13)</u></p> <p>It consists of 1197 Rows and 13 Columns.</p> 

Exploration of individual variables (mean, median, mode, etc.).

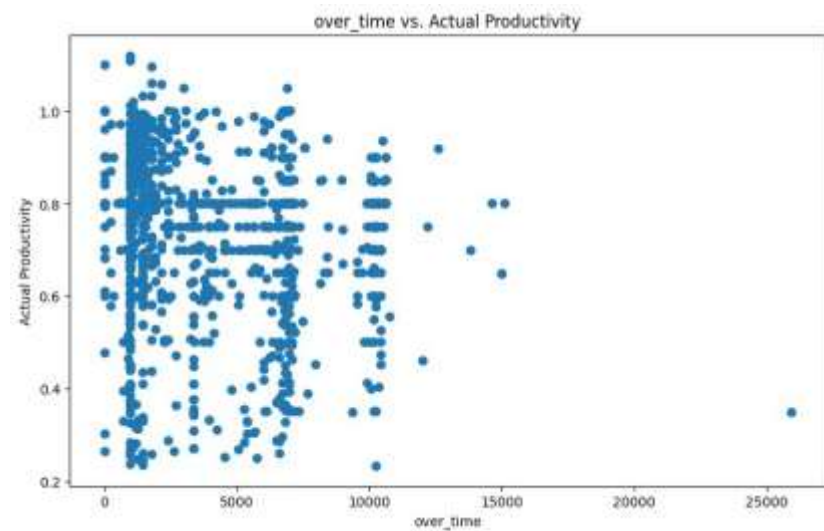
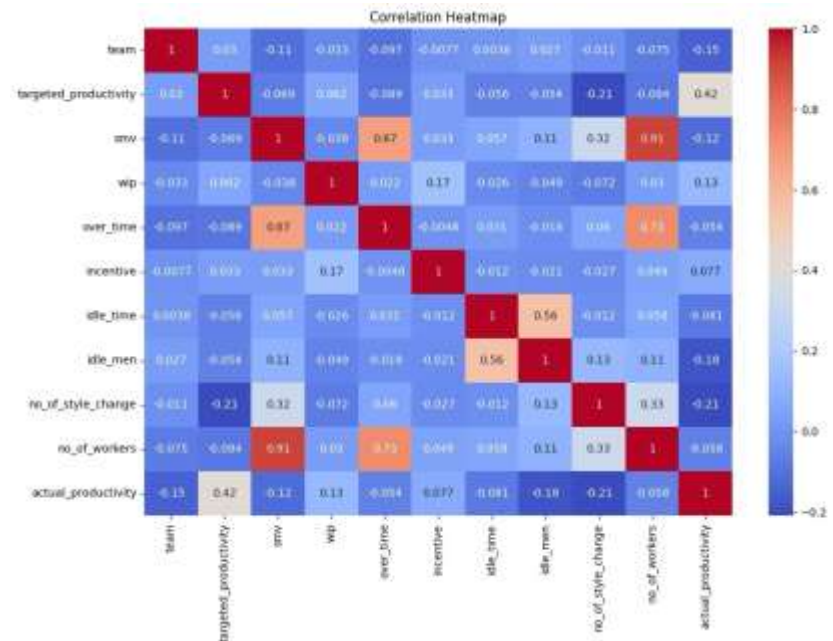
Univariate Analysis



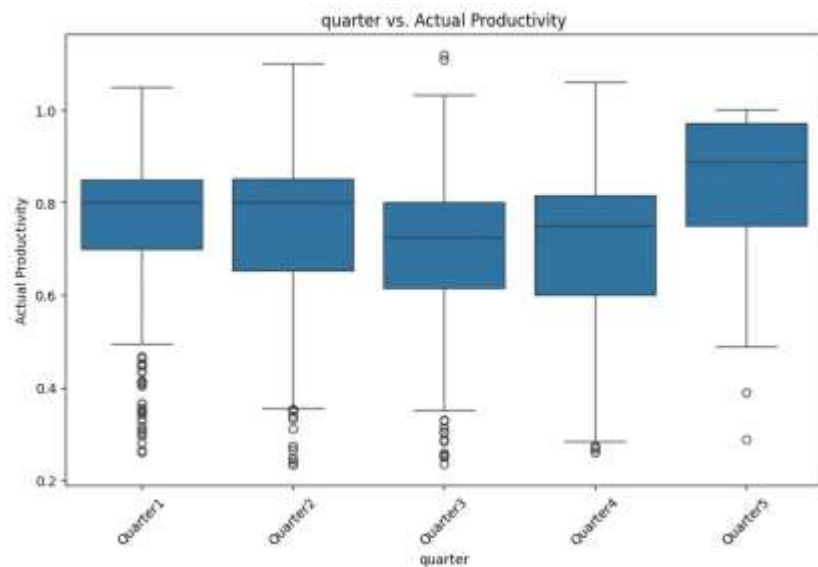
Univariate Analysis



Bivariate Analysis

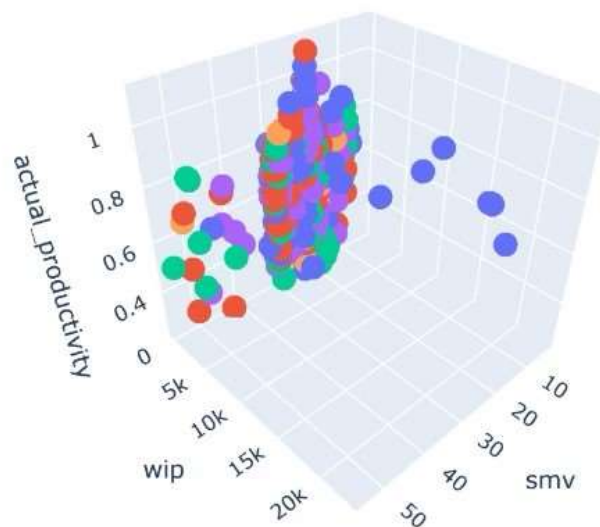


Bivariate Analysis



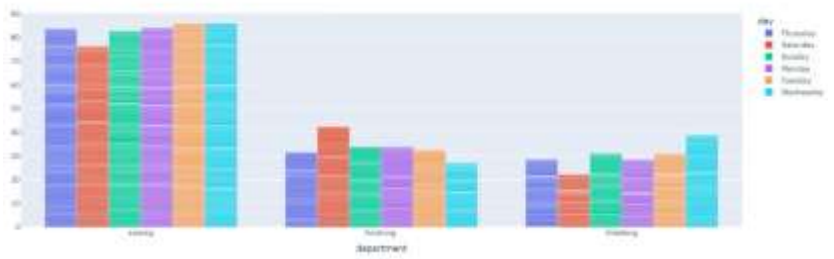
Multivariate Analysis

Patterns and relationships involving multiple variables.



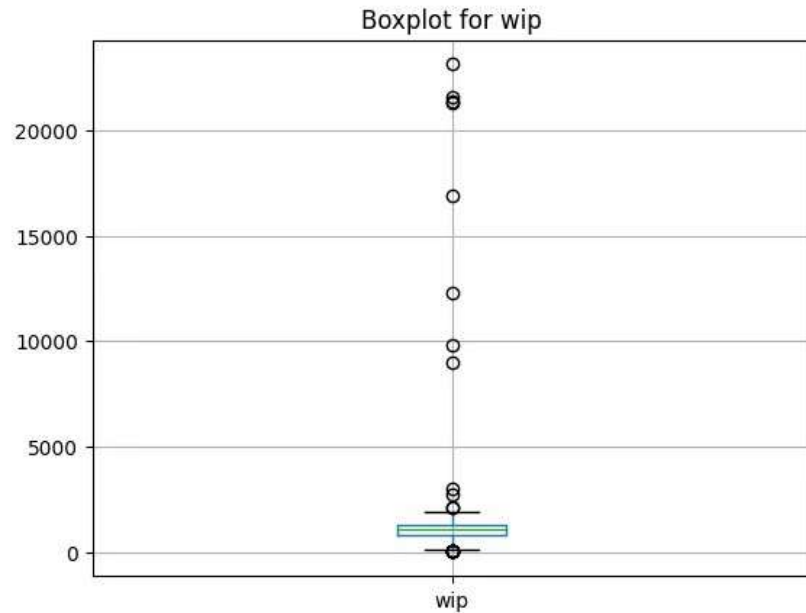
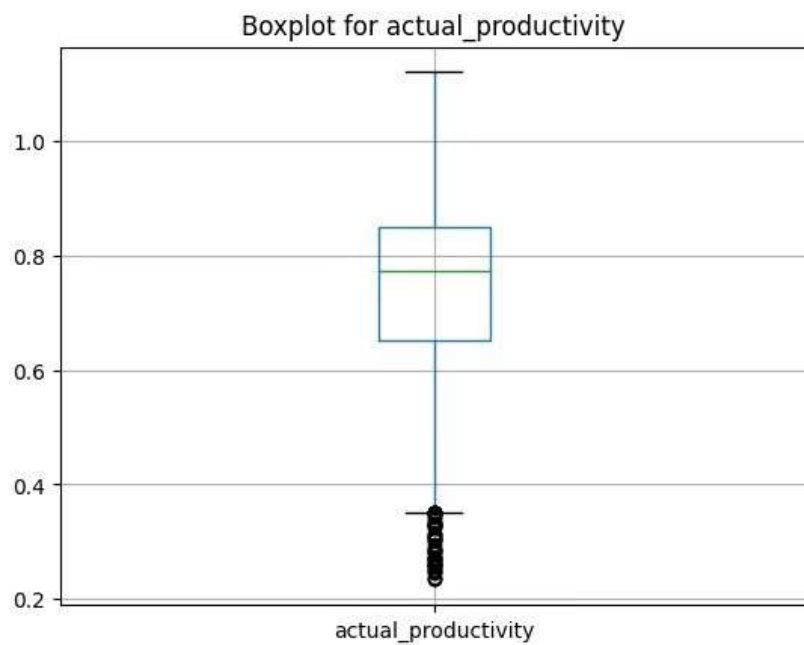
Multivariate Analysis

Actual Productivity by Department and Day



Outliers and Anomalies

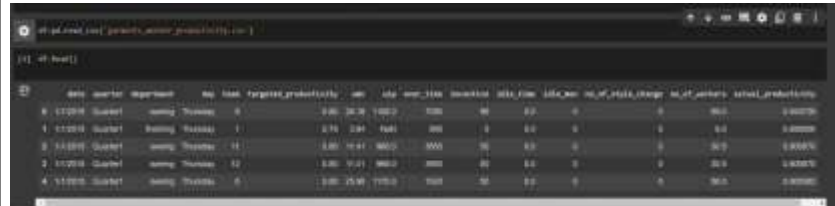
Identification and treatment of outliers.



Data Preprocessing Code Screenshots

Loading Data

Code to load the dataset into the preferred environment (e.g., Python, R).

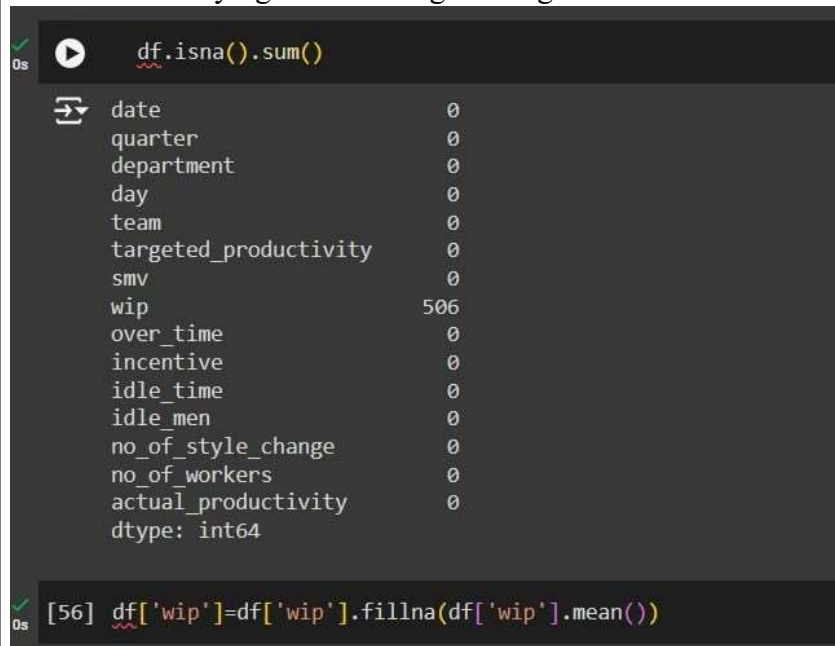


```
[1]: df.head()
```

	date	quarter	department	emp_hires	targeted_productivity	smv	idp	emp_size	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_productivity
0	11/2014	Quarter	Marketing	0	0.00	20.38	100.0	1000	0	0.0	0	0	0.0	0.000000
1	11/2014	Quarter	Marketing	1	0.74	2.94	74.0	800	0	0.0	0	0	0.0	0.000000
2	11/2014	Quarter	Marketing	11	0.00	10.41	980.0	3500	0	0.0	0	0	0.0	0.000000
3	11/2014	Quarter	Marketing	12	0.00	10.41	980.0	3500	0	0.0	0	0	0.0	0.000000

Handling Missing Data

Code for identifying and handling missing values.



```
df.isna().sum()
```

```

date          0
quarter       0
department    0
day           0
team          0
targeted_productivity  0
smv           0
wip          506
over_time     0
incentive     0
idle_time     0
idle_men      0
no_of_style_change  0
no_of_workers  0
actual_productivity  0
dtype: int64

```

```
[56] df['wip']=df['wip'].fillna(df['wip'].mean())
```


Data Transformation

Code for transforming variables (scaling, normalization).

“from sklearn.preprocessing import StandardScaler,
MinMaxScaler

```
scaler = StandardScaler()
```

```
numerical_features = ['smv', 'wip', 'over_time', 'incentive',  
'idle_time', 'idle_men', 'no_of_style_change', 'no_of_workers']
```

```
df[numerical_features] =
```

```
scaler.fit_transform(df[numerical_features])
```

df.head()”

center	department	dep_head	days	targeted_productivity	smv	smv_over_time	incentive	idle_time	idle_men	no_of_style_change	no_of_workers	actual_productivity
Center1	sewing	Production	0	0.96	1.078502	-0.077086	22	0.170202	0.074054	-0.021473	0.110000	0.661128
Center1	sewing	Production	1	0.74	1.000198	-0.098161	11	1.071000	0.200000	-0.007470	0.110000	0.680000
Center1	sewing	Production	11	0.96	-0.030079	-0.000000	22	0.271300	0.073001	-0.021473	0.110000	0.690070
Center1	sewing	Production	12	0.96	-0.030079	-0.000000	22	0.271300	0.073001	-0.021473	0.110000	0.690070
Center1	sewing	Production	0	0.96	0.990700	-0.077086	22	0.170202	0.074054	-0.021473	0.110000	0.661128

Feature Engineering

Code for creating new features or modifying existing ones.

```
df['efficiency'] = df['actual_productivity'] / df['smv']

median_overtime = df['over_time'].median()
df['over_time_category'] = df['over_time'].apply(lambda x: 'high' if x > median_overtime else 'low')
print(df[['efficiency', 'over_time_category']].head())
```

```
efficiency over_time_category
0    0.927232             high
1    -0.071872             low
2    -2.397794             low
3    -2.397794             low
4    -0.097828             low
```

Save Processed Data	<p>Code to save the cleaned and processed data for future use.</p> <pre data-bbox="600 331 1421 495">df.to_csv('preprocessed_data.csv', index=False)</pre>
---------------------	--