Group Assignment: Data Preparation and Visualization

1. Group members

Skylar Liu

---

2. Contribution report

After completing the assignment, please answer the following questions **as a group** by adding your answers within this markdown block.

1. In a few sentences, describe each group member's individual contributions to the submission. Be as specific as possible (e.g. coordinated group efforts, specific problems answered, specific problems reviewed/revised, sections of the submission written, etc.).

I completed this entire project on my own.

2. In a few sentences, describe what was learned in completing this assignment. In particular, describe what was learned through the **specific individual contributions** mentioned above.

I learned how to write more in depth SQL code through trial and error. I also learned how to visualize the relevant data, and the nuances of making the visualizations presentation ready. It was nice to get some experience with Python and utilizing SQL in real-world scenarios.

---

3. Assignment

Instructions:

You will be presented with a scenario and will need to utilize your SQL and python skills to complete this assignment successfully.

Put this .ipynb file in the `jupyter_notebooks` folder in your Docker SQLPython Container directory. Then you will be able to connect to the database and run your code without issue.

Each group will submit two files:

1. a single Jupyter Notebook (.ipynb). **You must run all cells before submitting.** This notebook should have all of the relevant visualizations and output displayed properly. We will restart and run all of the code from this notebook, which should not produce any errors.
2. a PDF version (.pdf) of the Jupyter Notebook. This PDF should have all of the relevant visualizations and output displayed properly.

In [80]:
```
#run this code first to connect to the database and verify the connection is working
## DO NOT MODIFY THIS CODE BLOCK
```

```
## If you have placed this notebook in the jupyter notebooks folder properly,
## this block should return the first two rows of the customers table

from sqlalchemy import create_engine
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

%matplotlib inline

cnxn_string = ("postgresql+psycopg2://{username}:{pswd}"
               "@{host}:{port}/{database}")
print(cnxn_string)

engine = create_engine(cnxn_string.format(
    username="postgres",
    pswd="behappy",
    host="postgres",
    port=5432,
    database="sqlda"))

engine.execute("SELECT * FROM customers LIMIT 2;").fetchall()
```

```
postgresql+psycopg2://{username}:{pswd}@{host}:{port}/{database}
```

Out[80]:

```
[(2, 'Dr', 'Ode', 'Stovin', None, 'ostovin1@npr.org', 'M', '16.97.59.186', '314-534-4361', '2573
Fordem Parkway', 'Saint Louis', 'MO', '63116', 38.5814, -90.2625, datetime.datetime(2014, 10, 2,
0, 0)),
 (5, None, 'Lonnie', 'Rembaud', None, 'lrembaud4@discovery.com', 'F', '18.131.58.65', '786-499-3
431', '38 Lindbergh Way', 'Miami', 'FL', '33124', 25.5584, -80.4582, datetime.datetime(2014, 3,
6, 0, 0))]
```

Scenario

You are a team of extremely successful data scientists at a top motor dealership company. You need to create summary tables and visualizations that your boss will present at the next company shareholder meeting. She has sent you the following e-mail describing what she needs.

From: importantboss@topmotordealershipcompany.com

To: datascienceteam@topmotordealershipcompany.com

Subject: Data request for shareholder meeting

For our next shareholder meeting, we need to provide more information about sales performance across states, across dealerships, and across sales channels. Please send me information to address the following items for our next shareholder meeting along with your thoughts.

1. Sales performance at the state level (top 5 and bottom 5 states)
2. For the best performing states, which dealerships are performing well and how are they trending?
3. In states with dealerships, does the distribution of sales amounts vary across channels (internet vs. dealership)?

Thank you!

-Important Boss

Your team promptly comes up with the following plan.

Part 1: Visualizing the top and bottom performing states

1. Write a SELECT query that returns the total sales amount for each state from January 1, 2017 to now. The table should have two columns, `state` and `total_sales_amount`, with one row for each state ordered by `total_sales_amount` in *descending* order. Make sure that `total_sales_amount` is rounded appropriately. Attribute sales to states based on the **state in which the customer that made the purchase resides**. This way we can capture both sales made through dealerships, as well as sales made through our website, in evaluating state-level performance.

2. Use SQLAlchemy to execute the query and store the results in a pandas dataframe called `sales_by_state`.

3. Display the rows in `sales_by_state` corresponding to the 5 states with the **largest** total sales amount in *descending* order.

4. Display the rows in `sales_by_state` corresponding to the 5 states with the **smallest** total sales amount in *ascending* order.

5. Visualize sales performance by state for the top and bottom performing states discovered in 1.3 and 1.4. You can use more than one visualization. These should be **presentation ready** (e.g. appropriate and complete titles and axis labels, remove unnecessary/distracting features, display date range for total sales, no overlapping axis labels, etc.).

Include the code needed for each component of part 1 in the appropriate code block below.

In [81]:
```
#1.1
query_1 = """
SELECT state, CAST(SUM(sales_amount) as DECIMAL(18,2)) AS total_sales_amount
FROM (
     (SELECT state, sales_amount, sales_transaction_date
      FROM dealerships AS d
      JOIN sales AS s
        ON d.dealership_id = s.dealership_id
      WHERE channel = 'dealership'
        AND sales_transaction_date >= '2017-01-01')
     UNION ALL
     (SELECT state, sales_amount, sales_transaction_date
      FROM customers AS c
      JOIN sales as s
        ON c.customer_id = s.customer_id
      WHERE channel = 'internet'
        AND sales_transaction_date >= '2017-01-01')) AS x
WHERE state is NOT NULL
GROUP BY state
ORDER BY total_sales_amount DESC
"""
```

In [83]:
```
#1.2 create dataframe
df_1 = pd.read_sql_query(query_1,engine)
```

In [84]:
```
#1.3 display top 5 performing states
top_5 = df_1.iloc[0:5,:]
top_5
```

Out[84]:

	state	total_sales_amount
0	TX	18335083.69
1	CA	17782468.83
2	IL	12483550.91
3	FL	11409465.84
4	VA	10175647.87

In [100…
```
#1.4 display bottom 5 performing states
bottom_5 = df_1.iloc[:-6:-1]
bottom_5
```

Out[100]:

	state	total_sales_amount
50	WY	1699.97
49	ME	2054.97
48	RI	3949.93
47	ND	7369.88
46	SD	9099.84

In [104…
```
#1.5 visualize top and bottom performing states
fig = plt.figure(figsize=(8,4))
#first subplot
ax1 = fig.add_subplot(121)
plt.bar(top_5.state, top_5.total_sales_amount, color=['lightblue','lightgreen','yellow', 'orange
ax1.set_title('Top 5 States')
ax1.set_xlabel('State')
ax1.set_ylabel('Total Sales (US Dollars)')

#second subplot
ax2 = fig.add_subplot(122)
plt.bar(bottom_5.state[:-6:-1], bottom_5.total_sales_amount[:-6:-1],
        color=['lightblue','lightgreen','yellow', 'orange', 'pink'])
ax2.set_title('Bottom 5 States')
ax2.set_xlabel('State')
ax2.set_ylabel('Total Sales (US Dollars)')
fig.suptitle('Total Sales by State from January 1, 2017 to Present', size=14)
plt.tight_layout()
```

Out[104]:
```
<BarContainer object of 5 artists>
```

Out[104]:
```
Text(0.5, 1.0, 'Top 5 States')
```

Out[104]:
```
Text(0.5, 0, 'State')
```

Out[104]:
```
Text(0, 0.5, 'Total Sales (US Dollars)')
```

```
/tmp/ipykernel_11/3446326771.py:12: FutureWarning: The behavior of `series[i:j]` with an integer
-dtype index is deprecated. In a future version, this will be treated as *label-based* indexing,
consistent with e.g. `series[i]` lookups. To retain the old behavior, use `series.iloc[i:j]`. To
get the future behavior, use `series.loc[i:j]`.
  plt.bar(bottom_5.state[:-6:-1], bottom_5.total_sales_amount[:-6:-1],
```

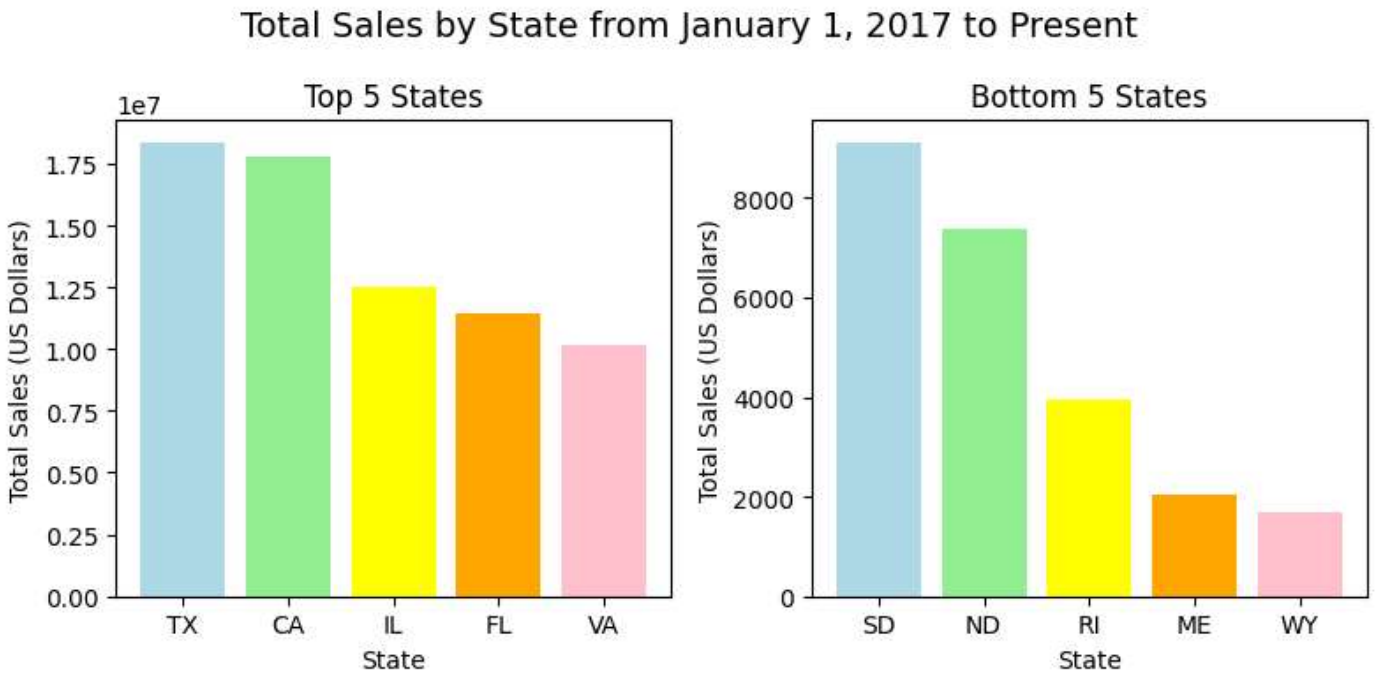
Out[104]: <BarContainer object of 5 artists>

Out[104]: Text(0.5, 1.0, 'Bottom 5 States')

Out[104]: Text(0.5, 0, 'State')

Out[104]: Text(0, 0.5, 'Total Sales (US Dollars)')

Out[104]: Text(0.5, 0.98, 'Total Sales by State from January 1, 2017 to Present')



Part 2: Top performing dealerships

Create a table and visualization of historical cumulative sales amounts by dealership from January 1, 2017 to now. Only include dealerships located in the *top two* states determined in Part 1. It is OK to reference these two states by their abbreviations (e.g. AL, MS, WY) in the query you will develop below since this is a one-off request.

To do this, perform the following steps:

1. Write a SELECT query that returns three columns: `dealership_id`, `sales_transaction_date`, and `cumulative_sales`. `cumulative_sales` represents the cumulative sales amount from January 1, 2017 to the `sales_transaction_date` for dealership identified by `dealership_id`. There should be a row for each distinct combination of `dealership_id` and `sales_transaction_date` in the `sales` table (*hint*: window function).

2. Use SQLAlchemy to execute the query and store the results in a pandas dataframe called `cumulative_sales_bydealership`.

3. Appropriately visualize historical cumulative sales by dealership across sales transaction dates *in a single plot* (*hint*: seaborn). Visualization should be **presentation ready** (e.g. appropriate and complete titles and legend/axis labels, remove unnecessary/distracting features, display date range for total sales, no overlapping axis labels, integer-valued dealership IDs, states indicated clearly, etc.).

In [136…

```
#2.1
query_2 = """
```

```
WITH c AS (
SELECT s.dealership_id::INT, sales_transaction_date::date,
    SUM(sales_amount) OVER (PARTITION BY s.dealership_id ORDER BY sales_transaction_date::date) A
    row_number(*) OVER (PARTITION BY s.dealership_id, sales_transaction_date::date) AS duplicate
FROM dealerships AS d
JOIN sales AS s
    ON d.dealership_id = s.dealership_id
WHERE state IN ('CA', 'TX')
AND sales_transaction_date >= '2017-01-01'
)
SELECT dealership_id, sales_transaction_date, CAST(cumulative_sales as DECIMAL(18,2))
FROM c
WHERE duplicate = 1
"""
```

In [143…

```
#2.2
cumulative_sales_bydealership = pd.read_sql_query(query_2,engine)
```

In [144…

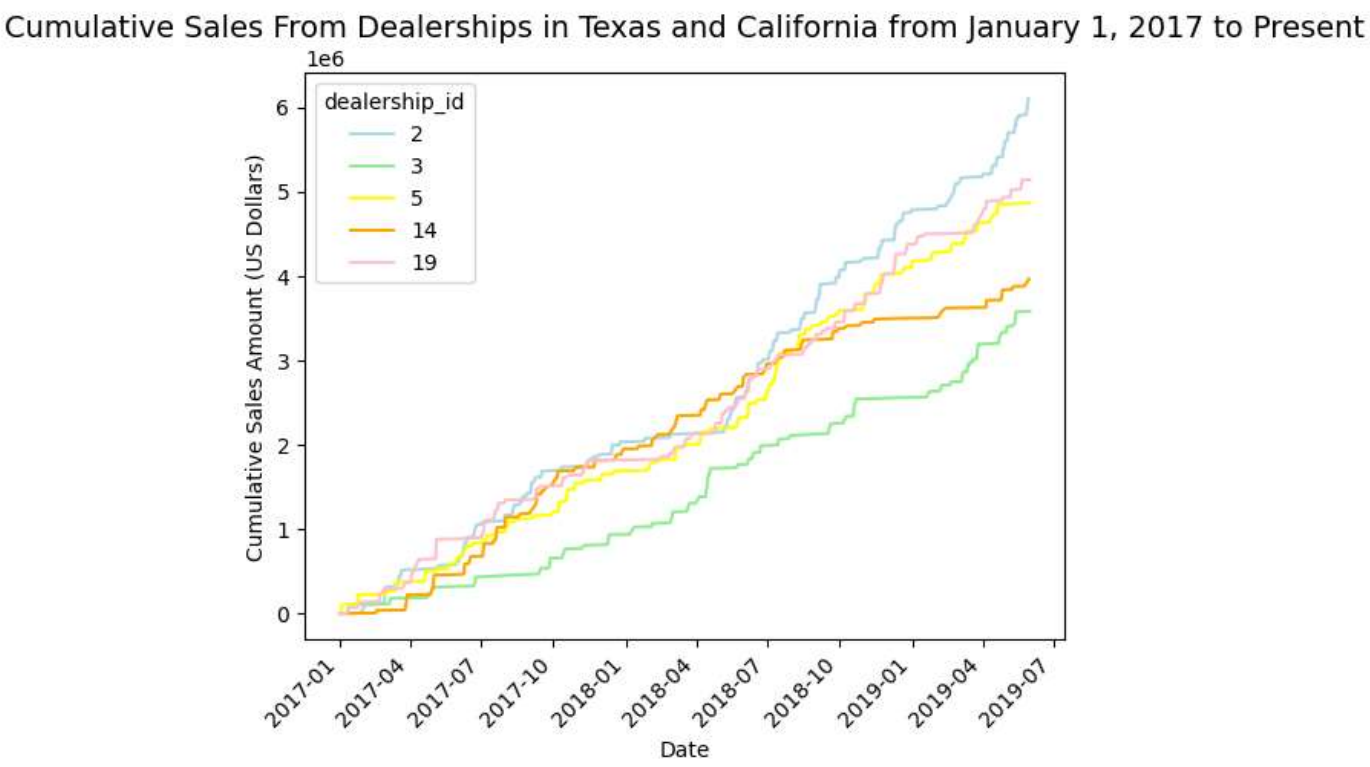
```
#2.3
k = sns.lineplot(cumulative_sales_bydealership, x="sales_transaction_date", y="cumulative_sales"
                 palette=['lightblue','lightgreen','yellow', 'orange', 'pink'])
k.set_title('Cumulative Sales From Dealerships in Texas and California from January 1, 2017 to Pr
plt.xlabel('Date')
plt.ylabel('Cumulative Sales Amount (US Dollars)')
plt.xticks(
    rotation=45,
    horizontalalignment='right',
    fontweight='light'
)
```

Out[144]:
```
Text(0.5, 1.0, 'Cumulative Sales From Dealerships in Texas and California from January 1, 2017 t
o Present')
```

Out[144]:
```
Text(0.5, 0, 'Date')
```

Out[144]:
```
Text(0, 0.5, 'Cumulative Sales Amount (US Dollars)')
```

Out[144]:
```
(array([17167., 17257., 17348., 17440., 17532., 17622., 17713., 17805.,
        17897., 17987., 18078.]),
 [Text(17167.0, 0, '2017-01'),
  Text(17257.0, 0, '2017-04'),
  Text(17348.0, 0, '2017-07'),
  Text(17440.0, 0, '2017-10'),
  Text(17532.0, 0, '2018-01'),
  Text(17622.0, 0, '2018-04'),
  Text(17713.0, 0, '2018-07'),
  Text(17805.0, 0, '2018-10'),
  Text(17897.0, 0, '2019-01'),
  Text(17987.0, 0, '2019-04'),
  Text(18078.0, 0, '2019-07')])
```



Part 3: Sales amount by sales channel

Create tables and visualizations to compare sales amounts by sales channel for sales made on or after January 1, 2015. **Only include sales made to customers that reside in a state that has a dealership.** To do this, perform the following steps:

1. Write a SELECT query that returns sales with a transaction date on or after January 1, 2015 from the `sales` table made to customers that reside in a state that has a dealership. This table should have the following four columns: `channel`, `sales_amount`, and `sales_type` and `sales_year`. `channel` and `sales_amount` are exactly as appears in the `sales` table. `sales_type` is a derived categorical field that takes on a value of `'High value'` when `sales_amount` is above 50000, `'Typical value'` when `sales_amount` is above 10000 but less than or equal to 50000, and `'Low value'` when `sales_amount` is less than 10000. `sales_year` is the year from the `sales_transaction_date` field.

2. Use SQLAlchemy to execute the query and store the results in a pandas dataframe called `sales_from_dealershipstates`.

3. Appropriately visualize the distribution of sales amounts and how it changes by `channel`, `sales_year`, and `sales_type`. To do this, create multiple plots, one for each distinct combination of `sales_year` and `sales_type`. For each plot, visualize and compare the distribution of dealership sales amounts and internet sales amounts. For example, one plot will compare the distribution of dealership sales amounts and internet sales amounts for low value sales in 2015. Arrange the plots so that you can see changes across `sales_year` and `sales_type` (*Hint: seaborn.FacetGrid*). Visualizations should be **presentation ready** (e.g. appropriate and complete titles and legend/axis labels, remove unnecessary/distracting features, display date range for total sales, no overlapping axis labels, etc.).

In [170…

```
#3.1 select query
query_3 = """
SELECT channel, CAST(sales_amount as DECIMAL(18,2)),
    CASE WHEN sales_amount < 10000 THEN 'Low'
    WHEN sales_amount >10000 AND sales_amount <=50000 THEN 'Typical value'
    ELSE 'High' END AS sales_type,
    EXTRACT(YEAR FROM sales_transaction_date)::INT AS sales_year
FROM (
     (SELECT state, sales_amount, sales_transaction_date, channel
      FROM dealerships AS d
      JOIN sales AS s
        ON d.dealership_id = s.dealership_id
      WHERE channel = 'dealership'
        AND sales_transaction_date >= '2015-01-01')
     UNION ALL
     (SELECT state, sales_amount, sales_transaction_date, channel
      FROM customers AS c
      JOIN sales as s
        ON c.customer_id = s.customer_id
      WHERE channel = 'internet'
        AND sales_transaction_date >= '2015-01-01')) AS x
WHERE state is NOT NULL
    AND state IN (SELECT state
                  FROM dealerships)
ORDER BY sales_year, channel, sales_type DESC

"""
```

In [171…

```
#3.2 create data frame
sales_from_dealershipstates = pd.read_sql_query(query_3,engine)
```

In [172…

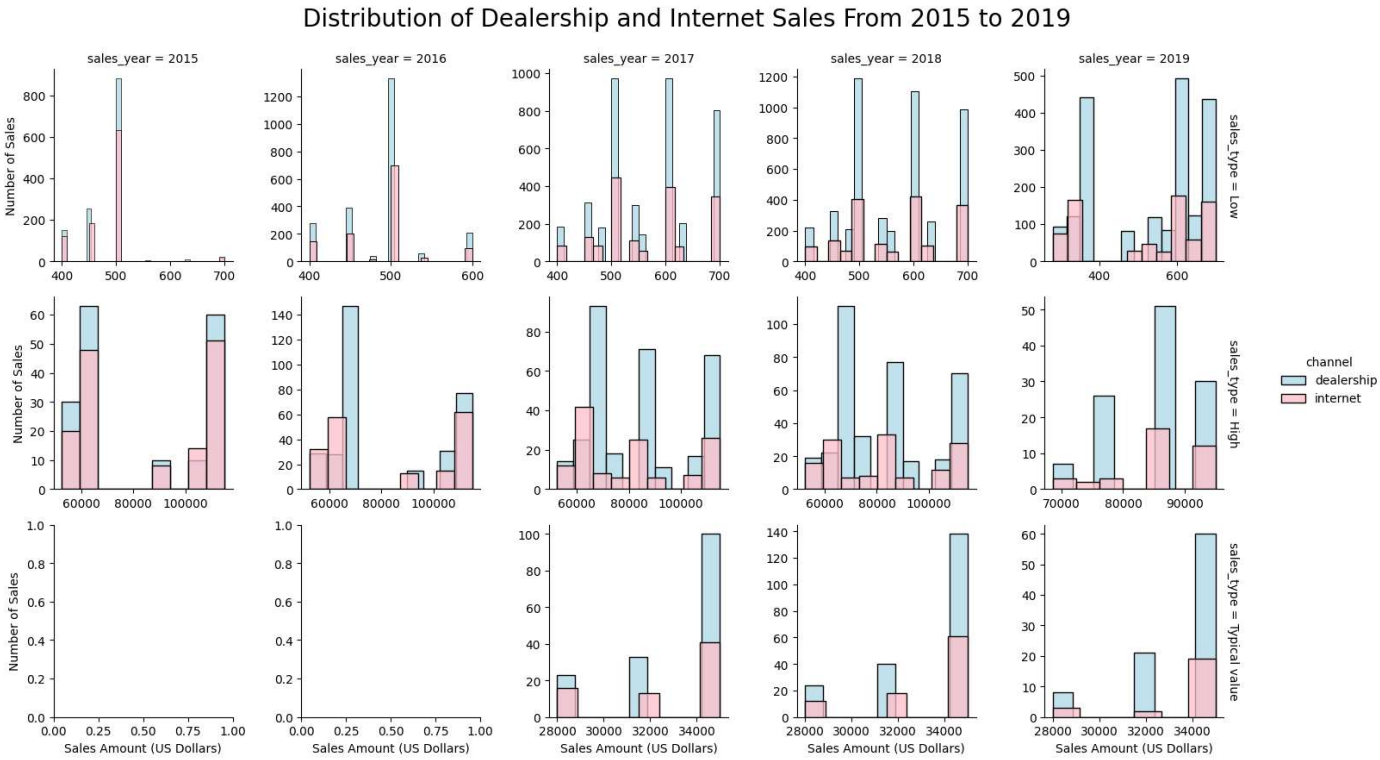
```
#3.3 visualization
g = sns.FacetGrid(sales_from_dealershipstates, row = 'sales_type', col = 'sales_year', hue = 'cha
                  sharex = False, sharey = False, palette=['lightblue', 'pink'])
g.map(sns.histplot, 'sales_amount')
g.set_axis_labels("Sales Amount (US Dollars)", "Number of Sales")
g.fig.subplots_adjust(top=0.9)
g.fig.suptitle('Distribution of Dealership and Internet Sales From 2015 to 2019', size = 20)
g.add_legend()
```

Out[172]: <seaborn.axisgrid.FacetGrid at 0x7fe96beb3580>

Out[172]: <seaborn.axisgrid.FacetGrid at 0x7fe96beb3580>

Out[172]: Text(0.5, 0.98, 'Distribution of Dealership and Internet Sales From 2015 to 2019')

Out[172]: <seaborn.axisgrid.FacetGrid at 0x7fe96beb3580>



I decided to leave the blank 'Typical Value' graphs from 2015 and 2016 to show that there were no typical value sales during these years.

Part 4: Takeaways from the analysis

Provide your thoughts about the analysis above by answering the following questions in the blank markdown cells provided below. No code should be run for this section.

1. (Part 1) What are some potential hypotheses as to why the top 5 performing states have the highest sales amounts? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

2. (Part 1) What are some potential hypotheses as to why the bottom 5 performing states have the lowest sales amounts? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

3. (Part 2): How would you characterize the historical performance of the dealerships visualized in Part 2 (e.g. good, bad, growing, declining, etc.)? Describe some of the trends in relative performance over time for the dealerships. Be specific and cite specific elements of the visualization created in Part 2 to support your claims. Specify any additional factors you would want to consider that would influence your performance assessment.

4. (Part 3): How does sales amount compare for the two channels (internet and dealership)? Is one channel generating more sales than another? Is one channel generating higher sales amounts than the other? Does this comparison change year-over-year? Does this comparison change by sales type? Be specific and cite specific elements of the visualization created in Part 3 to support your claims.

5. (Part 3): What are some potential hypotheses as to why the distribution of sales amounts compared across channel, year, and sales type behaves in the manner you described in 4.4? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

Part 4 Responses

For each of the following questions, answer in as much preciseness and clarity that you can. Refer back to the tables and plots that you have created to back up your answers if necessary. Answer each question in the cell below. You are NOT to code anything for this section. This is for you to reflect on the analysis developed in response to Parts 1-3.

1. (Part 1) What are some potential hypotheses as to why the top 5 performing states have the highest sales amounts? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

States in the top 5 performing states are bigger states with higher population density (ex. Texas, California, Florida). To test this, I would find data on these state's population for the years of interest and compare the proportion of number of sales per population density and see if these states are still the top 5 performing states when taking population density into account.

2. (Part 1) What are some potential hypotheses as to why the bottom 5 performing states have the lowest sales amounts? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

Similar to the last question, the bottom 5 performing states are smaller states/states with smaller population density. To test this, I would find data on these state's population for the years of interest and compare the proportion of number of sales per population density and see if these states are still the bottom 5 performing states when taking population density into account.

3. (Part 2): How would you characterize the historical performance of the dealerships visualized in Part 2 (e.g. good, bad, growing, declining, etc.)? Describe some of the trends in relative performance over time for the dealerships. Be specific and cite specific elements of the visualization created in Part 2 to support your claims. Specify any additional factors you would want to consider that could influence your performance assessment.

The overall trend of cumulative sales in Texas/California dealerships is overall steadily increasing at a steady rate, since the lines are generally straight with a positive slope. Dealership 2 seems to have a recent growing increase of sales as it's slope has increased beginning in the middle of 2018. Meanwhile dealership 3 seems to have a steady decline in sales, as it's slope has decreased since the beginning of 2017. Dealership 14 also is seeing a recent decline in sales since the mid/end of 2018, with a significant decrease in slope (almost no new sales - horizontal line). I'm wondering if the dealerships with decrease in sales belong to the same state/region as dealerships with increased sales, suggesting more people prefer one dealership to another.

4. (Part 3): How does sales amount compare for the two channels (internet and dealership)? Is one channel generating more sales than another? Is one channel generating higher sales amounts than the other? Does this comparison change year-over-year? Does this comparison change by sales type? Be specific and cite specific elements of the visualization created in Part 3 to support your claims.

It appears that dealerships are generating more sales than website sales in all aspects (year and sales type). The difference in channel seems to be about the same depending on sales type, with both channels generating more sales for 'Low' sales type than 'Typical' or 'High' sales types (ex. some dealership sales reaching in the thousands for 'Low' sales while only reaching in the 100s for 'Typical' or 'High' sales). Differences in channel also seem to be about the same throught the years, with maybe an exception for 2015 which has a higher proportion of website sales than other years. Sales seemed to have increased from years 2016-2018, with about half the total sales in 2015 and 2019 for each sales type.

5. (Part 3): What are some potential hypotheses as to why the distribution of sales amounts compared across channel, year, and sales type behaves in the manner you described in 4.4? Describe how you would test your hypotheses in further analysis. Do not conduct any additional analyses or write any more queries, just describe in words.

I would predict that dealerships get more sales than websites because customers would want to see the product in person, and possibly pushy salespeople would have more of an influence on making sales than a website. If I were to collect more data, I could test these claims by creating a survey poll at the time of sale asking customers to select which factor most influenced their purchase (ex. seeing the product first, pusy salesman, etc). If I wanted to only use the data provided, I could take a deeper dive and see if the product type had any influence on sales channel, year, or sales type (ex. Scooter vs. Automobile).