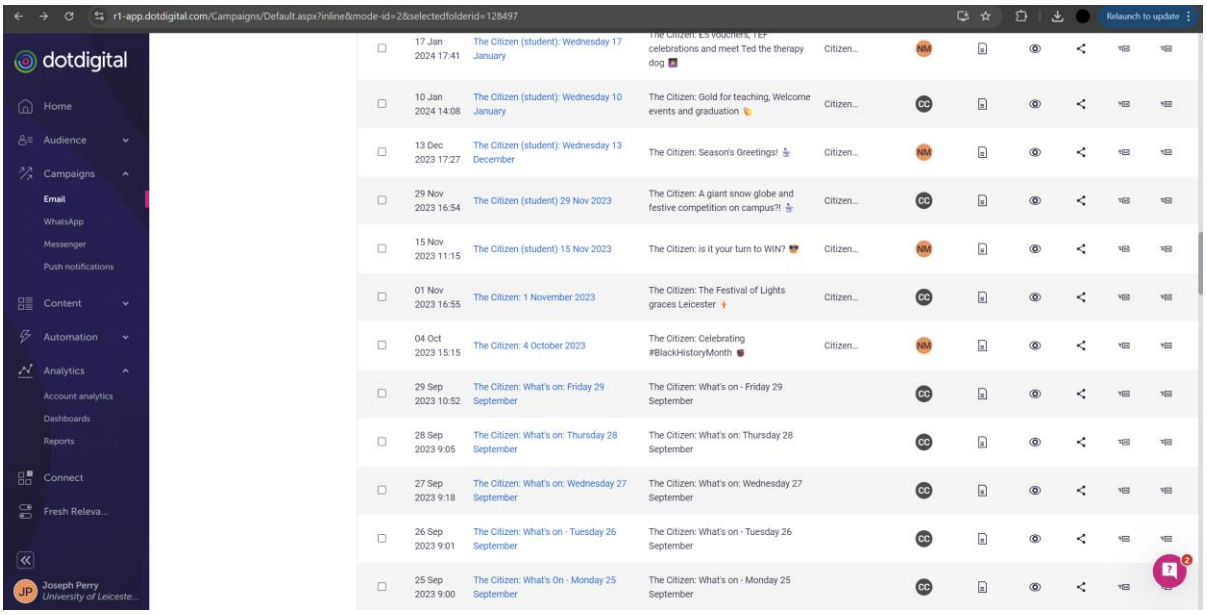
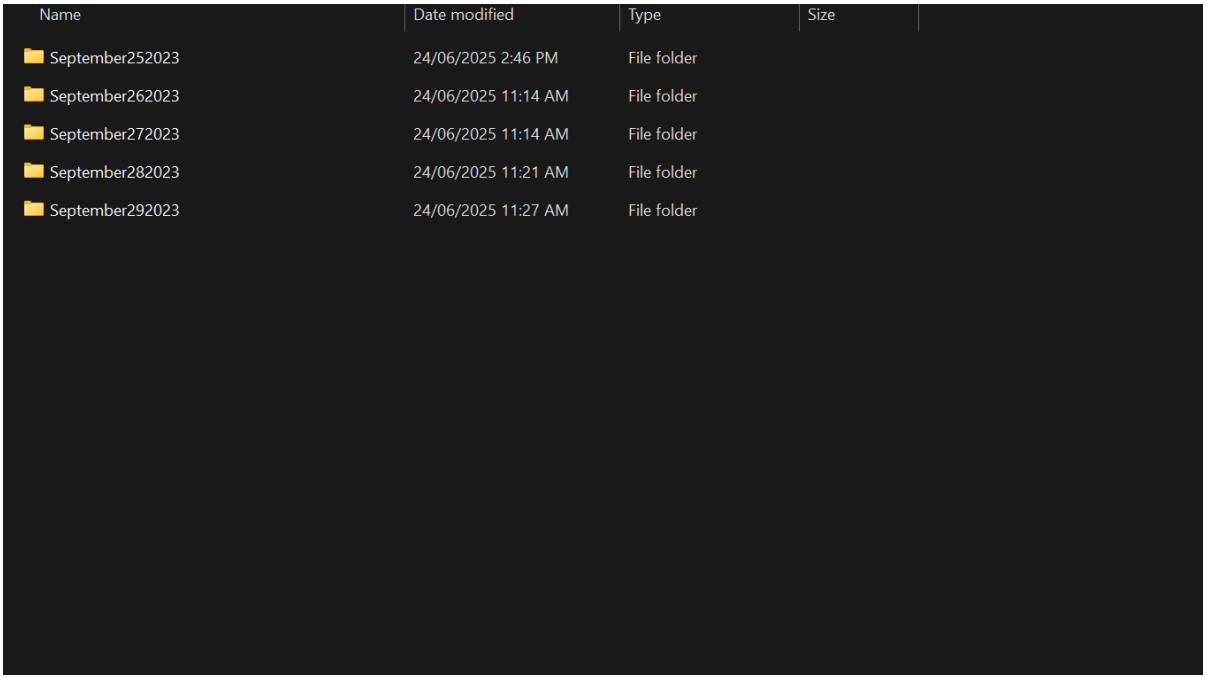
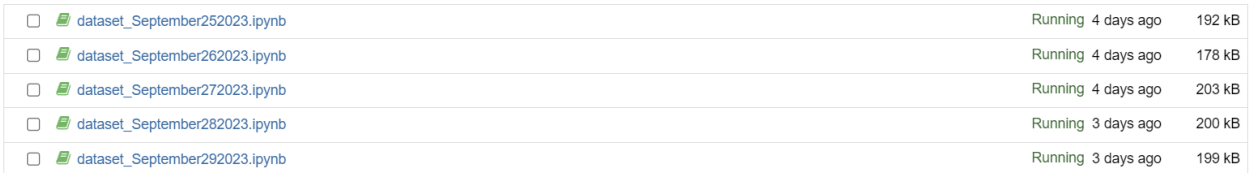
1. Accessed DotDigital portal where statistical data on university records is found.



2. Collected datasets from freshers week within the past two years.



3. Pre-processed data, using best approaches for the given data.



4. Defined functions and new columns for additional calculations.

In [44]:
```
def total_opens():
    total_opens = df_emailopened["NumOpens"].sum()
    print("Total number of NumOpens: ", total_opens)



total_opens()
```

```
Total number of NumOpens:  28672
```

In [47]:
```
average_opens = df_emailopened["NumOpens"].mean()
df_emailopened["AverageNumOpens"] = average_opens

print("Average number of NumOpens:", average_opens)
```

```
Average number of NumOpens: 2.1965831609591664
```

In [48]:
```
minimum_opens = df_emailopened["NumOpens"].min()
df_emailopened["MinimumOpens"] = minimum_opens
print("Minimum number of NumOpens: ", minimum_opens)
```

```
Minimum number of NumOpens:  1
```

In [32]:
```
maximum_opens = df_emailopened["NumOpens"].max()
print("Maximum number of NumOpens: ", maximum_opens)
```

```
Maximum number of NumOpens:  30
```

In [49]:
```
df_emailopened["DateOpened"] = pd.to_datetime(df_emailopened["DateOpened"], format="%d/%m/%Y %H:%M") #convert string to datetime
```

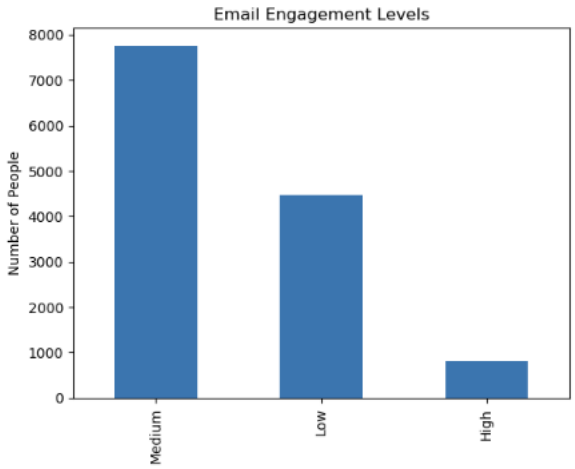
In [34]:
```
df_emailopened["DateOpenedLast"] = pd.to_datetime(df_emailopened["DateOpenedLast"], format="%d/%m/%Y %H:%M") #convert string to d
```

In [35]:
```
df_emailopened["EngagementSpan"] = df_emailopened["DateOpenedLast"] - df_emailopened["DateOpened"]
df_emailopened.head(10)
```

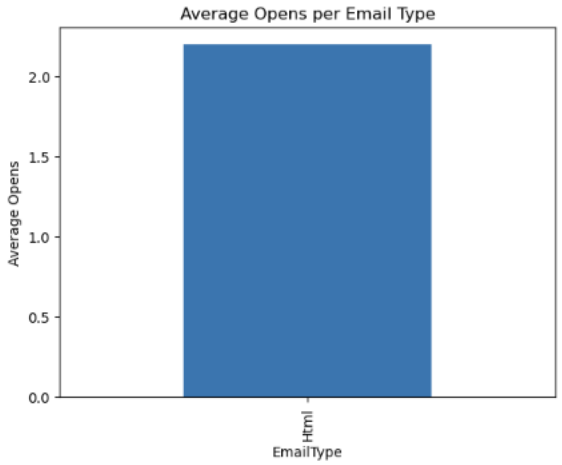
5. Analysed data between comparable variables and information.

6. Created visualisations and trends for the appropriate variables useful for addressing the business problem.

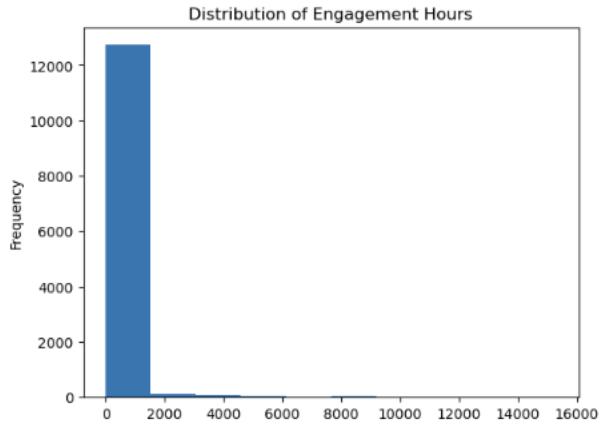
```
In [39]: df_emailopened["Openlevel"].value_counts().plot(kind='bar', title='Email Engagement Levels')
plt.ylabel("Number of People")
plt.show()
#high levels of medium open level
#Low levels of high open level
```



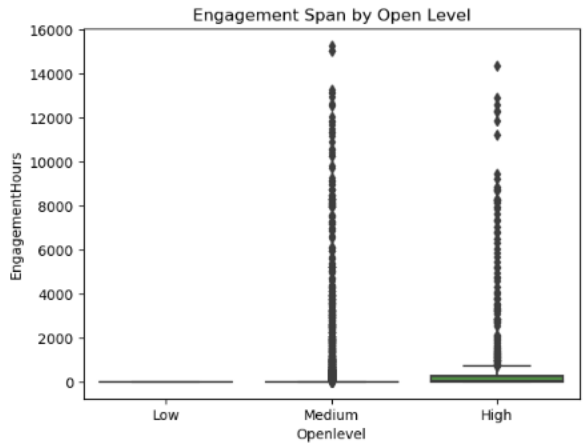

```
In [50]: df_emailopened.groupby("EmailType")["AverageNumOpens"].mean().plot(kind="bar", title="Average Opens per Email Type")
plt.ylabel("Average Opens")
plt.show()
```



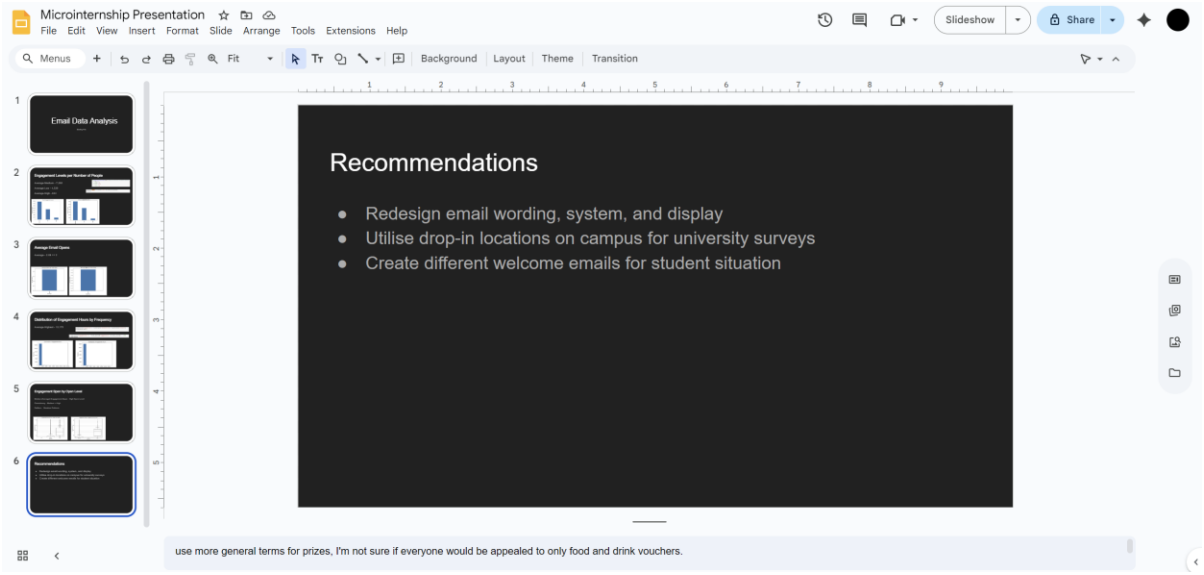
```
In [41]: df_emailopened['EngagementHours'].plot(kind='hist', bins=10, title='Distribution of Engagement Hours')
         plt.show()
```




```
In [42]: sns.boxplot(x='Openlevel', y='EngagementHours', data=df_emailopened).set_title('Engagement Span by Open Level')
Out[42]: Text(0.5, 1.0, 'Engagement Span by Open Level')
```



7. Created a presentation to present findings and deliver recommendations.


Microinternship Presentation
File Edit View Insert Format Slide Arrange Tools Extensions Help
Slideshow
Share
Menus
Fit
Background
Layout
Theme
Transition
Email Data Analysis
Recommendations
Redesign email wording, system, and display
Utilise drop-in locations on campus for university surveys
Create different welcome emails for student situation
use more general terms for prizes, I'm not sure if everyone would be appealed to only food and drink vouchers.