Importing necessary libraries

In [ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the Dataset

In [ ]:
```
df = pd.read_csv('bank-full.csv', delimiter=';')

print(df.head())
```

```
   age           job  marital  education default  balance housing loan  \
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married    unknown      no     1506     yes   no
4   33       unknown   single    unknown      no        1      no   no

   contact  day month  duration  campaign  pdays  previous poutcome   y
0  unknown    5   may       261         1     -1         0  unknown  no
1  unknown    5   may       151         1     -1         0  unknown  no
2  unknown    5   may        76         1     -1         0  unknown  no
3  unknown    5   may        92         1     -1         0  unknown  no
4  unknown    5   may       198         1     -1         0  unknown  no
```

Basic information about the dataset

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        45211 non-null  int64 
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64 
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64 
 10  month      45211 non-null  object
 11  duration   45211 non-null  int64 
 12  campaign   45211 non-null  int64 
 13  pdays      45211 non-null  int64 
 14  previous   45211 non-null  int64 
 15  poutcome   45211 non-null  object
 16  y          45211 non-null  object
dtypes: int64(7), object(10)
memory usage: 5.9+ MB
```

Checking for Null Values

In [ ]:
```
print(df.isnull().sum())
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

Summary Statistics

In [ ]:
```
print(df.describe())
```

```
                age        balance           day      duration      campaign
\
count  45211.000000   45211.000000  45211.000000  45211.000000  45211.000000
mean      40.936210    1362.272058     15.806419    258.163080      2.763841
std       10.618762    3044.765829      8.322476    257.527812      3.098021
min       18.000000   -8019.000000      1.000000      0.000000      1.000000
25%       33.000000      72.000000      8.000000    103.000000      1.000000
50%       39.000000     448.000000     16.000000    180.000000      2.000000
75%       48.000000    1428.000000     21.000000    319.000000      3.000000
max       95.000000  102127.000000     31.000000   4918.000000     63.000000

              pdays      previous
count  45211.000000  45211.000000
mean      40.197828      0.580323
std      100.128746      2.303441
min       -1.000000      0.000000
25%       -1.000000      0.000000
50%       -1.000000      0.000000
75%       -1.000000      0.000000
max      871.000000    275.000000
```

Target Value Percentage Distribution

In [ ]:
```
# Count the number of subscriptions to term deposits (target variable 'y')
print(df['y'].value_counts(normalize=True) * 100)  # Percentage of yes/no
```

```
y
no     88.30152
yes    11.69848
Name: proportion, dtype: float64
```

Separate numerical and categorical features

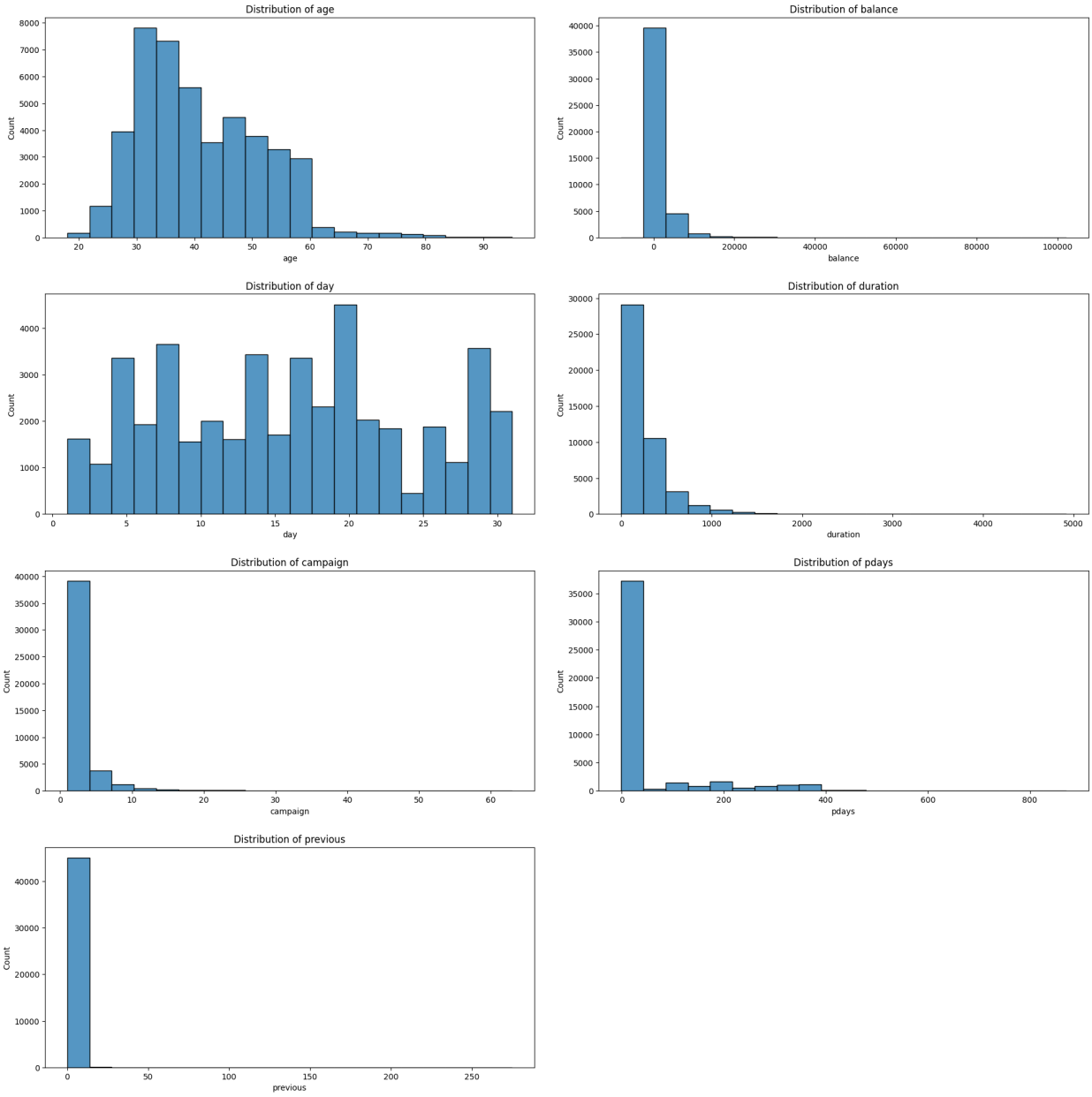
In [ ]:
```
numerical_features = df.select_dtypes(include=['int64', 'float64']).columns
categorical_features = df.select_dtypes(include=['object']).columns
```

EDA on Numerical Features

Histograms

In [ ]:
```
numerical_features = df.select_dtypes(include=['int64', 'float64']).columns.to

# Grid for subplots
n_cols = 2  # Number of columns in the grid
n_rows = (len(numerical_features) + n_cols - 1) // n_cols  # Calculate the requ

# Subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(20, 5 * n_rows))
fig.tight_layout(pad=5.0)  # Add space between plots
```

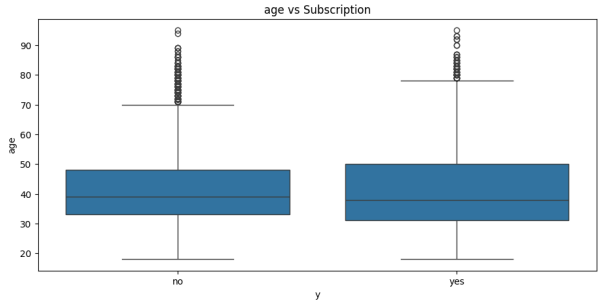
```
# Loop through the numerical features and create a histogram for each
for i, feature in enumerate(numerical_features):
    row = i // n_cols
    col = i % n_cols
    ax = axes[row, col]
    sns.histplot(df[feature], bins=20, ax=ax)
    ax.set_title(f'Distribution of {feature}')

# If the number of features is odd, remove the last ax
if len(numerical_features) % n_cols != 0:
    fig.delaxes(axes[-1, -1])  # Remove the empty subplot if necessary

plt.show()
```

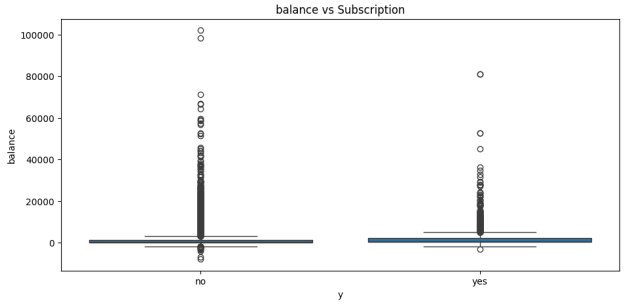


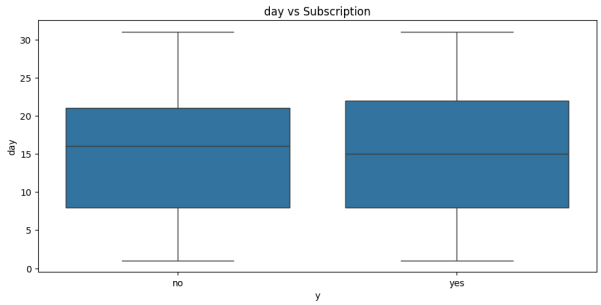
Box Plots

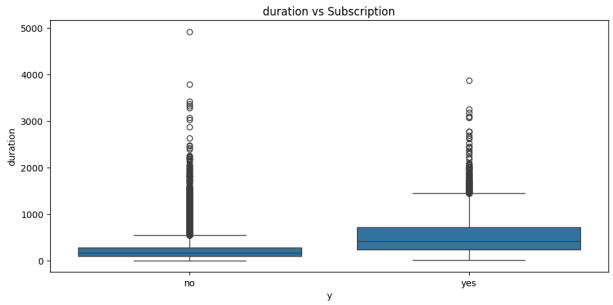
In [ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

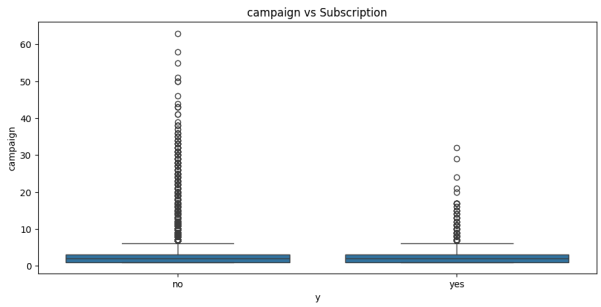
```
numerical_features = df.select_dtypes(include=['int64', 'float64']).columns.to

# Define the size of the grid
n_cols = 2  # Number of columns in the grid
n_rows = (len(numerical_features) + n_cols - 1) // n_cols  # Calculate the requ

# Subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(20, 5 * n_rows))
fig.tight_layout(pad=5.0)  # Add space between plots

# Loop through the numerical features and create a boxplot for each
for i, feature in enumerate(numerical_features):
    row = i // n_cols
    col = i % n_cols
    ax = axes[row, col]
    sns.boxplot(x='y', y=feature, data=df, ax=ax)
    ax.set_title(f'{feature} vs Subscription')

# If the number of features is odd, remove the last ax
if len(numerical_features) % n_cols != 0:
    fig.delaxes(axes[-1, -1])  # Remove the empty subplot if necessary

plt.show()
```

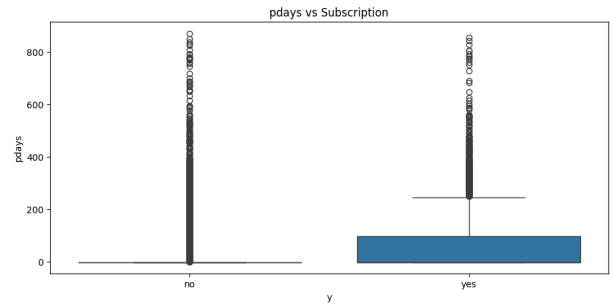


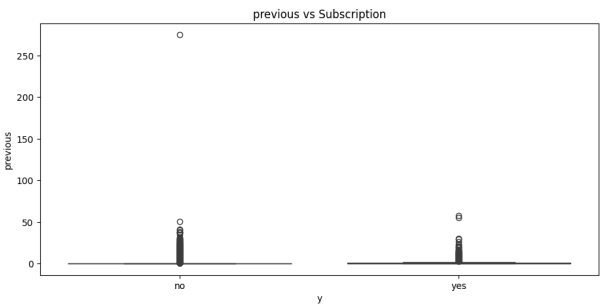








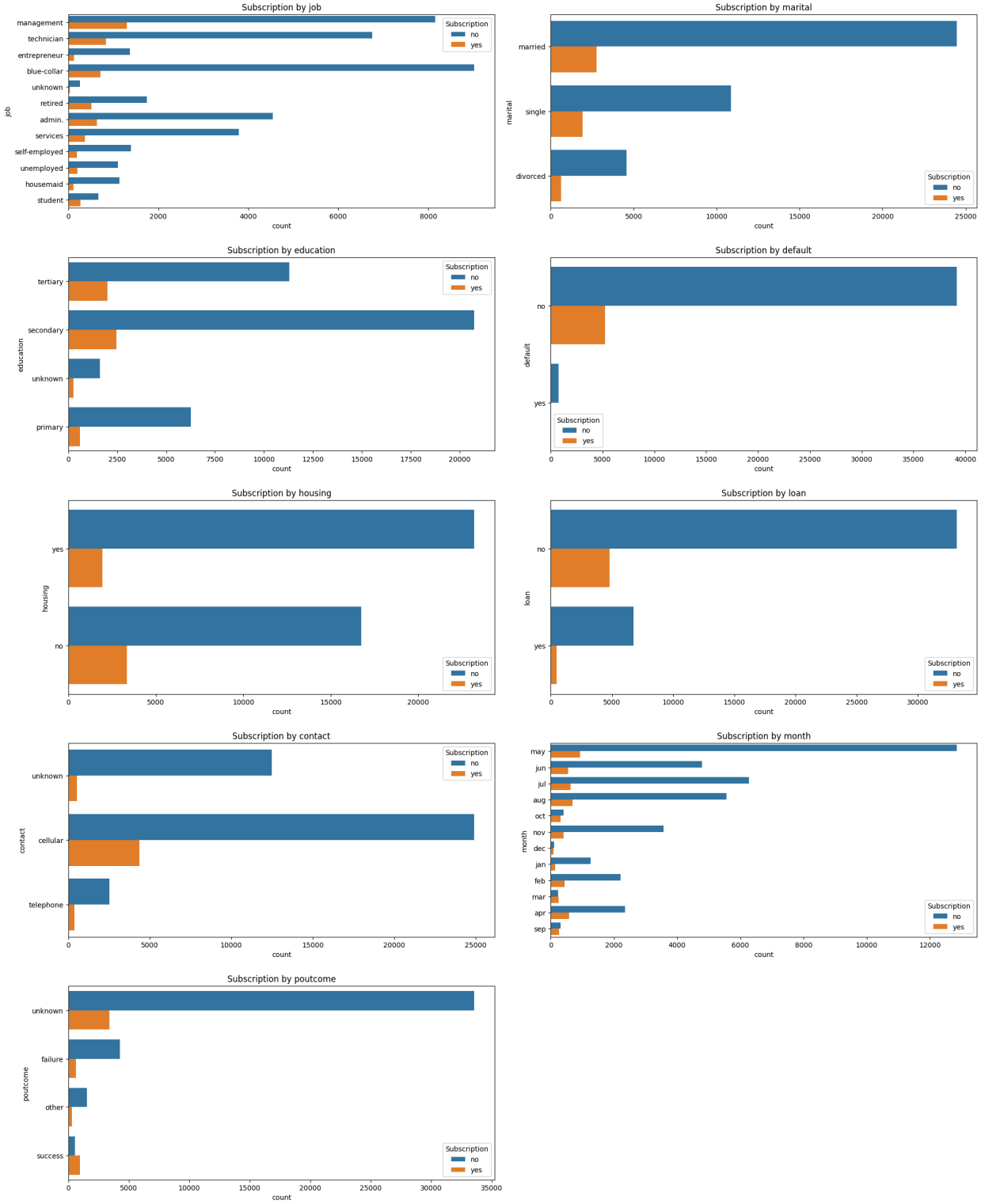




EDA on Categorical Features

In [ ]:
```
categorical_features = categorical_features.tolist()
categorical_features.remove('y')  # Remove the target variable if it's in the

# Grid for subplots
n_cols = 2
n_rows = (len(categorical_features) + n_cols - 1) // n_cols  # Calculate the r

# Grid for subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(20, 5 * n_rows))
fig.tight_layout(pad=5.0)  # Add space between plots

# Loop through the categorical features and create a countplot for each
for i, feature in enumerate(categorical_features):
    row = i // n_cols
    col = i % n_cols
    ax = axes[row, col]
```

```
    sns.countplot(y=feature, hue='y', data=df, ax=ax)
    ax.set_title(f'Subscription by {feature}')
    ax.legend(title='Subscription')

# If the number of features is odd, remove the last ax
if len(categorical_features) % n_cols != 0:
    fig.delaxes(axes[-1, -1])  # Remove the empty subplot if necessary

plt.show()
```



Converting ipynb to PDF and HTML

In [ ]:
```
bank_new = df.to_csv('bank_new.csv', index = True)
```

In [ ]:
```
!jupyter nbconvert --to pdf /content/Cruisebound.ipynb
!jupyter nbconvert --to html /content/Cruisebound.ipynb
```

```
[NbConvertApp] Converting notebook /content/Cruisebound.ipynb to pdf
[NbConvertApp] ERROR | Error while converting '/content/Cruisebound.ipynb'
Traceback (most recent call last):
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/nbconvertapp.py", li
ne 488, in export_single_notebook
    output, resources = self.exporter.from_filename(
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/exporter.p
y", line 189, in from_filename
    return self.from_file(f, resources=resources, **kw)
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/exporter.p
y", line 206, in from_file
    return self.from_notebook_node(
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/pdf.py", l
ine 181, in from_notebook_node
    latex, resources = super().from_notebook_node(nb, resources=resources, **k
w)
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/latex.py",
line 74, in from_notebook_node
    return super().from_notebook_node(nb, resources, **kw)
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/exporters/templateex
porter.py", line 413, in from_notebook_node
    output = self.template.render(nb=nb_copy, resources=resources)
  File "/usr/local/lib/python3.10/dist-packages/jinja2/environment.py", line 1
304, in render
    self.environment.handle_exception()
  File "/usr/local/lib/python3.10/dist-packages/jinja2/environment.py", line 9
39, in handle_exception
    raise rewrite_traceback_stack(source=source)
  File "/usr/local/share/jupyter/nbconvert/templates/latex/index.tex.j2", line
8, in top-level template code
    ((* extends cell_style *))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/style_jupyter.tex.j
2", line 176, in top-level template code
    \prompt{(((prompt)))}{(((prompt_color)))}{(((execution_count)))}{(((extra_
space)))}
  File "/usr/local/share/jupyter/nbconvert/templates/latex/base.tex.j2", line
7, in top-level template code
    ((*- extends 'document_contents.tex.j2' -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/document_contents.t
ex.j2", line 51, in top-level template code
    ((*- block figure scoped -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/display_priority.j
2", line 5, in top-level template code
    ((*- extends 'null.j2' -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 30,
in top-level template code
    ((*- block body -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/base.tex.j2", line
215, in block 'body'
    ((( super() )))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 32,
in block 'body'
    ((*- block any_cell scoped -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/null.j2", line 85,
in block 'any_cell'
    ((*- block markdowncell scoped-*)) ((*- endblock markdowncell -*))
  File "/usr/local/share/jupyter/nbconvert/templates/latex/document_contents.t
ex.j2", line 68, in block 'markdowncell'
    ((( cell.source | citation2latex | strip_files_prefix | convert_pandoc('ma
rkdown+tex_math_double_backslash', 'json',extra_args=[]) | resolve_references
```

```
| convert_pandoc('json','latex'))))
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/filters/pandoc.py",
line 24, in convert_pandoc
    return pandoc(source, from_format, to_format, extra_args=extra_args)
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", li
ne 51, in pandoc
    check_pandoc_version()
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", li
ne 99, in check_pandoc_version
    v = get_pandoc_version()
  File "/usr/local/lib/python3.10/dist-packages/nbconvert/utils/pandoc.py", li
ne 76, in get_pandoc_version
    raise PandocMissing()
nbconvert.utils.pandoc.PandocMissing: Pandoc wasn't found.
Please check that pandoc is installed:
https://pandoc.org/installing.html
[NbConvertApp] Converting notebook /content/Cruisebound.ipynb to html
[NbConvertApp] Writing 1149000 bytes to /content/Cruisebound.html
```