Importing Pacakages and Library

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
```

Data Wrangling

Gathering Data

```
df = pd.read_csv('../data/M5_World_Championship.csv')
df.head()
```

```
   Index        Hero  T_Picked  T_Wins  T_Loses T_WinRate
T_PickPercentage  \
0      1   Valentina       727     334      393    45.94%
52.00%
1      2    Fredrinn       726     383      343    52.75%
51.93%
2      3     Terizla       620     348      272    56.13%
44.35%
3      4       Edith       527     265      262    50.28%
37.70%
4      5     Irithel       502     255      247    50.80%
35.91%

   BS_Picked  BS_Wins  BS_Loses  ... RS_Loses  RS_WinRate  T_Banned  \
0        348      157       191  ...      202      46.70%       484
1        361      184       177  ...      166      54.52%       475
2        291      168       123  ...      149      54.71%       435
3        258      125       133  ...      129      52.04%       477
4        245      129       116  ...      131      49.03%       208

   T_BansPercentage T_PicksBans  T_PicksBansPercentage  \
0            34.62%        1211                 86.62%
1            33.98%        1201                 85.91%
2            31.12%        1055                 75.46%
3            34.12%        1004                 71.82%
4            14.88%         710                 50.79%

                                        Played By  \
0  19 (11-8)19 (10-9)19 (10-9)19 (7-12)17 (8-9)
1  21 (15-6)18 (14-4)18 (12-6)18 (9-9)17 (10-7)
2  22 (12-10)18 (10-8)15 (9-6)14 (10-4)14 (8-6)
3     14 (9-5)14 (9-5)14 (6-8)13 (10-3)13 (9-4)
4    16 (8-8)15 (10-5)15 (10-5)12 (7-5)12 (7-5)

                                          Played With  \
0   164 (83-81) 162 (79-83) 140 (71-69) 121 (58-6...
```

```
1   164 (83-81) 131 (81-50) 126 (69-57) 122 (69-5...
2   162 (79-83) 114 (76-38) 108 (56-52) 105 (49-5...
3   122 (69-53) 112 (48-64) 109 (55-54) 100 (42-5...
4   140 (71-69) 126 (69-57) 108 (56-52) 100 (42-5...

                                          Played Vs.     Roles
0   218 (85-133) 178 (68-110) 174 (77-97) 173 (82...      Mage
1   218 (133-85) 190 (106-84) 185 (79-106) 173 (8...   Support
2   185 (106-79) 178 (110-68) 132 (70-62) 119 (65...   Fighter
3   174 (97-77) 147 (76-71) 132 (62-70) 108 (51-5...      Tank
4   145 (73-72) 132 (60-72) 120 (68-52) 110 (47-6...  Marksman

[5 rows x 23 columns]
```

Assessing Data

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 115 entries, 0 to 114
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Index                  115 non-null    int64
 1   Hero                   115 non-null    object
 2   T_Picked               115 non-null    int64
 3   T_Wins                 115 non-null    int64
 4   T_Loses                115 non-null    int64
 5   T_WinRate              115 non-null    object
 6   T_PickPercentage       115 non-null    object
 7   BS_Picked              115 non-null    int64
 8   BS_Wins                115 non-null    int64
 9   BS_Loses               115 non-null    int64
 10  BS_WinRate             115 non-null    object
 11  RS_Picked              115 non-null    int64
 12  RS_Wins                115 non-null    int64
 13  RS_Loses               115 non-null    int64
 14  RS_WinRate             115 non-null    object
 15  T_Banned               115 non-null    int64
 16  T_BansPercentage       115 non-null    object
 17  T_PicksBans            115 non-null    int64
 18  T_PicksBansPercentage  115 non-null    object
 19  Played By              115 non-null    object
 20  Played With            115 non-null    object
 21  Played Vs.             115 non-null    object
 22  Roles                  115 non-null    object
dtypes: int64(12), object(11)
memory usage: 20.8+ KB

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

```
Index                     0
Hero                      0
T_Picked                  0
T_Wins                    0
T_Loses                   0
T_WinRate                 0
T_PickPercentage          0
BS_Picked                 0
BS_Wins                   0
BS_Loses                  0
BS_WinRate                0
RS_Picked                 0
RS_Wins                   0
RS_Loses                  0
RS_WinRate                0
T_Banned                  0
T_BansPercentage          0
T_PicksBans               0
T_PicksBansPercentage     0
Played By                 0
Played With               0
Played Vs.                0
Roles                     0
dtype: int64
```

```
print("Total Duplication: ", df.duplicated().sum())
```

```
Total Duplication:  0
```

Data Cleaning

```
df.drop(['Index', 'Played By', 'Played With','Played Vs.'],
axis=1,inplace=True)
df.head()
```

```
         Hero  T_Picked  T_Wins  T_Loses T_WinRate T_PickPercentage  \
0   Valentina       727     334      393    45.94%           52.00%
1    Fredrinn       726     383      343    52.75%           51.93%
2     Terizla       620     348      272    56.13%           44.35%
3       Edith       527     265      262    50.28%           37.70%
4     Irithel       502     255      247    50.80%           35.91%

   BS_Picked  BS_Wins  BS_Loses BS_WinRate  RS_Picked  RS_Wins
RS_Loses  \
0        348      157       191     45.11%        379      177
202
1        361      184       177     50.97%        365      199
166
2        291      168       123     57.73%        329      180
149
```

```
3        258      125       133     48.45%        269      140
129
4        245      129       116     52.65%        257      126
131

  RS_WinRate  T_Banned T_BansPercentage  T_PicksBans
T_PicksBansPercentage  \
0     46.70%       484           34.62%         1211
86.62%
1     54.52%       475           33.98%         1201
85.91%
2     54.71%       435           31.12%         1055
75.46%
3     52.04%       477           34.12%         1004
71.82%
4     49.03%       208           14.88%          710
50.79%

      Roles
0      Mage
1   Support
2   Fighter
3      Tank
4  Marksman
```

```
df['T_WinRate'] = pd.to_numeric(df['T_WinRate'].str.strip('%'),
errors='coerce')
df['T_PickPercentage'] =
pd.to_numeric(df['T_PickPercentage'].str.strip('%'), errors='coerce')
df['BS_WinRate'] = pd.to_numeric(df['BS_WinRate'].str.strip('%'),
errors='coerce')
df['RS_WinRate'] = pd.to_numeric(df['RS_WinRate'].str.strip('%'),
errors='coerce')
df['T_BansPercentage'] =
pd.to_numeric(df['T_BansPercentage'].str.strip('%'), errors='coerce')
df['T_PicksBansPercentage'] =
pd.to_numeric(df['T_PicksBansPercentage'].str.strip('%'),
errors='coerce')

df.head()
```

```
         Hero  T_Picked  T_Wins  T_Loses  T_WinRate  T_PickPercentage
\
0   Valentina       727     334      393      45.94             52.00

1    Fredrinn       726     383      343      52.75             51.93

2     Terizla       620     348      272      56.13             44.35

3       Edith       527     265      262      50.28             37.70
```

```
4     Irithel       502     255      247      50.80              35.91


   BS_Picked  BS_Wins  BS_Loses  BS_WinRate  RS_Picked  RS_Wins
RS_Loses  \
0        348      157       191       45.11        379      177
202
1        361      184       177       50.97        365      199
166
2        291      168       123       57.73        329      180
149
3        258      125       133       48.45        269      140
129
4        245      129       116       52.65        257      126
131

   RS_WinRate  T_Banned  T_BansPercentage  T_PicksBans
T_PicksBansPercentage  \
0       46.70       484             34.62         1211
86.62
1       54.52       475             33.98         1201
85.91
2       54.71       435             31.12         1055
75.46
3       52.04       477             34.12         1004
71.82
4       49.03       208             14.88          710
50.79

      Roles
0      Mage
1   Support
2   Fighter
3      Tank
4  Marksman
```

Exploratory Data Analysis (EDA)

Distribution of Win Rates

What is the distribution of overall win rates for all heroes?

```
import plotly.io as pio

# Atur renderer default
pio.renderers.default = "browser"  # Membuka grafik di browser

# DataFrame yang sudah disortir
df_wr = df.sort_values('T_WinRate')
```

```
# Membuat grafik bar
fig = px.bar(
    df_wr,
    x='Hero',
    y='T_WinRate',
    color='T_WinRate',
    labels={"T_WinRate": "Win Rate"},
    title='The Distribution of Overall Win Rates'
)

# Menampilkan grafik
fig.show()
```

```
fig = px.scatter(df, x='Hero', y='T_WinRate',
             title='Win Rate Distribution for Each Hero',
             labels={'Hero': 'Heroes', 'T_WinRate': 'Win Rate'},
             color='Hero',
             color_discrete_sequence=px.colors.qualitative.Set1
             )

fig.update_layout(xaxis_title='Heroes', yaxis_title='Win Rate')
fig.show()
```

Are there any outliers in terms of exceptionally high or low win rates?

```
fig = px.box(df, y='T_WinRate', title='Box Plot of Overall Win Rates')
fig.update_layout(xaxis_title='Win Rate')
fig.show()
```

Side-based Analysis

How do win rates vary between Blue Side and Red Side?

```
fig = px.bar(df.sort_values('T_WinRate'), x='Hero', y=['BS_WinRate',
'RS_WinRate'],
             title='Win Rates by Side',
             labels={'value': 'Win Rate', 'variable': 'Side'},
             color_discrete_map={'BS_WinRate': 'blue', 'RS_WinRate':
'red'},
             barmode='group')

fig.update_layout(xaxis_title='Hero', yaxis_title='Win Rate',
legend_title='Side')
fig.show()
```

What is the distribution of hero roles in the dataset?

```
role_counts = df['Roles'].value_counts().reset_index()
fig = px.bar(
    role_counts,
```

```
    x='count',
    y='Roles',
    labels={'count': 'Role', 'Role': 'Count'},
    title='Distribution of Hero Roles in the Dataset',
)
fig.update_layout(
    yaxis=dict(title='Count'),
)
fig.show()

fig = px.pie(
    role_counts,
    names='count',
    values='Roles',
    title='Distribution of Hero Roles in the Dataset',
)
fig.show()
```

Role-wise Performance

How are the win rates distributed among different hero roles?

```
avg_win_rate_by_role = df.groupby('Roles')
['T_WinRate'].mean().reset_index().sort_values(by='T_WinRate')

# Create a bar chart
fig = px.bar(avg_win_rate_by_role, x='Roles', y='T_WinRate',
             title='Average Win Rate by Hero Role',
             labels={'Roles': 'Hero Roles', 'T_WinRate': 'Average Win
Rate'},
             color='T_WinRate',
             color_continuous_scale='Viridis')

fig.update_layout(xaxis_title='Hero Roles', yaxis_title='Average Win
Rate')
fig.show()
```

Correlation between Picks and Wins

Is there a correlation between the number of times a hero is picked and their overall win rate?

```
fig = px.scatter(df, x='T_Picked', y='T_WinRate',
                 title='Correlation between Hero Picks and Win Rates',
                 labels={'T_Picked': 'Total Heroes Picked',
'T_WinRate': 'Win Rate'},
                 color='Hero',
                 color_continuous_scale='Viridis')

fig.update_layout(xaxis_title='Total Heroes Picked', yaxis_title='Win
```

```
Rate')
fig.show()
```

Banning Pattern

What is the overall ban percentage for each hero?

```
fig = px.bar(df.sort_values(by='T_BansPercentage'), x='Hero',
y='T_BansPercentage',
             title='Overall Ban Percentage for Each Hero',
             labels={'Hero': 'Heroes', 'T_BansPercentage': 'Ban
Percentage'},
             color='T_BansPercentage',
             color_continuous_scale='Viridis')

fig.update_layout(xaxis_title='Heroes', yaxis_title='Ban Percentage')
fig.show()
```

Picks and Bans Combined

How many heroes are frequently picked or banned (high pick and ban percentage)?

```
fig = px.bar(df.sort_values(by='T_PicksBansPercentage'), x='Hero',
y='T_PicksBansPercentage',
             title='Heroes with High Pick and Ban Percentages',
             labels={'Hero': 'Heroes', 'T_PicksBansPercentage': 'Pick
and Ban Percentage'},
             color='T_PicksBansPercentage',
             color_continuous_scale='Viridis')

fig.update_layout(xaxis_title='Heroes', yaxis_title='Pick and Ban
Percentage')
fig.show()
```

Role-wise Ban Percentage

Do certain hero roles have a higher ban percentage compared to others?

```
fig = px.bar(df.sort_values(by="T_BansPercentage",ascending=False),
x='Roles', y='T_BansPercentage',
             title='Ban Percentage by Hero Roles',
             labels={'Roles': 'Hero Roles', 'T_BansPercentage': 'Ban
Percentage'},
             color='T_BansPercentage',
             color_continuous_scale='Viridis')

fig.show()
```

Relationship between Picks, Bans, and Wins

Is there a correlation between the number of picks, bans, and overall win rates for a hero?

```
fig = px.scatter_matrix(df, dimensions=['T_Picked', 'T_Banned',
'T_WinRate'],
                        title='Scatter Plot Matrix of Picks, Bans, and
Win Rates',
                        color='T_WinRate',
                        color_continuous_scale='Viridis')

fig.show()

# Tambahkan renderer jika diperlukan
pio.renderers.default = "browser"

# Scatter plot pertama: Win Rate vs Total Games Played
fig = px.scatter(
    df,
    x='T_Picked',
    y='T_WinRate',
    title='Win Rate vs. Total Games Played',
    labels={'T_Picked': 'Total Games Played', 'T_WinRate': 'Win
Rate'},
    trendline='ols',  # Add a linear regression trendline
    color='T_WinRate',
    color_continuous_scale='Viridis'
)

fig.update_layout(xaxis_title='Total Games Played', yaxis_title='Win
Rate')
fig.show()

# Scatter plot kedua: Ban Percentage vs Total Games Played
fig = px.scatter(
    df,
    x='T_Picked',
    y='T_BansPercentage',
    title='Ban Percentage vs. Total Games Played',
    labels={'T_Picked': 'Total Games Played', 'T_BansPercentage': 'Ban
Percentage'},
    trendline='ols',  # Add a linear regression trendline
    color='T_BansPercentage',
    color_continuous_scale='Viridis'
)

fig.update_layout(xaxis_title='Total Games Played', yaxis_title='Ban
Percentage')
fig.show()
```