



Swift Traq

CUSTOMER LIFETIME VALUE

PYTHON 



Contents

03

Project Objectives

04

Project Scope

05

Background Context

06

Projects



Project Objectives



- To calculate CLV for individual customers and customer segments.
- To identify key factors driving customer profitability.
- To create compelling visualizations to communicate insights effectively.
- To support decision-making for marketing and sales strategies.



Project Scope

- Sales transactions, customer demographics, and purchase behaviors.
- Historical data trends to predict future customer value.
- Advanced visualizations using Plotly Express to illustrate insights.

Background Context



- **Business Overview:**

The AdventureWorks company operates in a competitive retail environment, offering a wide range of products. Understanding customer behavior and optimizing lifetime value are pivotal to driving sustained growth.

- **Problem Statement**

Retaining high-value customers and identifying profitable segments pose significant challenges. Traditional methods fail to leverage the full potential of data, leading to missed opportunities in targeted marketing and revenue optimization.



Sales Table

```
df_sales.head(3)
```

	ProductKey	OrderDateKey	DueDateKey	ShipDateKey	CustomerKey	PromotionKey	CurrencyKey	SalesTerritoryKey	SalesOrderNumber	SalesOrderLineNumber	...	ProductStandardCost	TotalProductCost	SalesOrderSubTotal
0	310	20101229	20110110	20110105	21768	1	19	6	SO-43697	1	...	2171.2942	2171.2942	2171.2942
1	346	20101229	20110110	20110105	28389	1	39	7	SO-43698	1	...	1912.1544	1912.1544	1912.1544
2	346	20101229	20110110	20110105	25863	1	100	1	SO-43699	1	...	1912.1544	1912.1544	1912.1544

3 rows \times 28 columns

Customer Table

	CustomerKey	GeographyKey	CustomerAlternateKey	Title	FirstName	MiddleName	LastName	NameStyle	BirthDate	MaritalStatus	...	EnglishOccupation	SpanishOccupation	FrenchOccupation	HouseOwner
0	11000	26	AW00011000	NaN	Jon	V	Yang	0	1971-10-06	M	—	Professional	Profesional	Cadre	
1	11001	37	AW00011001	NaN	Eugene	L	Huang	0	1976-05-10	S	—	Professional	Profesional	Cadre	
2	11002	31	AW00011002	NaN	Ruben	NaN	Torres	0	1971-02-09	M	—	Professional	Profesional	Cadre	

3 rows × 20 columns



Date Table

	DateKey	FullDateAlternateKey	DayNumberOfWeek	EnglishDayNameOfWeek	SpanishDayNameOfWeek	FrenchDayNameOfWeek	DayNumberOfMonth	DayNumberOfYear	WeekNumberOfYear	EnglishMonthName	SpanishMonthName
0	20050101	2005-01-01	7	Saturday	Sábado	Samedi	1	1	1	January	Enero
1	20050102	2005-01-02	1	Sunday	Domingo	Dimanche	2	2	2	January	Enero
2	20050103	2005-01-03	2	Monday	Lunes	Lundi	3	3	2	January	Enero

Geography Table

df_geography.head(3)

	GeographyKey	City	StateProvinceCode	StateProvinceName	CountryRegionCode	EnglishCountryRegionName	SpanishCountryRegionName	FrenchCountryRegionName	PostalCode	SalesTerritoryKey	IpAddress
0	1	Alexandria	NSW	New South Wales	AU	Australia	Australia	Australie	2015	9	
1	2	Coffs Harbour	NSW	New South Wales	AU	Australia	Australia	Australie	2450	9	
2	3	Darlinghurst	NSW	New South Wales	AU	Australia	Australia	Australie	2010	9	



Swift Traq

CLV Related Metrics

Aggregate sales by customer, compute CLV-related metrics

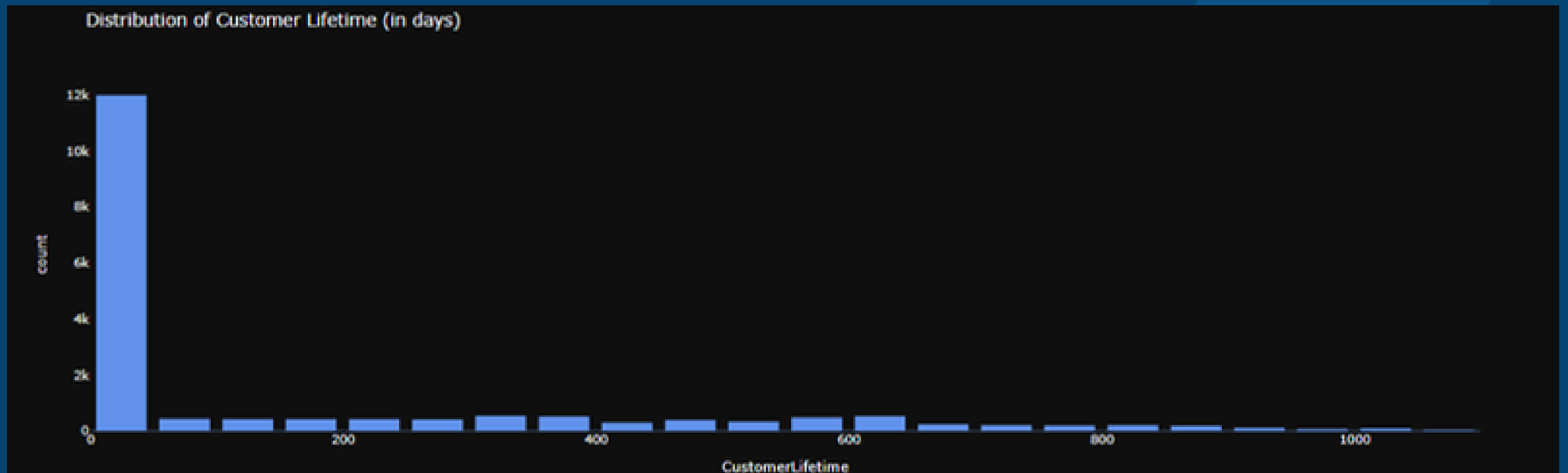
```
[ ] df_clv = df.groupby('CustomerKey').agg(  
    TotalRevenue = ('SalesAmount', 'sum'),  
    TotalOrders = ('SalesOrderNumber', 'nunique'),  
    AvgOrderValue=('SalesAmount', 'mean'),  
    LastPurchaseDate=('OrderDate', 'max'),  
    FirstPurchaseDate=('OrderDate', 'min')  
) .reset_index()  
  
df_clv['CustomerLifetime'] = (df_clv['LastPurchaseDate'] - df_clv['FirstPurchaseDate']).dt.days  
df_clv = pd.merge(df_clv, df_customer[['CustomerKey', 'GeographyKey', 'FullName']], on='CustomerKey')  
df_clv = pd.merge(df_clv, df_geography[['GeographyKey', 'EnglishCountryRegionName', 'CountryRegionCode']], on='GeographyKey')
```




Swift Traq

Distribution of Customer Lifetime (days)

```
fig1 = px.histogram(  
    df_clv,  
    x='CustomerLifetime',  
    nbins=30,  
    title='Distribution of Customer Lifetime (in days)',  
    template='plotly_dark',  
    color_discrete_sequence=['#6495ED'] ##636EFA  
)  
fig1.update_layout(  
    xaxis=dict(showgrid=False),  
    yaxis=dict(showgrid=False),  
    bargap=0.2  
)  
fig1.show()
```





Swift Traq

Revenue Contribution by Region

```
fig2 = px.bar(  
    df_civ.groupby('EnglishCountryRegionName')['TotalRevenue'].sum().reset_index(),  
    x='EnglishCountryRegionName',  
    y='TotalRevenue',  
    title='Revenue Contribution by Region',  
    labels={'TotalRevenue': 'Total Revenue', 'EnglishCountryRegionName': 'Country'},  
    template='plotly_dark',  
    color='TotalRevenue',  
    color_continuous_scale='blues'  
)  
fig2.update_layout(  
    xaxis=dict(showgrid=False),  
    yaxis=dict(showgrid=False)  
)  
fig2.show()
```

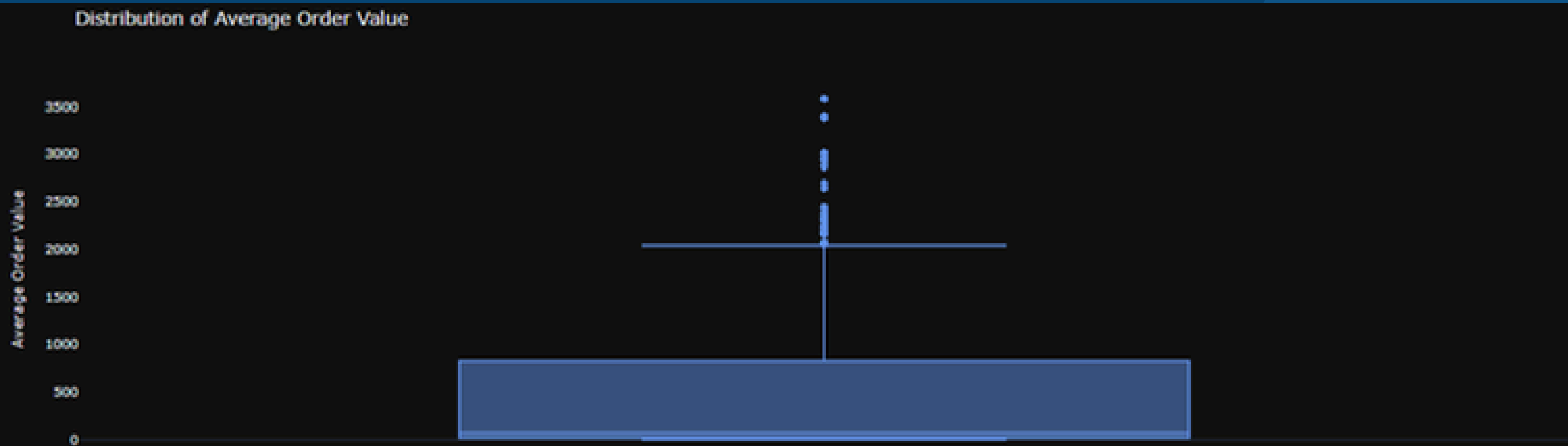




Swift Traq

Distribution of Average Order Value

```
] fig = px.box(  
    df_clv,  
    y='AvgOrderValue',  
    title='Distribution of Average Order Value',  
    labels={'AvgOrderValue': 'Average Order Value'},  
    template='plotly_dark',  
    color_discrete_sequence=['#6495ED']  
)  
fig.update_layout(  
    xaxis=dict(showgrid=False),  
    yaxis=dict(showgrid=False)  
)  
fig.show()
```

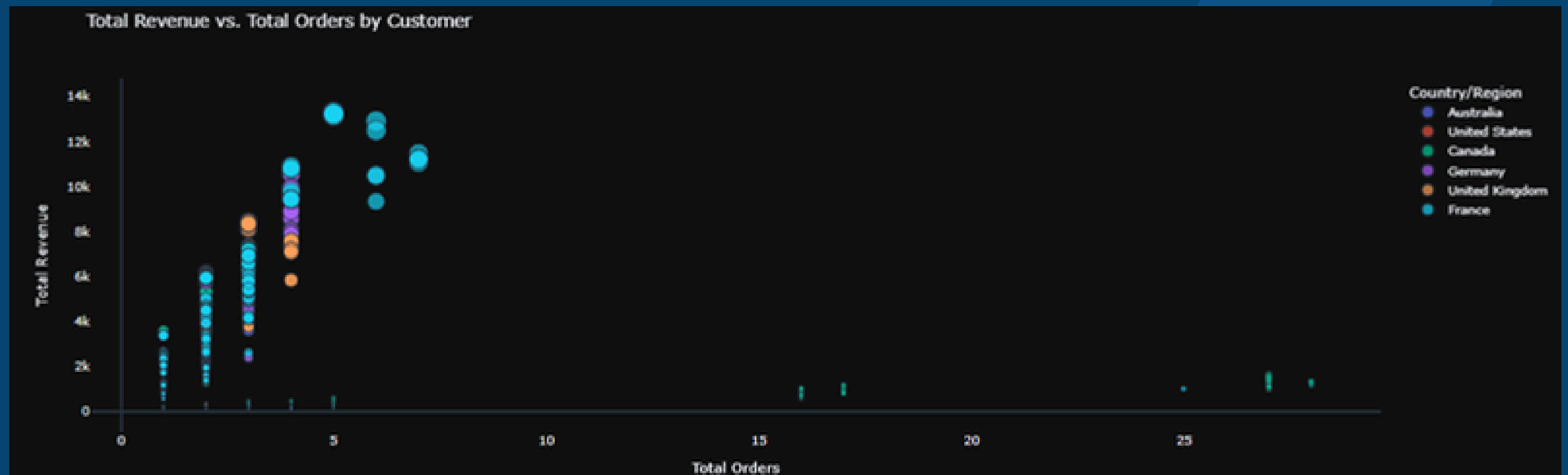




Swift Traq

Total Revenue Vs. Total Orders by Customers

```
] fig4 = px.scatter(  
    df_clv,  
    x='TotalOrders',  
    y='TotalRevenue',  
    size='TotalRevenue',  
    color='EnglishCountryRegionName',  
    title='Total Revenue vs. Total Orders by Customer',  
    labels={'TotalOrders': 'Total Orders', 'TotalRevenue': 'Total Revenue'},  
    template='plotly_dark'  
)  
fig4.update_layout(  
    xaxis=dict(showgrid=False),  
    yaxis=dict(showgrid=False),  
    legend_title_text='Country/Region'  
)  
fig4.show()
```

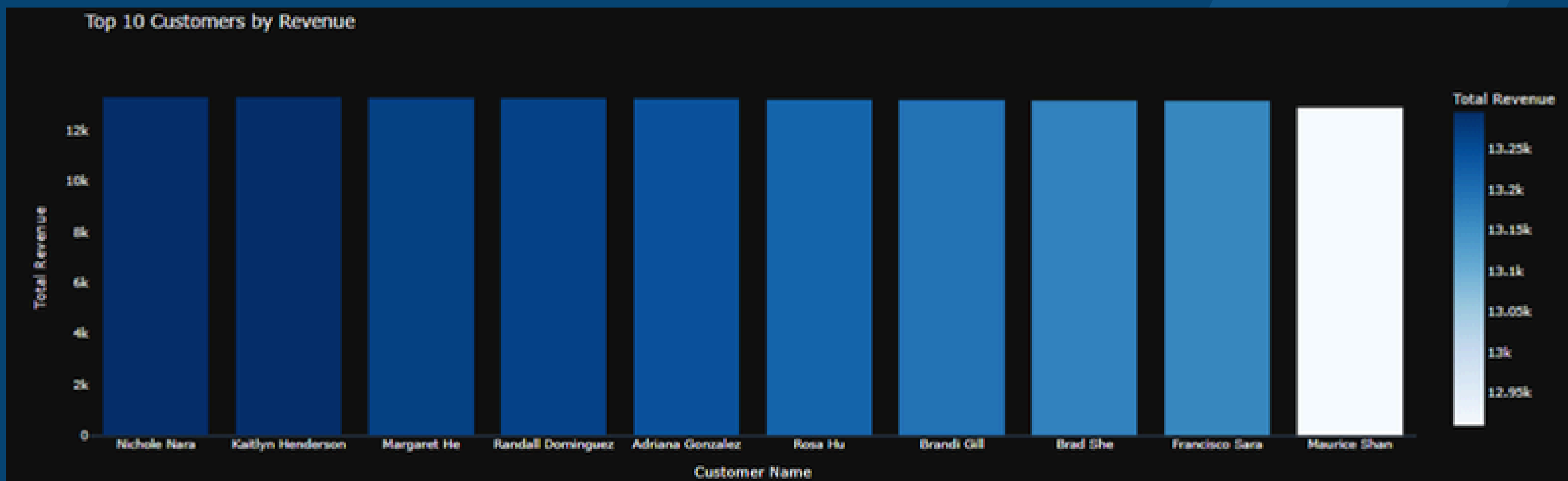




Swift Traq

Top 10 Customers by Revenue

```
] df_top_customers = df_civ.nlargest(10, 'TotalRevenue')
fig6 = px.bar(
    df_top_customers,
    x='FullName',
    y='TotalRevenue',
    title='Top 10 Customers by Revenue',
    labels={'FullName': 'Customer Name', 'TotalRevenue': 'Total Revenue'},
    template='plotly_dark',
    color='TotalRevenue',
    color_continuous_scale='Blues' #Bluered
)
fig6.update_layout(
    xaxis=dict(showgrid=False),
    yaxis=dict(showgrid=False)
)
fig6.show()
```



Benefit Of The Project

10+

1000+

- Improved Decision-Making: Data-driven insights for targeted marketing campaigns.
- Customer Retention: Strategies to retain high-value customers based on predictive analytics.
- Revenue Growth: Identification of key revenue drivers and opportunities for upselling.
- Enhanced Visualization: Interactive, clear, and professional plots for stakeholders.



**THANK
YOU**