

```

import streamlit as st
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
from snowflake.snowpark.context import get_active_session

# Function to load data from Snowflake using Snowpark
@st.cache_data
def load_data():
    session = get_active_session()
    sql = "SELECT * FROM STRIPE.PUBLIC.CONVERSIONS"
    data = session.sql(sql).to_pandas()
    return data

# Load the dataset
data = load_data()

# Data Quality Analysis
def data_quality_analysis(data):
    st.subheader("Data Quality Analysis")
    st.markdown("In this section, we assess the quality of the data by checking for missing values, duplicate entries, and data types.")

    st.markdown("""
<style>
.grey-box {
    background-color: #f0f0f0;
    padding: 10px;
    border-radius: 5px;
    margin-bottom: 10px;
}
</style>
""", unsafe_allow_html=True)

    st.markdown("<div class='grey-box'><strong>Missing Values</strong></div>",
unsafe_allow_html=True)
    st.write(data.isnull().sum())

    st.markdown("<div class='grey-box'><strong>Duplicate Entries</strong></div>",
unsafe_allow_html=True)
    st.write(data.duplicated().sum())

    st.markdown("<div class='grey-box'><strong>Data Types</strong></div>",
unsafe_allow_html=True)
    st.write(data.dtypes)


# Simple Plot of Data Over Time

```

```

def plot_data_over_time(data):
    st.subheader("Data Overview")
    st.markdown("In this section, we provide an overview of the visits and conversions over time. This will help us understand the overall trend and any significant changes.")

    weekly_data = data.groupby("WEEK_DATE").sum().reset_index()
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=weekly_data["WEEK_DATE"], y=weekly_data["VISITS"], mode='lines+markers', name='Visits'))
    fig.add_trace(go.Scatter(x=weekly_data["WEEK_DATE"], y=weekly_data["CONVERSIONS"], mode='lines+markers', name='Conversions'))
    fig.update_layout(title='Weekly Visits and Conversions', xaxis_title='Week Date', yaxis_title='Count', hovermode='x unified')
    st.plotly_chart(fig)

    st.markdown("""
    <div class='grey-box'>
    <strong>Observations 

```

Conversion Rate Analysis

```

def conversion_rate_analysis(data):
    st.subheader("Conversion Rate Analysis")
    st.markdown("In this section, we analyze the conversion rate over time, by country, and by browser. The conversion rate is a crucial metric to understand the effectiveness of our website or campaign.")

```

```

    # Overall weekly conversion rate
    data['conversion_rate'] = data['CONVERSIONS'] / data['VISITS']
    weekly_conversion_rate = data.groupby("WEEK_DATE")[['CONVERSIONS', 'VISITS']].sum()
    weekly_conversion_rate['conversion_rate'] =
weekly_conversion_rate['CONVERSIONS'] / weekly_conversion_rate['VISITS']
    fig = go.Figure()
    fig.add_trace(go.Scatter(x=weekly_conversion_rate.index, y=weekly_conversion_rate['conversion_rate'], mode='lines+markers'))
    fig.update_layout(title='Weekly Conversion Rate', xaxis_title='Week Date', yaxis_title='Conversion Rate', hovermode='x unified')
    st.plotly_chart(fig)


```

```

    st.markdown("""
    <div class='grey-box'>


```

```

<strong>Observations  </strong>
<ul>
<li>The weekly conversion rate provides insight into how effectively visits are
turning into conversions.</li>
<li>There is a noticeable increase in conversion rate starting from week 6.</li>
</ul>
</div>
''''', unsafe_allow_html=True)

# Conversion rate by country
country_weekly_conversion_rate = data.groupby(['WEEK_DATE', 'COUNTRY'])
[['CONVERSIONS', 'VISITS']].sum()
country_weekly_conversion_rate['conversion_rate'] =
country_weekly_conversion_rate['CONVERSIONS'] /
country_weekly_conversion_rate['VISITS']
country_weekly_conversion_rate = country_weekly_conversion_rate.reset_index()

fig = go.Figure()
for country in country_weekly_conversion_rate['COUNTRY'].unique():
    subset =
country_weekly_conversion_rate[country_weekly_conversion_rate['COUNTRY'] ==
country]
    fig.add_trace(go.Scatter(x=subset['WEEK_DATE'], y=subset['conversion_rate'],
mode='lines+markers', name=country))
    fig.update_layout(title='Weekly Conversion Rate by Country', xaxis_title='Week
Date', yaxis_title='Conversion Rate', hovermode='x unified')
    st.plotly_chart(fig)

st.markdown('''''
<div class='grey-box'>
<strong>Observations  </strong>
<ul>
<li>Country AC shows a significant increase in visits starting from week 6, which is
impacting the overall conversion rate.</li>
<li>Other Countries have a relatively stable conversion rate, while country AD
shows fluctuations.</li>
</ul>
</div>
''''', unsafe_allow_html=True)

# Conversion rate by browser
browser_weekly_conversion_rate = data.groupby(['WEEK_DATE', 'BROWSER'])
[['CONVERSIONS', 'VISITS']].sum()
browser_weekly_conversion_rate['conversion_rate'] =
browser_weekly_conversion_rate['CONVERSIONS'] /
browser_weekly_conversion_rate['VISITS']
browser_weekly_conversion_rate =
browser_weekly_conversion_rate.reset_index()

fig = go.Figure()


```

```

for browser in browser_weekly_conversion_rate['BROWSER'].unique():
    subset =
browser_weekly_conversion_rate[browser_weekly_conversion_rate['BROWSER'] ==
browser]
    fig.add_trace(go.Scatter(x=subset['WEEK_DATE'], y=subset['conversion_rate'],
mode='lines+markers', name=browser))
    fig.update_layout(title='Weekly Conversion Rate by Browser', xaxis_title='Week
Date', yaxis_title='Conversion Rate', hovermode='x unified')
    st.plotly_chart(fig)


```

```

st.markdown("""
<div class='grey-box'>
<strong>Observations 

```

```

# Top and lower performers
st.markdown("### Top and Lower Performers ")
top_performers = data.groupby(['COUNTRY', 'BROWSER'])
['conversion_rate'].mean().sort_values(ascending=False).head(10)
st.markdown("#### Top Performers")
st.write(top_performers)


```

```

lower_performers = data.groupby(['COUNTRY', 'BROWSER'])
['conversion_rate'].mean().sort_values(ascending=True).head(10)
st.markdown("#### Lower Performers")
st.write(lower_performers)

```

```

st.markdown("""
<div class='grey-box'>
<strong>Observations 

```

```

# Enhanced Correlation Analysis
def enhanced_correlation_analysis(data):

```

```
st.subheader("Enhanced Correlation Analysis")
st.markdown("In this section, we analyze the correlation between visits and conversions to understand the relationship between these two metrics.")
```

```
# Pair Plot
```

```
st.markdown("## Pair Plot Analysis 🇫🇷")
```

```
st.markdown("A pair plot provides a visualization of the pairwise relationships between variables, allowing us to observe potential correlations and distributions.")
```

```
pairplot_fig = sns.pairplot(data[['VISITS', 'CONVERSIONS', 'WEEK_DATE', 'COUNTRY']])
```

```
st.pyplot(pairplot_fig)
```

```
st.markdown("""
```

```
<div class='grey-box'>
```

```
<strong>Observations 📝 </strong>
```

```
<ul>
```

```
<li>The pair plot shows a positive correlation between visits and conversions, indicating that as the number of visits increases, the number of conversions also increases.</li>
```

```
<li>There are distinct patterns visible for different countries and weeks.</li>
```

```
</ul>
```

```
</div>
```

```
""", unsafe_allow_html=True)
```

```
# Analysis for Country AC
```

```
def country_ac_analysis(data):
```

```
    st.subheader("Country AC Analysis")
```

```
    st.markdown("In this section, we take a closer look at the significant increase in visits and conversions for country AC.")
```

```
    country_ac_data = data[data['COUNTRY'] == 'AC']
```

```
    weekly_ac_data = country_ac_data.groupby('WEEK_DATE').sum().reset_index()
```

```
    fig = go.Figure()
```

```
    fig.add_trace(go.Scatter(x=weekly_ac_data['WEEK_DATE'],
                             y=weekly_ac_data['VISITS'], mode='lines+markers', name='Visits'))
```

```
    fig.add_trace(go.Scatter(x=weekly_ac_data['WEEK_DATE'],
                             y=weekly_ac_data['CONVERSIONS'], mode='lines+markers', name='Conversions'))
```

```
    fig.update_layout(title='Country AC: Weekly Visits and Conversions',
                      xaxis_title='Week Date', yaxis_title='Count', hovermode='x unified')
```

```
    st.plotly_chart(fig)
```

```
st.markdown("""
```

```
<div class='grey-box'>
```

```
<strong>Observations 📝 </strong>
```

```
<ul>
```

```
<li>There is a significant increase in visits and conversions starting from week 6.</li>
```

```
</li>This spike is driving the overall increase in the global conversion rate.</li>
</ul>
</div>
""" , unsafe_allow_html=True)
```

```
st.markdown("""
<div class='grey-box'>
<strong>Possible Theories for the Increase:</strong>
<ul>
<li><strong>Marketing Campaigns</strong>: A new marketing campaign may
have been launched in week 6, leading to increased traffic and conversions.</li>
<li><strong>Product Launches or Updates</strong>: A new product or feature
might have been introduced, attracting more users and improving conversion
rates.</li>
<li><strong>Seasonal Effects</strong>: Seasonal promotions or events could
have increased user activity during this period.</li>
<li><strong>Technical Improvements</strong>: Enhancements in website
performance or user experience may have led to higher conversions.</li>
</ul>
</div>
""" , unsafe_allow_html=True)
```

Final Recommendation Section

```
def final_recommendations():
    st.subheader("Final Recommendations")
    st.markdown("""
    Based on the findings from the analysis, here are the recommendations for
    improving conversion rates and understanding the root causes behind the observed
    trends:
```

Overall Recommendations:

```
- **Aside from the increase created by Country AC, there is no significant
improvement in the conversion rate with the current data.**
- **Action Items:**
    - Conduct a deeper analysis to understand the underlying factors affecting
    conversion rates in countries other than AC.
    - Perform A/B testing on different strategies to see which ones yield better
    conversion rates.
```

Country-Specific Recommendations:

```
- **Country AC:**
    - **Observation:** Country AC exhibits a significant increase in conversion rate
    starting from week 6, likely influencing the global conversion rate.
    - **Action Items:**
        - Follow up with the team to confirm the cause of the change in the data.
        - If it's due to a marketing campaign, consider replicating this campaign in other
        countries.
        - Analyze the specific elements of the campaign that led to the increase and
```

apply those insights to other markets.

- **Country AB:**
 - **Observation:** Country AB has a stable conversion rate.
 - **Action Items:**
 - Reach out to the responsible team to understand what is driving the stability.
 - Explore additional data or macroeconomic factors that could be influencing this stability.

- **Country AD:**
 - **Observation:** Country AD shows more variability in conversion rates.
 - **Action Items:**
 - Investigate the factors contributing to this variability.
 - Collaborate with the country team to identify potential causes and solutions.
 - Use more granular data to pinpoint specific issues or opportunities for improvement.

Performance Analysis:

- **Top Performers:**
 - **Observation:** The highest conversion rates are observed in Country AC using Chrome.
 - **Action Items:**
 - Use the insights from the top-performing segments to develop best practices.
 - Apply successful strategies from these segments to underperforming ones.

- **Lower Performers:**
 - **Observation:** The lowest conversion rates are found in Countries AB and AD using less popular browsers.
 - **Action Items:**
 - Focus on improving the user experience in lower-performing browsers.
 - Implement targeted interventions in countries AB and AD to address specific challenges.

Correlation Analysis:

- **Observation:** There is a positive correlation between visits and conversions, indicating that higher visits generally lead to higher conversions. Distinct patterns are visible for different countries and weeks, suggesting varying factors influencing conversion rates.

- **Action Items:**
 - Use segmentation in data analysis to capture the unique patterns and factors influencing conversion rates in different markets.
 - Consider developing market-specific strategies rather than a one-size-fits-all approach to improve conversion rates.

```
""" , unsafe_allow_html=True)
```

Streamlit App Layout

```
def main():  
    st.title("Conversion Rate Analysis")  
    st.write("Unlocking Conversion Potential: Data-Driven Insights for Stripe")
```

```
# Introduction and Data Quality
st.header("Introduction")
st.write("This app provides an analysis of visits and conversions data over time,
broken down by country and browser.")

data_quality_analysis(data)

plot_data_over_time(data)

# Conversion Rate Analysis
conversion_rate_analysis(data)

# Enhanced Correlation Analysis
enhanced_correlation_analysis(data)

# Analysis for Country AC
country_ac_analysis(data)

# Final Recommendations
final_recommendations()

# Debug: Display the data to ensure it's loaded correctly
st.write("### Loaded Data")
st.write(data.head())

if __name__ == "__main__":
    main()
```