Data Analyst Professional Practical Exam Submission

You can use any tool that you want to do your analysis and create visualizations. Use this template to write up your summary for submission.

You can use any markdown formatting you wish. If you are not familiar with Markdown, read the Markdown Guide before you start.

```
import pandas as pd
sales = pd.read_csv('product_sales.csv')
sales
```

		sales...	customer_id			years_as_cus...	nb_site_...	state
719	1	Email	ae6df2ea-c55a-4251-803f-3daaf49d3f94	9	85.49	39	22	Te
5567	2	Email	38a2040b-ab8d-4522-a453-07b3d1ff47af	10	101.41	39	25	Te
6890	5	Email + Call	ace324ea-fb31-4644-8ad6-12e41e76b1a8	12		39	29	N
6180	1	Email	41de5549-3522-4345-9ec5-39c5834531dc	9	93.34	38	21	N
6944	3	Call	673e5550-c258-4b9b-985b-eca9427a8d55	8	40.64	38	25	Te
8379	1	Email	01b729b7-0371-4207-bfeb-a346d9dee726	8	80.77	37	16	W
32	5	Call	57a6a6fd-842f-4b28-8033-b2137096f086	11	55.74	36	26	C
671	1	Email	93b798f3-9c6a-42d1-b339-b6b69c44f14a	8	83.71	36	22	Te
3747	2	Call	4cab6233-f827-4d2d-9f55-199eb79e528a	8	42.15	36	29	Io
11526	1	Email	22f90018-dfe2-429d-ab9f-baf6676f209b	9	85.62	36	25	G
3270	1	Email	6888c315-6537-412f-b989-1a6c19d942a7	9	92.9	35	29	Te
3691	1	Email	a5a30979-0947-4709-8f35-ac6fb9228b0e	8	82.66	35	16	G
7300	3	Call	1e0d242a-b097-44a4-bc1a-1eb3a7aa6418	9	42.6	35	28	M
1676	1	Email	4ddd4f29-01b2-486d-b7dd-54cb1d45cab5	8	83.13	34	23	O
2011	1	Email	414bf270-0f65-4e0d-867a-6d41a27f1bf7	9		34	19	N
4524	3	Email	9b568a34-0f7a-46e6-99d6-119dfefd31ae	9	89.27	34	27	N

Rows: 12,500 ⚠ truncated from 15,000 rows Expand Table

```
sales["sales_method"] = sales["sales_method"].replace({
    "email + call": "Email + Call",
    "email": "Email",
    "em + call": "Email + Call"
})

# email_counts = sales[sales["sales_method"]=="Email"]["customer_id"].value_counts()
# email_counts

cust_counts = sales.groupby("sales_method")["customer_id"].nunique()
cust_counts
```

sales_method	customer_id
Call	
Email	
Email + Call	

Rows: 3 Expand Table

#Data validation Initial Observations: Data Types:

All columns have appropriate data types.

Missing Values:

revenue has 1,074 missing values.

Unique Constraints:

customer_id appears to be unique (I'll verify this next).

Range Checks:

week should be positive.

years_as_customer should not exceed 40 (since the company was founded in 1984).

nb_sold, revenue, and nb_site_visits should be non-negative.

Format Checks:

sales_method should have only three categories.

state should be a valid US state.

I'll now perform deeper validation.

Data Validation Findings: Unique customer_id ✅ - All customer IDs are unique.

Valid week values ✅ - All values are positive.

Valid revenue values ✅ - No negative values.

Valid years_as_customer - All less than 40.

Valid nb_sold and nb_site_visits ✅ - No negative values.

Sales Method Issues ❌ - Inconsistent labels:

Expected: "Email", "Email + Call", "Call"

Found: Variants like "em + call" and "email" (case inconsistency).

State Validity ✅ - All states appear valid.

So, I tried to replace case incosistent labels for sales_method to the correct! Then I tried to calculate number of customers for each sales_method.

```
sales["revenue"].max()
```

```
238.32
```

```
sales["revenue"].min()
```

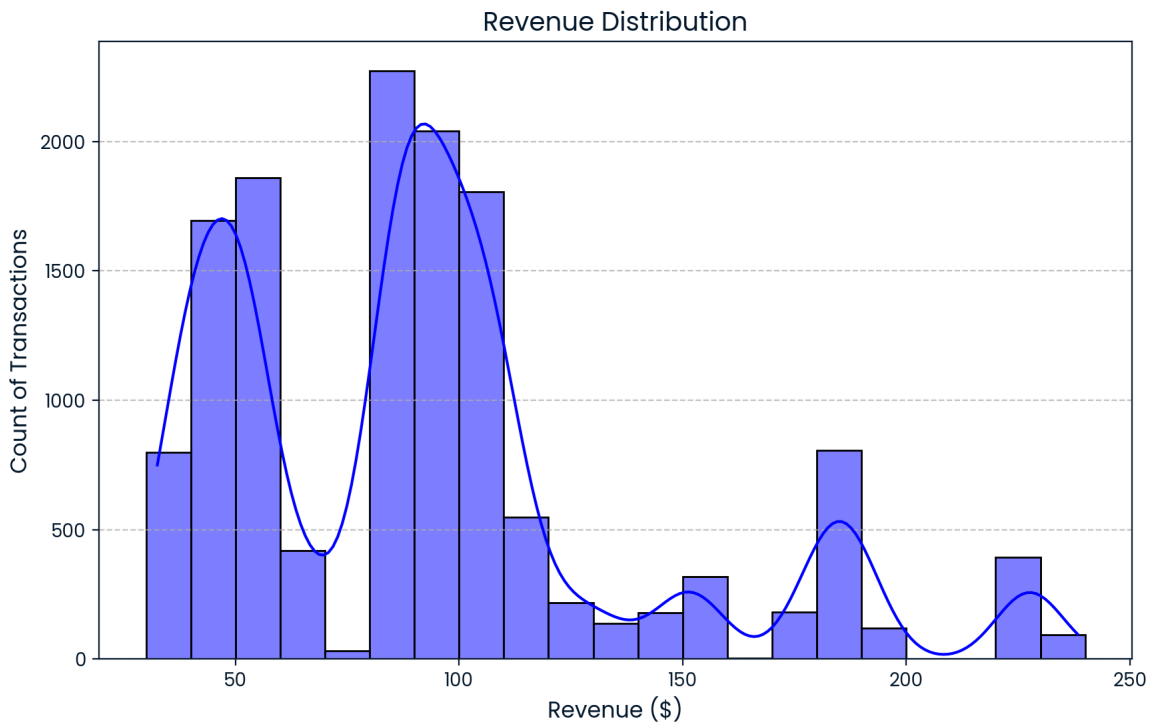
```
32.54
```

```
sales["revenue"].describe()
```

index	revenue
count	
mean	9
std	4
min	
25%	
50%	
75%	
max	

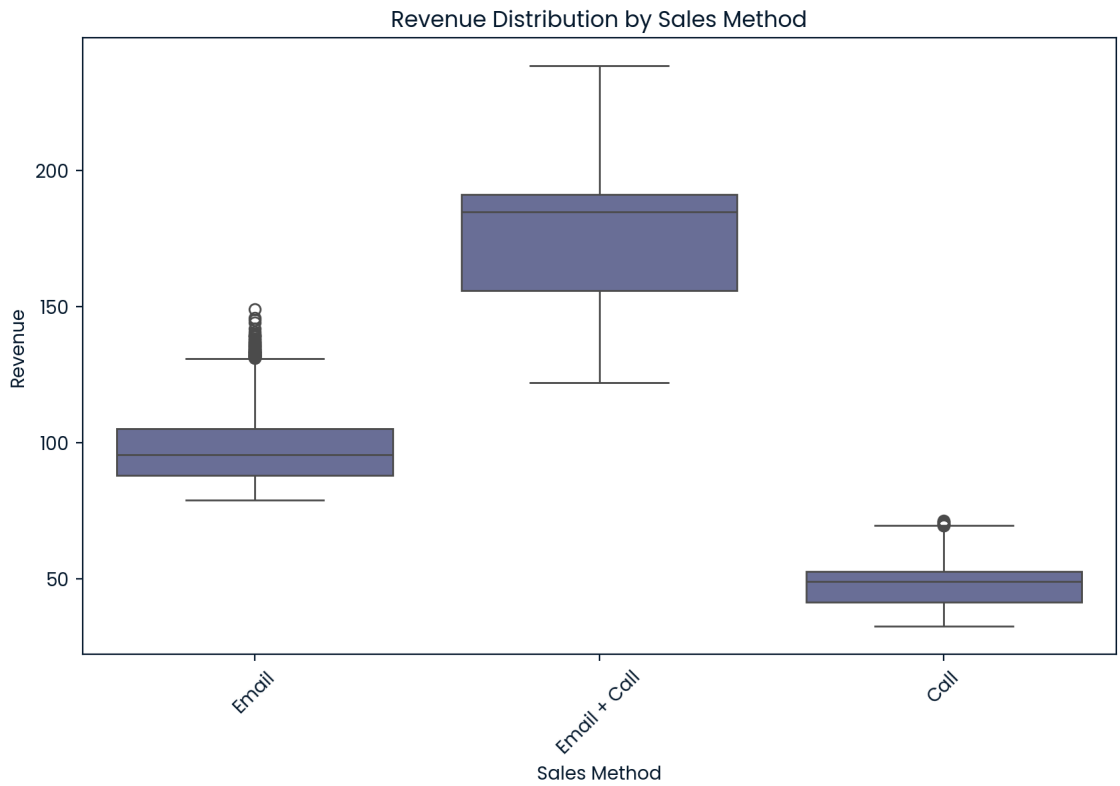
Rows: 8 Expand Table

```python
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd

# Ensure df is loaded (uncomment if needed)
# df = pd.read_csv("your_file.csv")

# Ensure revenue column is numeric and drop NaNs
sales["revenue"] = pd.to_numeric(sales["revenue"], errors="coerce")  # Convert to numeric
sales = sales.dropna(subset=["revenue"])  # Drop missing values

# Define revenue bins (30-40, 40-50, etc.)
bin_width = 10
max_revenue = sales["revenue"].max() if not sales["revenue"].isna().all() else 100  # Avoid errors if empty
bins = np.arange(30, max_revenue + bin_width, bin_width)

# Create histogram
plt.figure(figsize=(10, 6))
sns.histplot(sales["revenue"], bins=bins, kde=True, color="blue")

# Labels & title
plt.title("Revenue Distribution", fontsize=14)
plt.xlabel("Revenue ($)", fontsize=12)
plt.ylabel("Count of Transactions", fontsize=12)
plt.grid(axis="y", linestyle="--", alpha=0.7)

# Show plot
plt.show()
```



#Data Viz1 As we see it here, it looks like there might be missing data in-between some areas like 70-80, 160-170 and 200-220 bins are really low or no transactions at all for this area, which shows there are less data or these areas might be the borders for the data meaning some factors are separating the data into multiple sectors. Furthermore, there are many transactions which revenues are less than $110, eventhemaxrevenuegoestoaround240$. The most of the transactions has been 80-90 bin with about 2500 transactions, while most of the other transactions were made between 90 to 110 and 40 to 60 as well.

```
plt.figure(figsize=(10, 6))
sns.boxplot(data=sales, x="sales_method", y="revenue")
plt.title("Revenue Distribution by Sales Method")
plt.xlabel("Sales Method")
plt.ylabel("Revenue")
plt.xticks(rotation=45)
plt.show()
```



#Data Viz2 As we see above, for each sales_method revenues were different: Call - starting from about 25 to 70 Email - starting from around 75 to 150 Email + Call - starting from about 125 to 240

Then, in reality there is difference between revenues of each sales_method and sales_method is dividing the revenues data into 3 categories!

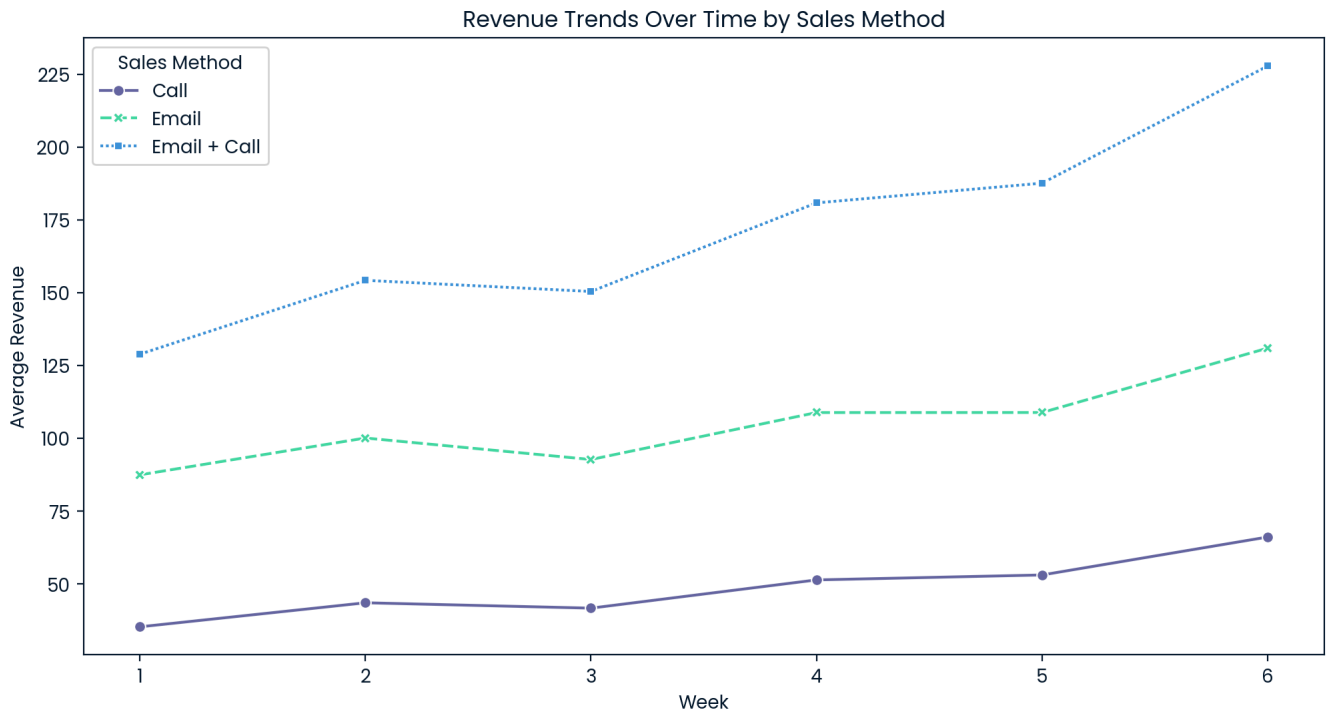
```
sales.groupby("sales_method")["revenue"].describe()
```

sales...		mean	std					
Call	4781	47.5974670571	8.609898986	32.54	41.47	49.07	52.68	71.36
Email	6922	97.1276841953	11.2104690128	78.83	87.88	95.58	105.17	148.97
Email + Call	2223	183.6512325686	29.0839242965	122.11	155.775	184.74	191.11	238.32

Rows: 3 Expand Table

```
sales_grouped = sales.groupby(["week", "sales_method"])["revenue"].mean()
```

```python
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd

# Group data by week and sales method, calculating average revenue
df_grouped = sales.groupby(["week", "sales_method"])["revenue"].mean().reset_index()

# Plot the revenue trend over time
plt.figure(figsize=(12, 6))
sns.lineplot(data=df_grouped, x="week", y="revenue", hue="sales_method", style="sales_method", markers=True)

# Chart labels and title
plt.title("Revenue Trends Over Time by Sales Method")
plt.xlabel("Week")
plt.ylabel("Average Revenue")
plt.legend(title="Sales Method")

# Display the plot
plt.show()
```



#Data Viz3 Overall view is showing that revenues for each sales_methods got increased by time and during the weeks at least the average revenue doubled for each sales_method.

📝 Task List

Your written report should include written text summaries and graphics of the following:

- Data validation:
  - Describe validation and cleaning steps for every column in the data
- Exploratory Analysis:
  - Include two different graphics showing single variables only to demonstrate the characteristics of data
  - Include at least one graphic showing two or more variables to represent the relationship between features
  - Describe your findings
- Definition of a metric for the business to monitor
  - How should the business use the metric to monitor the business problem
  - Can you estimate initial value(s) for the metric based on the current data
- Final summary including recommendations that the business should undertake

Start writing report here..

#Data Viz1 As we see it here, it looks like there might be missing data in-between some areas like 70-80, 160-170 and 200-220 bins are really low or no transactions at all for this area, which shows there are less data or these areas might be the borders for the data meaning some factors are separating the data into multiple sectors. Furthermore, there are many transactions which revenues are less than $110, eventhemaxrevenuegoestoaround240$. The most of the transactions has been 80-90 bin with about 2500 transactions, while most of the other transactions were made between 90 to 110 and 40 to 60 as well.

#Data Viz2 As we see above, for each sales_method revenues were different: Call - starting from about 25 to 70 Email - starting from around 75 to 150 Email + Call - starting from about 125 to 240

Then, in reality there is difference between revenues of each sales_method and sales_method is dividing the revenues data into 3 categories!

#Data Viz3 Overall view is showing that revenues for each sales_methods got increased by time and during the weeks at least the average revenue doubled for each sales_method.

I guess as much more specific data we have, that much more detailed research and recommendation we can give, while here most useful data probably is the count of transactions by sales_method and the average revenue by sales_method.

According to all the data, I suggest to use more Email + Call sales_method. As it is causing to get more revenue and it does not take too much effort. Furthermore, it is better to keep Email is also as a side option. Because it does not take as much time and effort as the other 2 options, while it caused second biggest average revenue from sales_methods. Maybe, it is better to send emails always and give calls after some time, that actually gives a client to learn about the product more, to know the product rather than directly calling and having difficulty to explain everything in the first time while customer also do not always understand it perfectly which causes less revenue.

✅ When you have finished...

- Publish your Workspace using the option on the left
- Check the published version of your report:
  - Can you see everything you want us to grade?
  - Are all the graphics visible?
- Review the grading rubric. Have you included everything that will be graded?
- Head back to the **Certification Dashboard** to submit your practical exam report and record your presentation