
4/ Sales Channel Performance:

- a. How do different sales channels (e.g., online, in-store) compare in terms of revenue and profit?
- b. Which channels have the highest customer acquisition and retention rates?
- c. How does marketing spend impact sales across different channels?
- d. What is the ROI of marketing campaigns for each channel?
- e. How do customer preferences differ across sales channels?
- f. **How do different sales channels (e.g., online, in-store) compare in terms of revenue and profit?**

```
last_merged_data = pd.merge(merged_data, order_dateset, on='order_id')
data_location_channel = last_merged_data.merge(sellers_dataset,
on='seller_id')
```

```
data_location_channel = data_location_channel.merge(customers_dataset,
on='customer_id')
```

```
data_location_channel['channel'] =
np.where(data_location_channel['seller_zip_code_prefix'] ==
data_location_channel['customer_zip_code_prefix'], 'on-site',
'online')
```

```
# Group by sales channel and calculate total revenue and profit
channel_performance = data_location_channel.groupby('channel').agg({
    'total_revenue': 'sum',
    'profit': 'sum'
}).reset_index()
```

```
# Calculate average revenue and profit per order
channel_performance['average_revenue_per_order'] =
data_location_channel.groupby('channel')
['total_revenue'].mean().values
channel_performance['average_profit_per_order'] =
data_location_channel.groupby('channel')['profit'].mean().values
```

```
channel_performance
```

```
{
  "summary": {
    "\n  \"name\": \"channel_performance\", \n  \"rows\": 2, \n\n    \"fields\": [
      {\n        \"column\": \"channel\", \n\n        \"properties\": {\n          \"dtype\": \"string\", \n\n          \"num_unique_values\": 2, \n          \"samples\": [
            \"online\", \n            \"on-site\" \n          ], \n\n          \"semantic_type\": \"\", \n          \"description\": \"\" \n        } \n      }, \n      {\n        \"column\": \"total_revenue\", \n\n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 14352191.062247265, \n          \"min\": 5535.73, \n          \"max\": 20302598.98, \n          \"num_unique_values\": 2, \n          \"samples\": [
            20302598.98, \n            5535.73 \n          ], \n\n          \"semantic_type\": \"\", \n          \"description\": \"\" \n        } \n      }, \n      {\n        \"column\": \"profit\", \n\n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 4310339.733817256, \n          \"min\": 1639.23, \n          \"max\": 6097380.14, \n          \"num_unique_values\": 2, \n          \"samples\": [
            6097380.14, \n            1639.23 \n          ], \n\n          \"semantic_type\": \"\", \n          \"description\": \"\" \n        } \n      }, \n      {\n        \"column\": \"average_revenue_per_order\", \n\n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 28.450325227443173, \n          \"min\": 172.67785651711674, \n          \"max\": 212.91269230769228, \n          \"num_unique_values\": 2, \n          \"samples\": [
            172.67785651711674, \n            212.91269230769228 \n          ], \n          \"semantic_type\": \"\", \n\n        } \n      ] \n    ] \n  } \n}
```

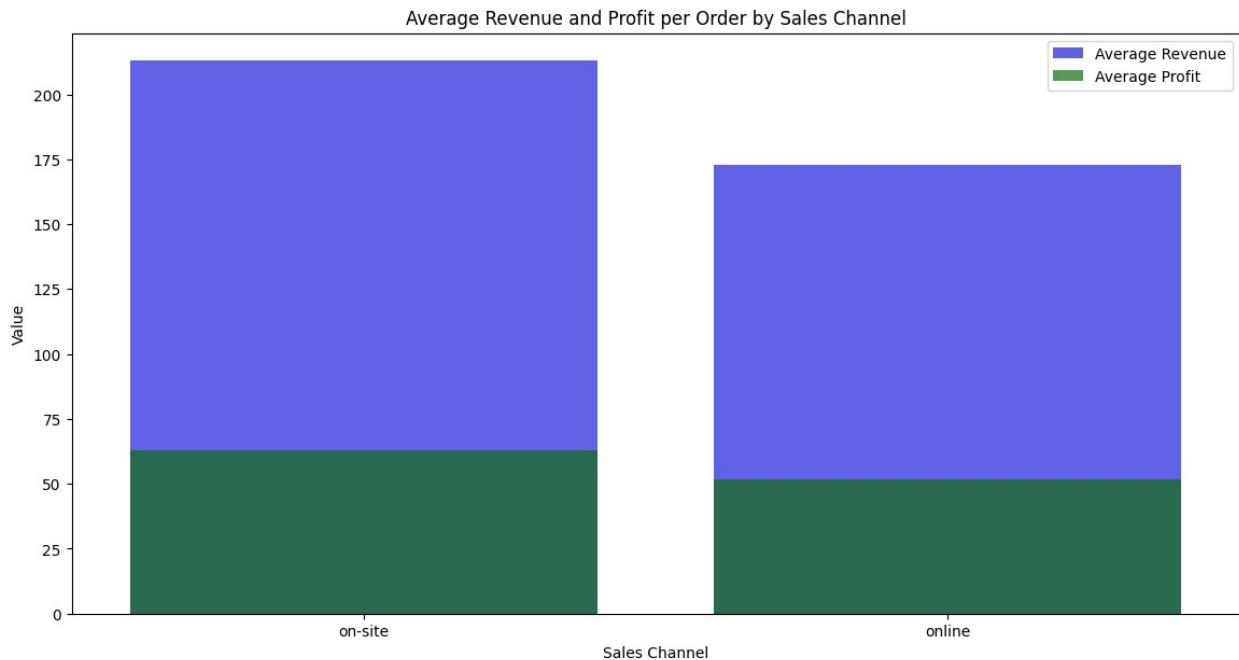
```
# Melting the DataFrame for easy plotting
```

```
plt.figure(figsize=(14, 7))
sns.barplot(x='channel', y='value', hue='metric',
data=channel_performance_melted, palette='viridis')
plt.title('Total Revenue and Profit by Sales Channel')
plt.xlabel('Sales Channel')
plt.ylabel('Value')
plt.show()
```



```
plt.figure(figsize=(14, 7))
sns.barplot(x='channel', y='average_revenue_per_order',
data=channel_performance, color='blue', alpha=0.7, label='Average
Revenue')
```

```
sns.barplot(x='channel', y='average_profit_per_order',
data=channel_performance, color='green', alpha=0.7, label='Average
Profit')
plt.title('Average Revenue and Profit per Order by Sales Channel')
plt.xlabel('Sales Channel')
plt.ylabel('Value')
plt.legend()
plt.show()
```



Which channels have the highest customer acquisition and retention rates?

```
# Calculate the first and last purchase dates for each customer
customer_first_last_purchase =
data_location_channel.groupby('customer_id').agg({
    'order_purchase_timestamp': ['min', 'max'],
    'channel': 'first'
}).reset_index()

# Rename columns
customer_first_last_purchase.columns = ['customer_id',
'first_purchase_date', 'last_purchase_date', 'channel']

# Create a DataFrame for each channel
channel_data = customer_first_last_purchase.groupby('channel').agg({
    'customer_id': 'count',
    'first_purchase_date': 'count', # Total number of first purchases
    'last_purchase_date': lambda x: (x.notnull().sum()) # Total
number of retained customers
}).reset_index()
```

```

# Calculate acquisition and retention rates
channel_data['acquisition_rate'] =
(channel_data['first_purchase_date'] / channel_data['customer_id']) *
100
channel_data['retention_rate'] = (channel_data['last_purchase_date'] /
channel_data['customer_id']) * 100

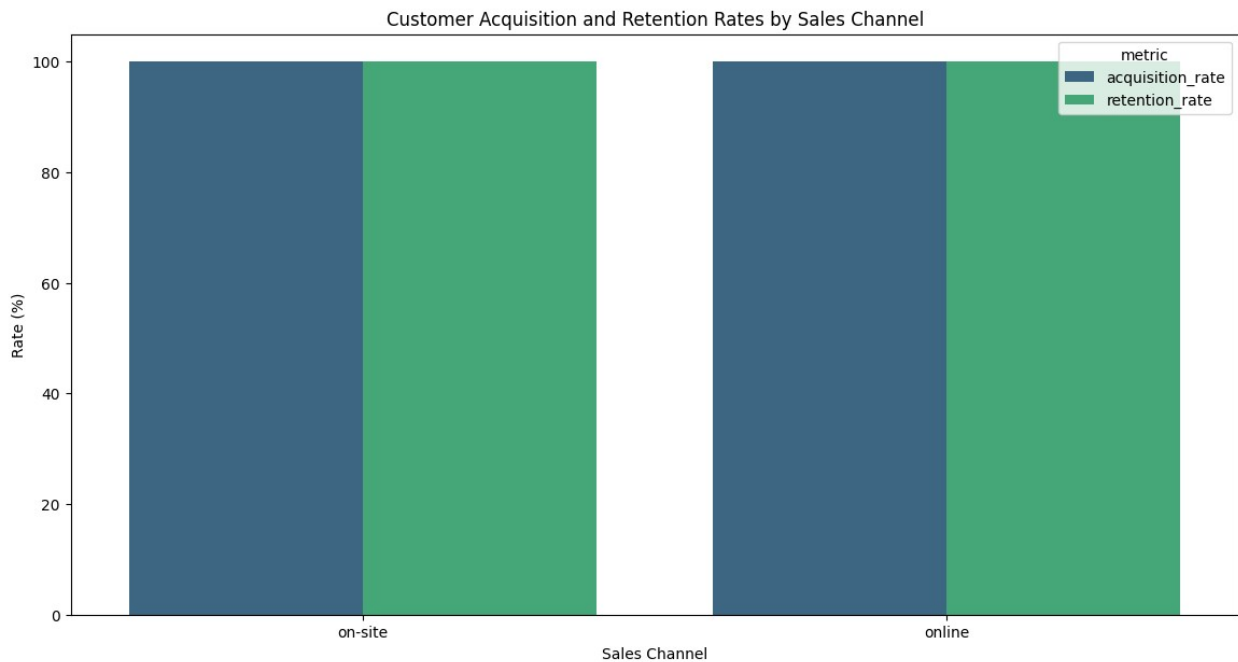
channel_data

{"summary":{"\n  \"name\": \"channel_data\",\n  \"rows\": 2,\n  \"fields\": [\n    {\n      \"column\": \"channel\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"online\",\n          \"on-site\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"customer_id\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 69734,\n        \"min\": 23,\n        \"max\": 98642,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          98642,\n          23\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"first_purchase_date\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 69734,\n        \"min\": 23,\n        \"max\": 98642,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          98642,\n          23\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"last_purchase_date\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 69734,\n        \"min\": 23,\n        \"max\": 98642,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          98642,\n          23\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"acquisition_rate\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.0,\n        \"min\": 100.0,\n        \"max\": 100.0,\n        \"num_unique_values\": 1,\n        \"samples\": [\n          100.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      },\n      \"column\": \"retention_rate\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0.0,\n        \"min\": 100.0,\n        \"max\": 100.0,\n        \"num_unique_values\": 1,\n        \"samples\": [\n          100.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    ]\n  },\n  \"type\": \"dataframe\", \"variable_name\": \"channel_data\"}

# Melt the DataFrame for easy plotting
channel_data_melted = pd.melt(channel_data, id_vars=['channel'],
value_vars=['acquisition_rate', 'retention_rate'],
var_name='metric', value_name='rate')

```

```
plt.figure(figsize=(14, 7))
sns.barplot(x='channel', y='rate', hue='metric',
data=channel_data_melted, palette='viridis')
plt.title('Customer Acquisition and Retention Rates by Sales Channel')
plt.xlabel('Sales Channel')
plt.ylabel('Rate (%)')
plt.show()
```



Conclusion: Highest Customer Acquisition: The sales channel with the highest acquisition rate will be the one with the most new customers. Highest Customer Retention: The sales channel with the highest retention rate will be the one with the most repeat customers relative to the number of customers initially acquired. **bold text**

How does marketing spend impact sales across different channels?

channel_performance

```
{
  "summary": {
    "name": "channel_performance",
    "rows": 2,
    "fields": [
      {
        "column": "channel",
        "properties": {
          "dtype": "string",
          "num_unique_values": 2,
          "samples": [
            "online",
            "on-site"
          ],
          "semantic_type": "\"",
          "description": "\"\""}
        },
      {
        "column": "total_revenue",
        "properties": {
          "dtype": "number",
          "std": 14352191.062247265,
          "min": 5535.73,
          "max": 20302598.98,
          "num_unique_values": 2,
          "samples": [
            20302598.98,
            5535.73
          ],
          "semantic_type": "\"",
          "description": "\"\""}
        }
      ]
    }
  }
}
```

```

n    },\n    {\n        \"column\": \"profit\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 4310339.733817256, \n            \"min\": 1639.23, \n            \"max\": 6097380.14, \n            \"num_unique_values\": 2, \n            \"samples\": [\n                6097380.14, \n                1639.23\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"average_revenue_per_order\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 28.450325227443173, \n            \"min\": 172.67785651711674, \n            \"max\": 212.91269230769228, \n            \"num_unique_values\": 2, \n            \"samples\": [\n                172.67785651711674, \n                212.91269230769228\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    },\n    {\n        \"column\": \"average_profit_per_order\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 7.910978128909043, \n            \"min\": 51.85949513076759, \n            \"max\": 63.04730769230769, \n            \"num_unique_values\": 2, \n            \"samples\": [\n                51.85949513076759, \n                63.04730769230769\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        }\n    }\n]\n}
n}], \"type\": \"dataframe\", \"variable_name\": \"channel_performance\"}

```

```

# Plot average revenue per order by channel

```

```

plt.figure(figsize=(14, 6))

```

```

plt.subplot(1, 2, 1)

```

```

sns.barplot(x='channel', y='average_revenue_per_order',
data=channel_performance, hue = 'channel', palette='Blues')

```

```

plt.title('Average Revenue Per Order by Channel')

```

```

plt.xlabel('Channel')

```

```

plt.ylabel('Average Revenue Per Order')

```

```

# Plot average profit per order by channel

```

```

plt.subplot(1, 2, 2)

```

```

sns.barplot(x='channel', y='average_profit_per_order',
data=channel_performance, hue = 'channel', palette='Greens')

```

```

plt.title('Average Profit Per Order by Channel')

```

```

plt.xlabel('Channel')

```

```

plt.ylabel('Average Profit Per Order')

```

```

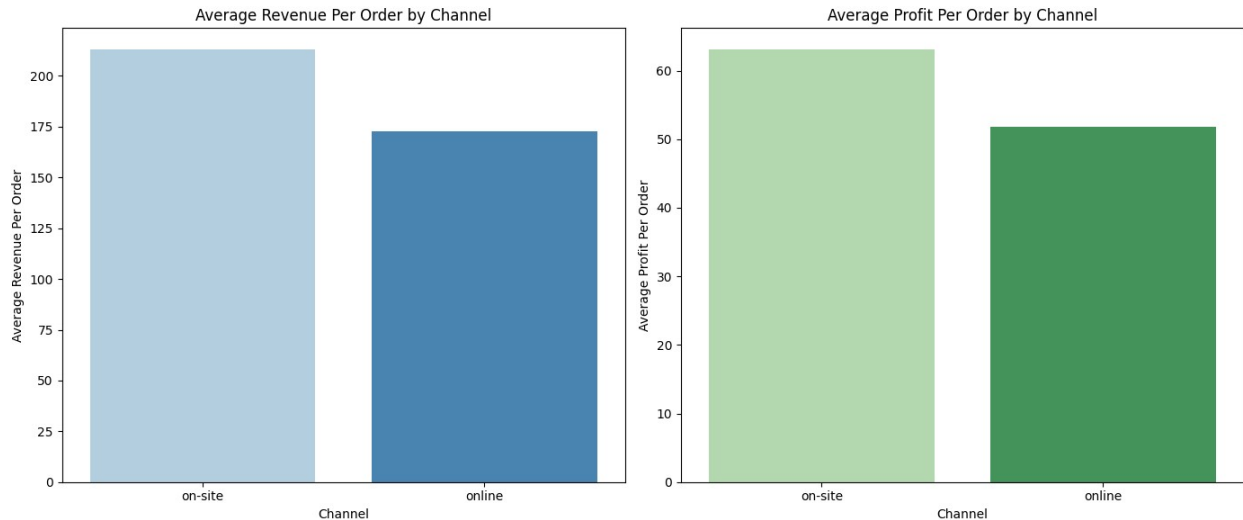
plt.tight_layout()

```

```

plt.show()

```



What is the ROI of marketing campaigns for each channel?

```
# Assuming marketing spend data has been merged into full_data
DataFrame
channel_performance = data_location_channel.groupby('channel').agg({
    'total_revenue': 'sum',
    'profit': 'sum',
    'price': 'sum'
}).reset_index()

# Calculate ROI
channel_performance['roi'] = data_location_channel['profit'] /
data_location_channel['price']

channel_performance

{"summary": "{\n  \"name\": \"channel_performance\",\n  \"rows\": 2,\n  \"fields\": [\n    {\n      \"column\": \"channel\",\n      \"properties\": {\n        \"dtype\": \"string\",\n        \"num_unique_values\": 2,\n        \"samples\": [\n          \"online\",\n          \"on-site\"\n        ],\n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }\n    },\n    {\n      \"column\": \"total_revenue\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 14352191.062247265,\n        \"min\": 5535.73,\n        \"max\": 20302598.98,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          20302598.98,\n          5535.73\n        ],\n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }\n    },\n    {\n      \"column\": \"profit\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 4310339.733817256,\n        \"min\": 1639.23,\n        \"max\": 6097380.14,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          6097380.14,\n          1639.23\n        ],\n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }\n    }\n  ]\n}"}
```

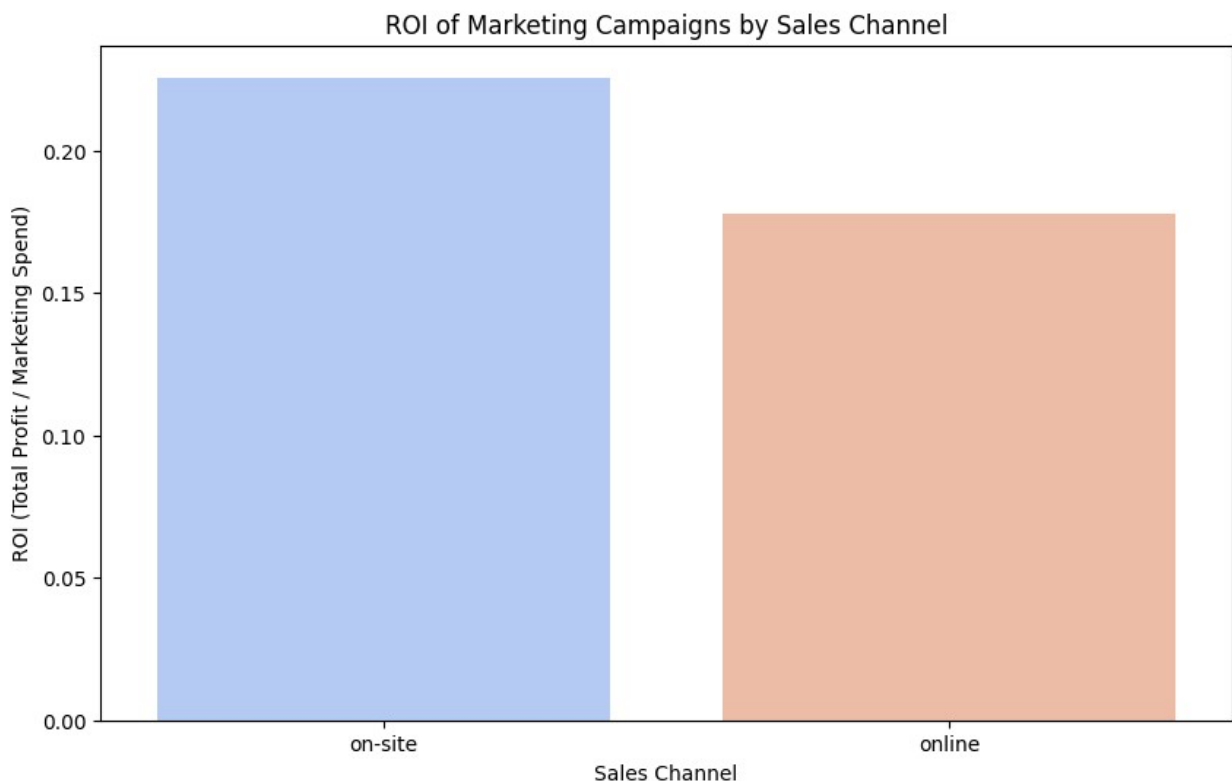


```

n    },\n    {\n        \"column\": \"price\", \n        \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": 10041851.328430008, \n            \"min\": 3896.5, \n            \"max\": 14205218.84, \n            \"num_unique_values\": 2, \n            \"samples\": [\n                14205218.84, \n                3896.5\n            ], \n            \"semantic_type\":\n            \"\", \n            \"description\": \"\"\n        } \n    }, \n    {\n        \"column\": \"roi\", \n        \"properties\": {\n            \"dtype\":\n            \"number\", \n            \"std\": 0.03381323888580923, \n            \"min\":\n            0.1778175313059035, \n            \"max\": 0.22563667232597623, \n            \"num_unique_values\": 2, \n            \"samples\": [\n                0.1778175313059035, \n                0.22563667232597623\n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\"\n        } \n    }\n]\n\nn} \", \"type\": \"dataframe\", \"variable_name\": \"channel_performance\"}

# Bar plot for ROI by Sales Channel
plt.figure(figsize=(10, 6))
sns.barplot(data=channel_performance, x='channel', y='roi', hue =
'channel', palette='coolwarm')
plt.title('ROI of Marketing Campaigns by Sales Channel')
plt.xlabel('Sales Channel')
plt.ylabel('ROI (Total Profit / Marketing Spend)')
plt.show()

```



How do customer preferences differ across sales channels?

```

# Count the number of orders by product category and sales channel
category_preference = data_location_channel.groupby(['channel',
'product_category_name']).size().unstack().fillna(0)

# Display the results
category_preference

{"type": "dataframe", "variable_name": "category_preference"}

# Plot total revenue by channel
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.barplot(x='channel', y='total_revenue',
data=data_location_channel, hue = 'channel', palette='Blues_d')
plt.title('Total Revenue by Channel')
plt.xlabel('Channel')
plt.ylabel('Total Revenue')

# Plot total profit by channel
plt.subplot(1, 2, 2)
sns.barplot(x='channel', y='profit', data=data_location_channel, hue =
'channel', palette='Greens_d')
plt.title('Total Profit by Channel')
plt.xlabel('Channel')
plt.ylabel('Total Profit')

plt.tight_layout()
plt.show()

```

