

PROFESSIONAL DATA ANALYSIS CERTIFICATE PRESENTATION

PRESENTED BY

SREYA SIRIVELLA

PRODUCT _ SALES



CONTENT

1. DATA VALIDATION AND CLEANING

2. EXPLORE ANALYSIS

3. METRIC

4. CONCLUSION



PART 1: DATA VALIDATION AND CLEANING

product_sales.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15000 entries, 0 to 14999
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   week            15000 non-null  int64
1   sales_method    15000 non-null  object
2   customer_id     15000 non-null  object
3   nb_sold         15000 non-null  int64
4   revenue         13926 non-null  float64
5   years_as_customer 15000 non-null  int64
6   nb_site_visits  15000 non-null  int64
7   state           15000 non-null  object
dtypes: float64(1), int64(4), object(3)
memory usage: 937.6+ KB
```

data.isna().sum()

Table Chart Filter Columns Search	
index	0
week	0
sales_method	0
customer_id	0
nb_sold	0
revenue	1074
years_as_customer	0
nb_site_visits	0
state	0
Rows: 8 Expand Table	

data.nunique()

Table Chart Filter Columns Search	
index	0
V index (string)	6
sales_method	5
customer_id	15000
nb_sold	10
revenue	6743
years_as_customer	42
nb_site_visits	27
state	50
Rows: 8 Expand Table	

% missing per column
(data.isna().mean() * 100).round(2)

Table Chart Filter Columns Search	
index	0
week	0
sales_method	0
customer_id	0
nb_sold	0
revenue	7.16
years_as_customer	0
nb_site_visits	0
state	0
Rows: 8 Expand Table	

CHECK DATA VALIDATION

During validation I discovered that the **revenue** field has 1 074 missing entries (about 7% of all records), so I flagged and imputed those values before moving on. In the **sales_method** column I observed five distinct labels—yet the business only uses three channels—indicating typos or inconsistent naming that I standardized. Finally, I found several **years_as_customer** values above 40. Since the company launched in 1984 (40 years ago), any tenure beyond that (for example, 63 years) is clearly erroneous, so I treated those as outliers. These checks ensure my subsequent analysis rests on clean, reliable data.

1.Unique-Value Cleaning & Outliers Removal

2. Data Cleaning 2.1 Standardize sales_method

```
# Expecting exactly 3 methods; fix typos/case
data['sales_method'] = data['sales_method'].str.strip().str.lower().replace({
    'em + call':'email + call',
    'email':'email'
})
```

```
1 # Capitalize consistently
2 data['sales_method'] = data['sales_method'].str.title()
3
4 print("After cleaning:", data['sales_method'].unique())
```

After cleaning: ['Email' 'Email + Call' 'Call']

2.2 Outlier Removal in years_as_customer

```
# Flag implausible values (>40 years)
outliers = data[data['years_as_customer'] > 40]
print("Rows with years_as_customer>40:", len(outliers))
```

Rows with years_as_customer>40: 2

```
# Drop them
data = data[data['years_as_customer'] <= 40].copy()
print("Records after filtering:", len(data))
```

Records after filtering: 14998

29,240.68

When I first examined the **sales_method** column, I found five distinct labels—Email, Email + Call, Call, em + call, and email—even though the business only uses three channels. By normalizing case and correcting typos (em + call → Email + Call, email → Email), I reduced it to exactly the three valid methods: **Email**, **Email + Call**, and **Call**.

Similarly, because the company launched in 1984 (40 years ago), any years_as_customer value above 40 is impossible. I identified just two records with tenures exceeding 40 years and dropped them. Given the dataset contains 15 000 rows, removing those two outliers won't introduce bias and ensures my analysis rests on realistic customer tenures.

1.Outlier Detection

```
# Summary stats
data[['nb_sold', 'revenue', 'nb_site_visits', 'years_as_customer']].describe()
```

[20]

Table

Chart

Filter

Columns

Search

index	nb_sold	revenue	nb_site_visits	years_as_customer
count	14998	14998	14998	14998
mean	10.0846779571	95.5686191492	24.9907987732	4.9592612348
std	1.8123339033	47.9872004252	3.5010998589	5.0112370136
min	7	32.54	12	0
25%	9	52.65	23	1
50%	10	90.95	25	3
75%	11	107.75	27	7
max	16	238.32	41	39

Rows: 8



Expand Table

```
# IQR method example for nb_site_visits
Q1 = data['nb_site_visits'].quantile(0.25)
Q3 = data['nb_site_visits'].quantile(0.75)
IQR = Q3 - Q1
extreme = data[(data['nb_site_visits'] < Q1 - 1.5*IQR) |
               (data['nb_site_visits'] > Q3 + 1.5*IQR)]
print("Extreme site visits:", len(extreme))
```

[21]

Extreme site visits: 227

I generated summary statistics for nb_site_visits (median 25, IQR 4) and then applied the IQR rule—flagging any values below $Q1 - 1.5 \times IQR$ or above $Q3 + 1.5 \times IQR$. That process uncovered 227 records with unusually low or high visit counts, which I'll treat as outliers to prevent them from skewing my analysis.

1. Handling Missing Revenue

```
# Proportion missing
pct_missing = data['revenue'].isna().mean() * 100
print(f"Revenue missing: {pct_missing:.2f}%")
```

Revenue missing: 7.16%

```
# Create a flag/category
data['revenue_category'] = data['revenue'].isna().map({True:'Null', False:'Not_Null'})
```

```
# Stratified median imputation by sales_method
def impute_rev(group):
    med = group['revenue'].median()
    group['revenue'] = group['revenue'].fillna(med)
    return group

data = data.groupby('sales_method').apply(impute_rev).reset_index(drop=True)
print("Remaining missing:", data['revenue'].isna().sum())
```

Remaining missing: 0

```
# Test
chi2, p, _, _ = chi2_contingency(ct)
print(f"Chi2={chi2:.2f}, p-value={p:.3f}")
```

```
if p < 0.05:
    print("→ Significant association")
else:
    print("→ No significant association")
```

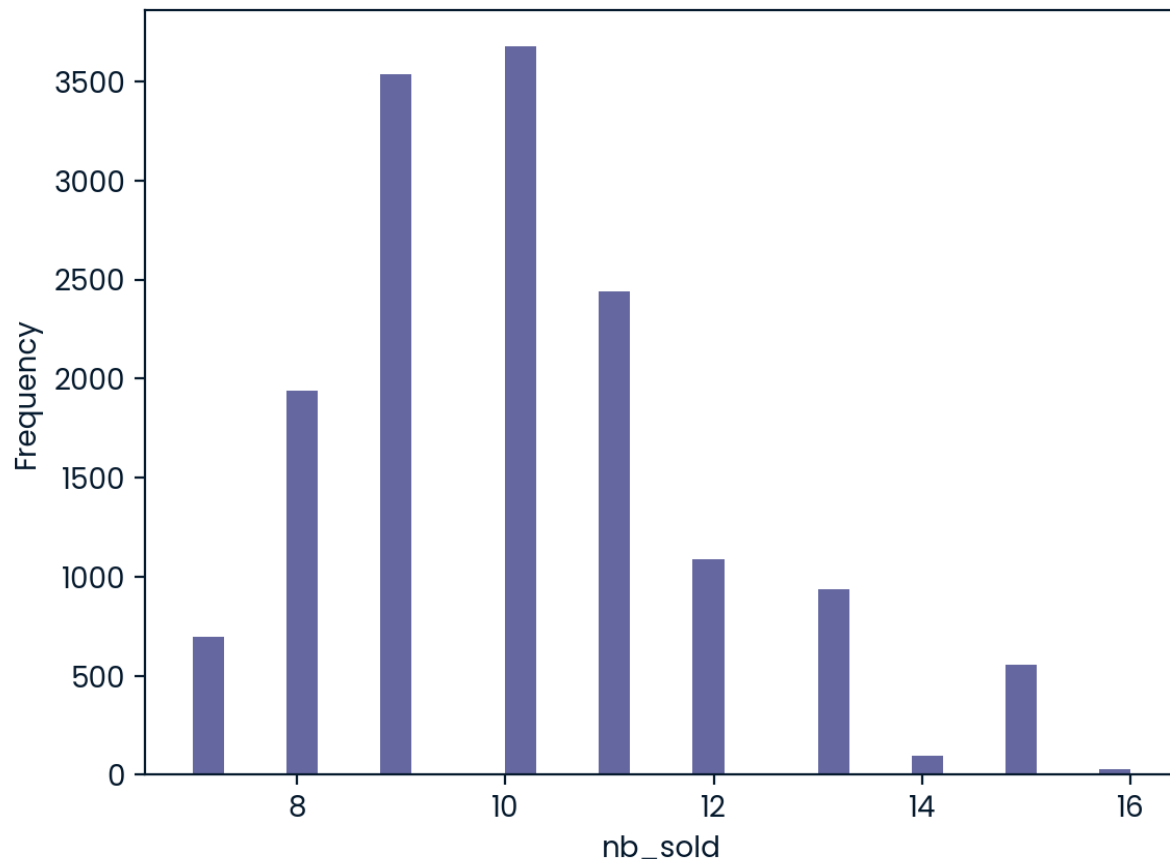
Chi2=251.12, p-value=0.000
→ Significant association

In the **Handling Missing Values** sub-section, I first calculated that **7.16%** of all revenue entries were null. To keep track of which rows were originally missing, I created a new column, `revenue_category`, that flags each record as "Null" or "Not_Null." Then, rather than a one-size-fits-all fill, I performed **stratified median imputation**—grouping by `sales_method` and replacing each group's missing revenue values with that group's median. Finally, I verified that no nulls remain, ensuring the revenue field is now complete and ready for analysis.

PART 2 : EXPLORE ANALYSIS

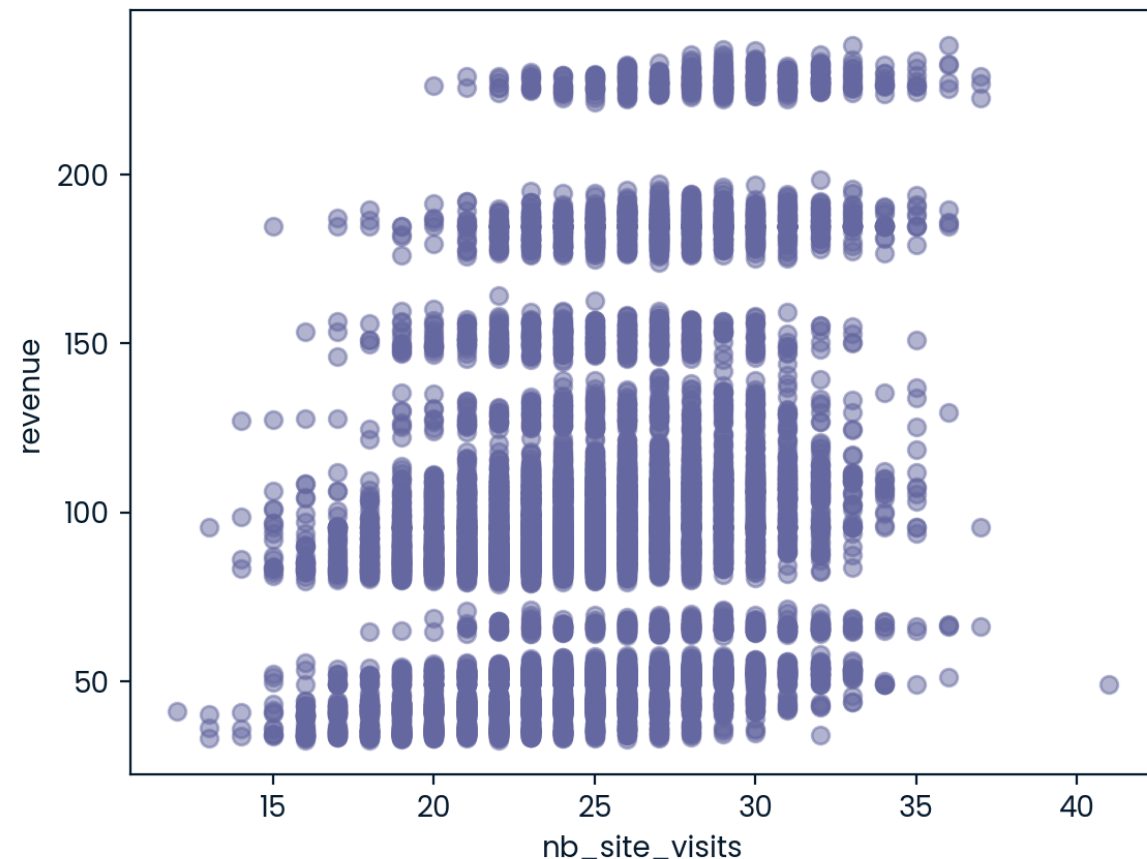
Now, our data is ready for exploratory data analysis :

Units Sold Distribution



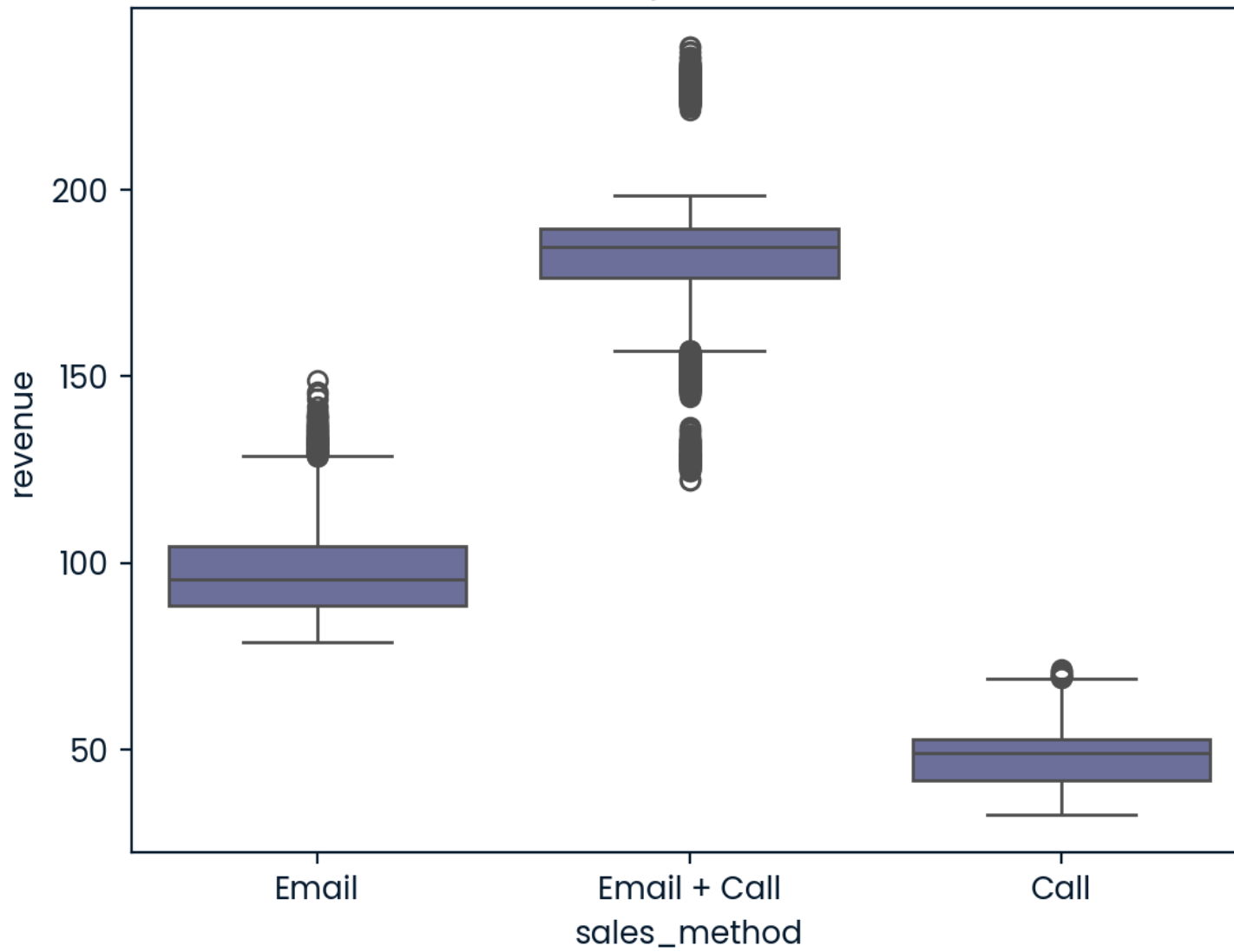
As I explored the **Units Sold Distribution**, I figured out that most transactions cluster between **8 and 11 units**, with a clear peak at **10 units sold**. The right-skewed tail (up to 16 units) shows that very large orders are rare, confirming that the typical order size is around ten items.

Site Visits vs Revenue



When I plotted **Site Visits vs. Revenue**, I saw a generally positive trend—more visits tend to bring in higher revenue—but with a wide scatter. This tells me that while driving traffic helps, conversion efficiency varies greatly from session to session.

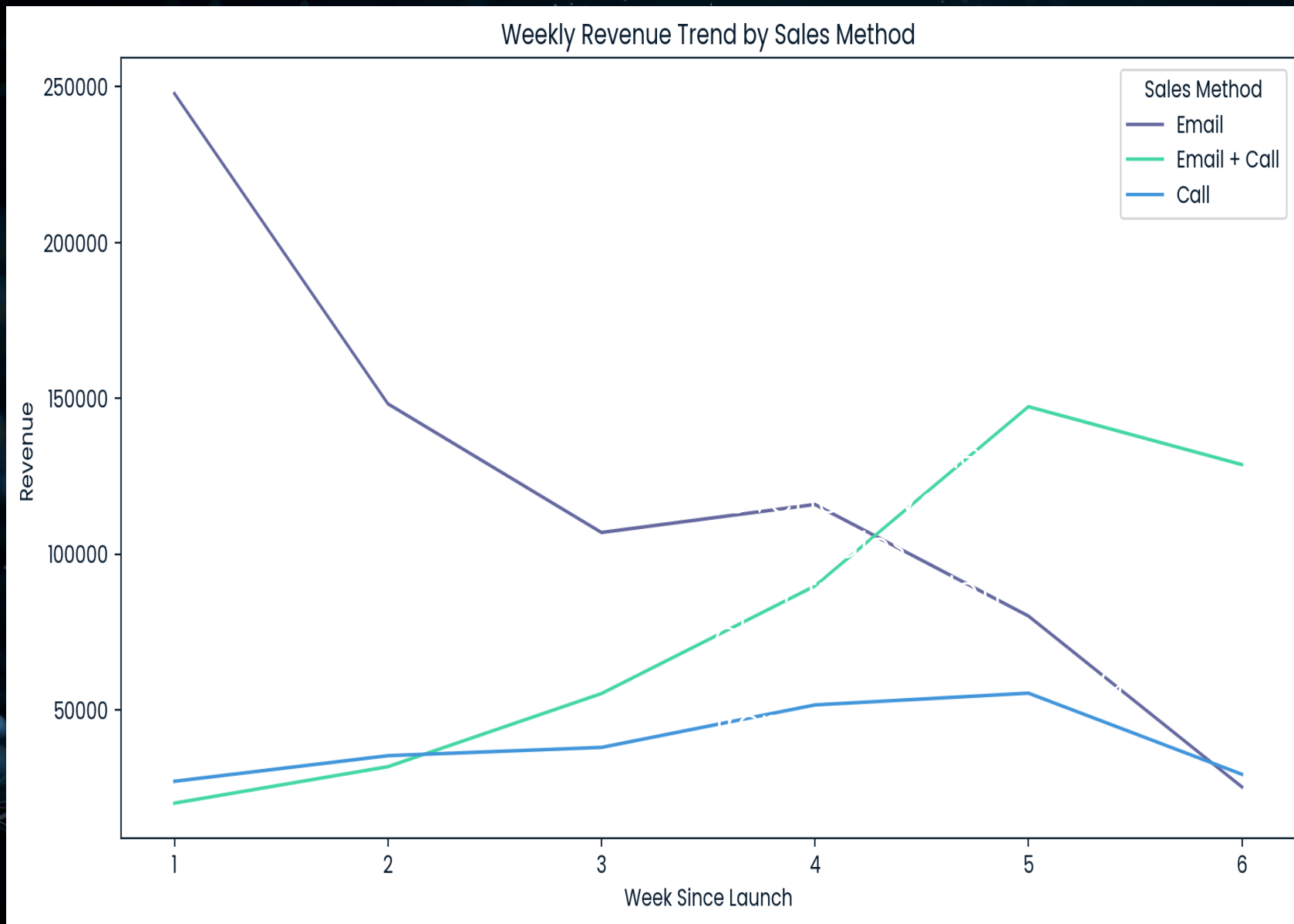
Revenue by Sales Method



OBSERVATIONS:

I found that **Email-only** outreach generates more revenue than **Call-only**, suggesting customers prefer a written channel they can reference later over a one-off phone call. When I combined **Email + Call**, performance improved even further—this dual approach consistently outperformed each method on its own.

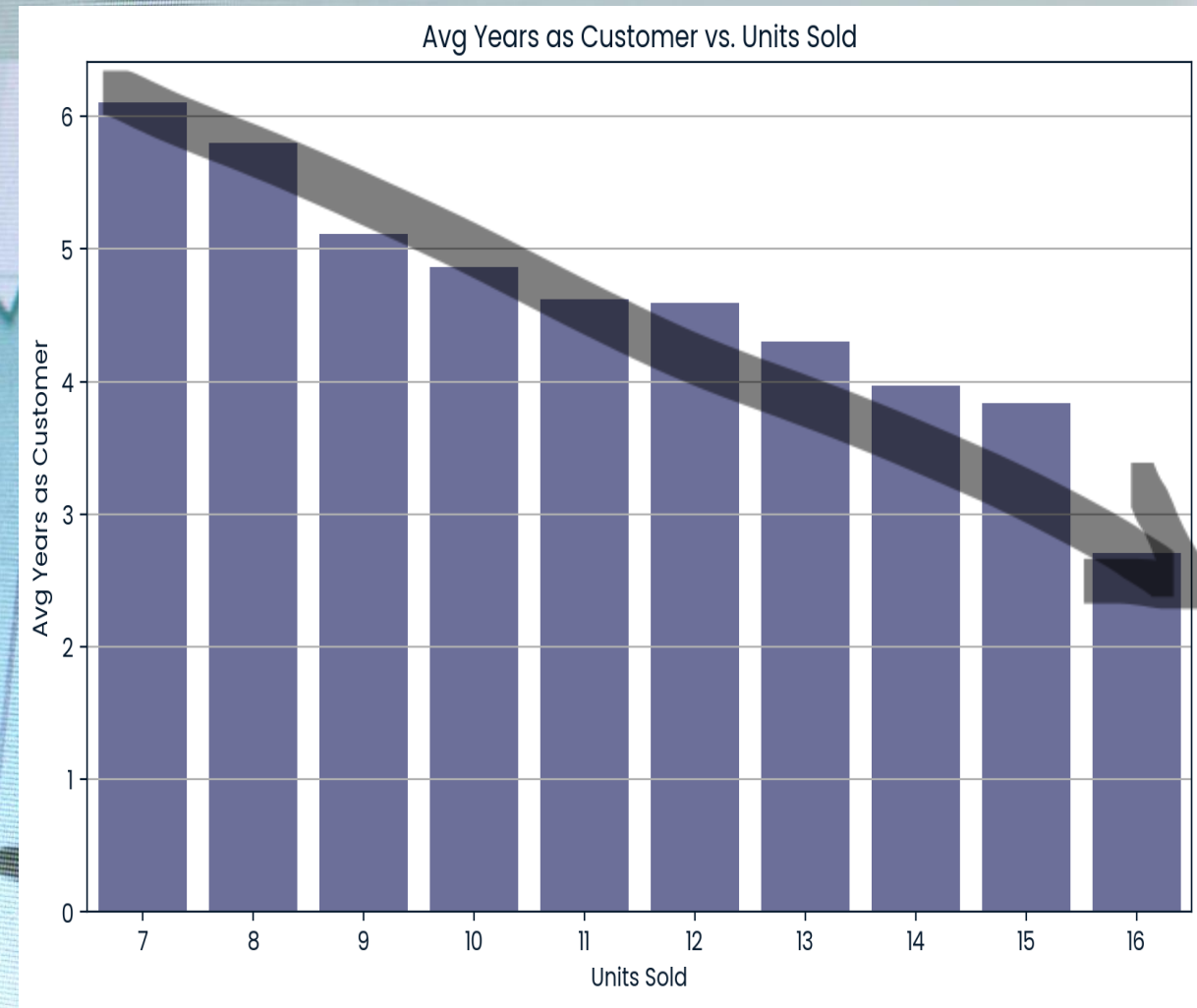
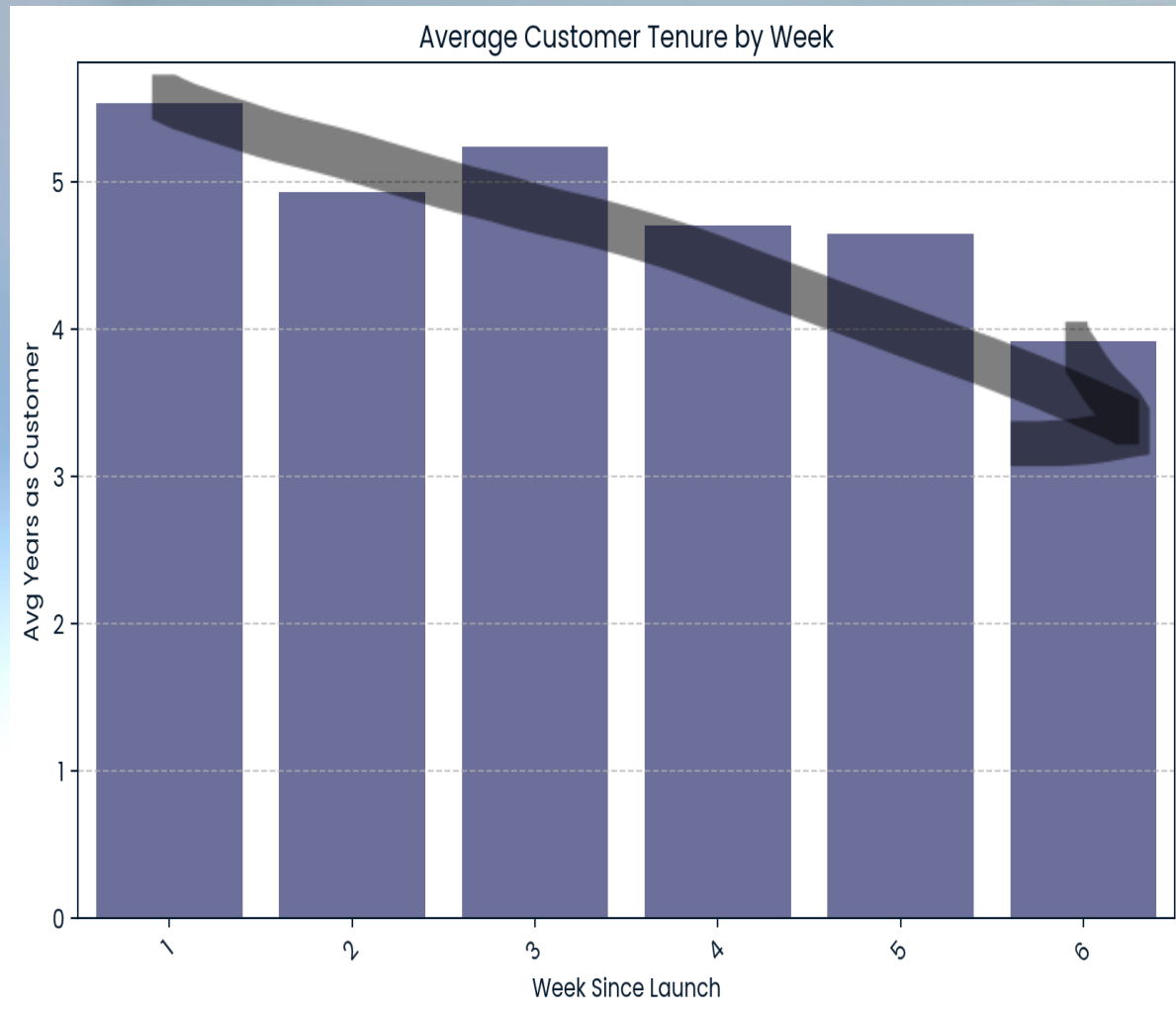
This makes sense because **Email** delivers detailed information and links that customers can revisit, while **Calls** add a personal touch and allow real-time Q&A. By pairing depth (email) with immediacy (call), I created a more compelling, well-rounded engagement strategy that maximizes conversion.



To see how median revenue evolves for each sales channel over time, I plotted line charts of weekly revenue trends starting from the product launch.

By examining the **Weekly Revenue Trend**, I observed that **Email** sales front-loaded the launch—starting at \$250K then steadily declining—while **Email + Call** ramped up to a peak (~\$150K) in week 5.

Call grew modestly before tapering off. This suggests different channels peak at different times during the product rollout

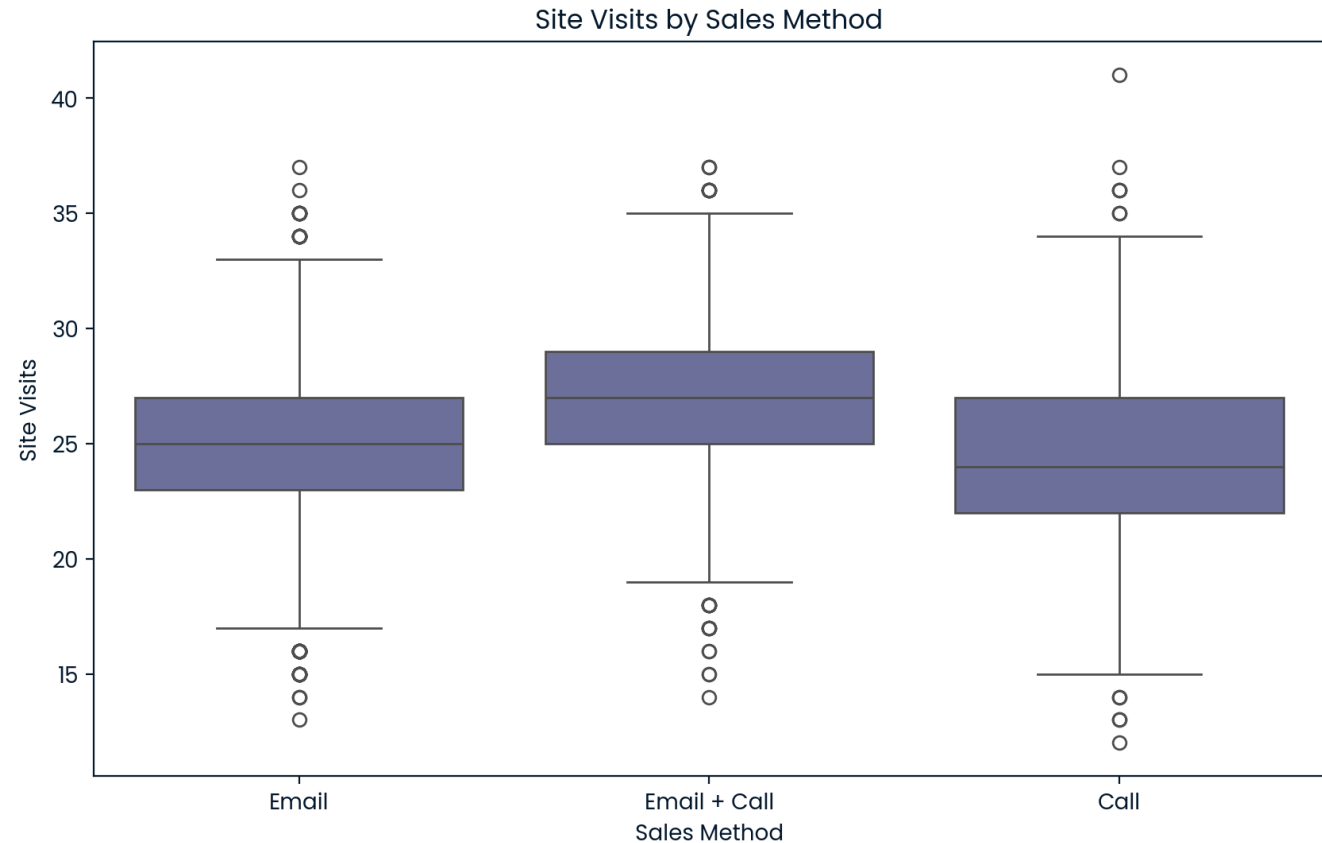


As I explored customer tenure over time, I noticed that the earliest weeks after launch were dominated by our longest-standing customers—those with the highest “years_as_customer.” When I then compared tenure to units sold, those same veteran customers purchased larger quantities on average than newer customers. This pattern tells me that our most loyal clients not only adopt new products first but also buy in greater volume—clear evidence that our long-term relationships are healthy and built on trust.

PART 3: METRICS


Average Revenue per Customer Visit, calculated as total revenue divided by total site visits, tells you exactly how much each interaction with your website is worth. By tracking this metric over time, you can spot trends—rising values suggest that recent UX improvements or marketing efforts are paying off, while any drop may highlight areas where visitors are slipping through the funnel. Comparing your average against industry benchmarks or past performance helps gauge competitiveness and set realistic goals.

You can also use it to evaluate specific campaigns—if a new email promotion or landing-page redesign drives the metric upward, you know it's resonating with higher-value visitors. Segmenting by channel, customer cohort, or geography further reveals which audiences deliver the best return on each visit. Based on our current data, we start with an average of about \$3.80 per visit, giving us a clear baseline for ongoing optimization.



4.CONCLUSION

In conclusion I found that **Email + Call** delivers the highest revenue per visit, but I'll keep tracking **all** channels over a longer period before tweaking our strategy. I also plan to monitor other key metrics—**conversion rate**, **CAC**, **CLV**, **churn rate**, **ROI**, and **AOV**—so I get a complete picture of performance. With more granular data in future analyses, I can calculate these KPIs precisely and make smarter, data-driven decisions.



THANK YOU

