perplexity

So these are the experiments for my DMV Practical tomorrow where I'll randomly be assigned one of these. I'll manage the coding part but the problem is VIVA.

So assume you are the examiner/external supervisor, and ask questions on each experiment based on the 3 tiers- basic, intermediate and advanced. You can even ask to solve any certain query like an examiner would.

Refer to both, code files and manual. And then ask questions and query to perform. Can you also be considerate enough to provide the answers as well ▯

Absolutely! Here’s the structured set of *examiner-style* viva (oral exam) questions you might face for each DMV experiment covered in both your **lab manual** and **code notebooks**—arranged by basic, intermediate, and advanced tiers, plus *certain tasks/queries to solve* as an examiner could ask. Each set is immediately followed by well-structured answers. This holistic approach blends technical, practical, and oral-exam nuances, tailored for your practical evaluation.

Experiment 1: Analyzing Sales Data from Multiple File Formats

Basic

- **Q1:** What are the steps to load sales data from multiple file formats in Python?
- **Q2:** Which pandas functions are used for reading CSV, Excel, and JSON files?

Answer:

- Use `pd.read_csv()` for CSV, `pd.read_excel()` for Excel, and `pd.read_json()` for JSON.
- Steps: Import libraries → Read each format into DataFrames → Integrate/clean → Analyze.

Intermediate

- **Q3:** How do you clean missing or inconsistent data?
- **Q4:** What methods help unify data from different formats?

Answer:

- Use `.fillna()`, `.dropna()`, or transformations for missing data; `.drop_duplicates()` for redundancy; and standardize columns for consistency.
- Unify using `pd.concat([...], ignore_index=True)` or merging on common keys.

Advanced

- **Q5:** Given a merged DataFrame, how would you find total sales and average order value by product category?

Answer:

```
# Assuming unified DataFrame is 'df'
total_sales = df['SALES'].sum()
avg_order_value = df.groupby('PRODUCTLINE')['SALES'].mean()
```

Query/Task

- *“Show how to visualize sales trend over months and detect any seasonal effects.”*

Answer:

```
df['ORDERDATE'] = pd.to_datetime(df['ORDERDATE'])
monthly = df.groupby(df['ORDERDATE'].dt.to_period('M'))['SALES'].sum()
monthly.plot(kind='line')
```

Experiment 2: Analyzing Weather Data from OpenWeatherMap API

Basic

- **Q1:** What parameters can you retrieve from OpenWeatherMap API?
- **Q2:** What is an API key?

Answer:

- Typical weather parameters: temperature, humidity, wind speed, pressure, weather description.
- API key: Unique string for authentication/access to API endpoints.

Intermediate

- **Q3:** How do you perform data cleaning on API weather data?
- **Q4:** Describe the process to convert JSON API data to a pandas DataFrame.

Answer:

- Check for missing (`NaN`) values; forward-fill or interpolate as needed; standardize date/time.
- Load API response JSON → extract relevant fields → `pd.DataFrame()` constructor.

Advanced

- **Q5:** How would you model and visualize trends such as weekly temperature variations?

Answer:

```
weather['datetime'] = pd.to_datetime(weather['datetime'])
weekly = weather.set_index('datetime').resample('W')['temperature'].mean()
weekly.plot(kind='line')
```

Query/Task

- *“Write code to compute correlation between weather attributes such as temp and humidity.”*

Answer:

```
weather[['temperature','humidity']].corr()
```

Experiment 3: Data Cleaning and Preparation (Customer Churn)

Basic

- **Q1:** What is the importance of data cleaning?
- **Q2:** Name two strategies for handling missing values.

Answer:

- Ensures accuracy, removes bias, enables reliable analysis.
- Remove rows/columns, or impute (mean/median fill).

Intermediate

- **Q3:** How do you detect outliers in a dataset?
- **Q4:** What is feature engineering?

Answer:

- Visual (boxplot), statistical (IQR, Z-score) analysis.
- Creating new features from existing ones to enhance model prediction.

Advanced

- **Q5:** You find a column with mixed date formats. How would you standardize it?

Answer:

```
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
```

Query/Task

- *"Check if any columns have > 20% missing data, and how would you handle them?"*

Answer:

```
missing = df.isnull().mean()
high_missing = missing[missing > 0.2]
df = df.drop(columns=high_missing.index)
```

Experiment 4: Data Wrangling on Real Estate Data

Basic

- **Q1:** What's the difference between data cleaning and transformation?
- **Q2:** Why is data wrangling important for ML?

Answer:

- Cleaning = fix/remove "bad" data; transformation = reformat or derive new features.
- Ensures the model ingests accurate, consistent data.

Intermediate

- **Q3:** How do you handle categorical variables in wrangling?

Answer:

- Employ encoding: label encoding, one-hot encoding (`pd.get_dummies()`).

Advanced

- **Q4:** How do you process outliers or extreme values when preparing for regression?

Answer:

- Cap/floor, transformation (e.g., log), or removal based on domain impact.

Query/Task

- *“Show code to aggregate average price by property type.”*

Answer:

```
df.groupby('property_type')['price'].mean()
```

Experiment 5: Data Visualization using Matplotlib (AQI Trends)

Basic

- **Q1:** List at least three plot types supported in Matplotlib.

Answer:

- Line, bar, scatter, histogram, pie, box plot.

Intermediate

- **Q2:** What’s the difference between Pyplot and the OOP interface in Matplotlib?

Answer:

- Pyplot: simple API, quick; OOP: advanced control, creates Figure/Axes objects for subplots, customization.

Advanced

- **Q3:** How would you make a multi-line plot for multiple pollutant trends?

Answer:

```
plt.plot(df['date'], df['PM2.5'], label='PM2.5')
plt.plot(df['date'], df['PM10'], label='PM10')
plt.legend()
plt.show()
```

Query/Task

- *"Suppose AQI for a recent day is unusually high. How would you highlight it on a time-series plot?"*

Answer:

- Add a vertical line (`plt.axvline()`) or annotate with `plt.annotate()`.

Experiment 6: Data Aggregation (Retail Sales)

Basic

- **Q1:** What is data aggregation? Provide a business use-case.

Answer:

- Combining/grouping data to produce concise, meaningful summaries. E.g., sales by region for business insights.

Intermediate

- **Q2:** How do you aggregate data in pandas?

Answer:

- Using `groupby()` with functions like `.sum()`, `.mean()`, `.count()`, etc.

Advanced

- **Q3:** Write code to find and visualize top 3 regions by sales.

Answer:

```
region_sales = df.groupby('region')['sales'].sum()
top_regions = region_sales.nlargest(3)
top_regions.plot(kind='bar')
```

Query/Task

- *“Create a pie chart of sales distribution by product category.”*

Answer:

```
df.groupby('product_category')['sales'].sum().plot.pie()
```

Experiment 7: Stock Market Time Series Analysis & Visualization

Basic

- **Q1:** What are candlestick charts used for?

Answer:

- Visual representation of open, close, high, low for each trading period; used for stock analysis.

Intermediate

- **Q2:** How do you compute a moving average in pandas?

Answer:

```
df['MA20'] = df['Close'].rolling(window=20).mean()
```

Advanced

- **Q3:** How would you forecast stock prices? Name an algorithm.

Answer:

- Use ARIMA/SARIMA, exponential smoothing—fit model, predict future values.

Query/Task

- *“Given a DataFrame with a ‘Close’ price column, fit an ARIMA model and forecast next 5 days.”*

Example Code:

```
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(df['Close'], order=(5,1,0))
fit = model.fit()
forecast = fit.forecast(steps=5)
```

Pro Tips for Viva:

- Whenever code is requested, narrate the algorithm/logic and *then* the line of code, as per examiner style.
- If given a visualization query, describe which plot is suitable and why, then what insights you expect.
- Always link practical steps (from code) back to your underlying concepts (from the manual).

Let me know if you want flashcards, randomizer, or further drill-down per experiment for rapid revision!

⁂

1. DMV-Lab-Manual_print_ready.pdf
2. DMV_7.ipynb
3. DMV_2.ipynb
4. DMV_1.ipynb
5. DMV_4.ipynb
6. DMV_3.ipynb