ECON 0150 | Economic Data Analysis

The economist's data analysis pipeline.

Part 2.5 | Merging Data

Final Project Preview

Exploring real questions with real data

Lets start working through a project together:

- ***Question 1****: Were there systematic patterns in vote share changes between 2020 and 2024?*
- ***Question 2****: Do income levels relate to these voting shifts?*
- ***Data****: MIT Election Lab (county-level returns) + Census (median income)*

> *this is the kind of analysis you'll do for your final project*

The Data Challenge

We have two separate datasets that need to be connected

Dataset 1: Presidential election results by county

- *County name*
- *State*
- *Vote counts for 2020*
- *Vote counts for 2024*

Dataset 2: Median household income by county

- *County identifier*
- *Median income*

Step 1: Explore Vote Shares

Q. What do county-level vote shares look like?

First understand what we’re working with.

```
# Histogram of 2024 vote shares
sns.histplot(elections, x='2020')
```

```
# Histogram of 2024 vote shares
sns.histplot(elections, x='2024')
```

Step 2: Compare Elections

Q. How did county vote shares change between 2020 and 2024?

Second, lets look at the relationship between Democratic Share in 2020 and 2024.

```
# Scatterplot comparing elections
sns.scatterplot(elections, x='2020', y='2024', alpha=0.5)

# Add 45-degree line
plt.plot([0,1], [0,1], 'r--', alpha=0.5)
```

> *points above the line shifted more Democratic*

> *points below the line shifted more Republican*

> *but what explains these shifts?*

Step 3: Add Income Data

Q. Does county income relate to voting shifts?

To answer this, we need to:

1. Load the income data

*2. **Merge** it with our election data*

3. Calculate the vote share change

4. Visualize the relationship

> but how do we connect two separate datasets?

Merging Data: The Concept

Combining datasets based on common identifiers

The Key: Find a common column that uniquely identifies observations

- *In our case: County FIPS codes (Federal Information Processing Standards)*
- *FIPS uniquely identify every US county*
- *Format: State code (2 digits) + County code (3 digits) = 5 digits total*

> *example: Allegheny County, PA = 42003*

Types of Merges

Different ways to combine datasets

Merge Type	Description	Example
1:1	Each row in A matches exactly one row in B	County → County
1:m	One row in A matches multiple rows in B	State → Counties
m:1	Multiple rows in A match one row in B	Counties → State

> our county merge is 1:1 - each county appears once in each dataset

Step 3: Perform the Merge

Combining our datasets

```
# Merge datasets on county FIPS
data = pd.merge(elections,
                  income,
                  left_on='county_fips',
                  right_on='county_fips',
                  how='inner')
```

Merge options:

- `inner`: *Keep only counties in both datasets*
- `left`: *Keep all counties from elections data*
- `right`: *Keep all counties from income data*
- `outer`: *Keep all counties from either dataset*

> *we use 'inner' to focus on counties with complete data*

Step 4: Calculate Vote Shifts

Creating our analysis variable

```
# Calculate the shift in Democratic vote share
data['dem_shift'] = data['2024'] - data['2016']
```

```
# Summarize this new variable
sns.histplot(elections, x='dem_shift')
```

> *now we can explore the relationship with income*

Step 5: Analyze the Relationship

Q. Does county income relate to voting shifts?

```
# Scatterplot of income vs vote shift
sns.scatterplot(data,
                x='median_income',
                y='dem_shift',
                alpha=0.3)

# Add horizontal line at zero
plt.axhline(y=0, color='r', linestyle='--', alpha=0.5)

plt.xlabel('Median Household Income ($)')
plt.ylabel('Change in Democratic Vote Share (2024-2016)')
```

> what patterns do you see?

Common Merge Issues

Watch out for these problems

- ***Missing values***: *Some counties might not have income data*
- ***Duplicate keys***: *Same county appearing multiple times*
- ***Type mismatches***: *FIPS stored as numbers vs strings*
- ***Different naming***: *"St. Louis" vs "Saint Louis"*

```
# Check for duplicates before merging
elections['county_fips'].duplicated().sum()
income['FIPS'].duplicated().sum()
```

Summary

Merging allows us to answer richer questions

- ***Identify*** *common columns to join on (FIPS codes)*
- ***Prepare*** *data for merging (create consistent identifiers)*
- ***Merge*** *using appropriate join type (inner, left, right, outer)*
- ***Transform*** *to create analysis variables (vote shift)*
- ***Analyze*** *the combined dataset*

ECON 0150 | Economic Data Analysis

Part 2: Data Operations Practice

Practice Problems for MiniExam 2

Practice 1: Trace the Filter

Which products remain after filtering?

Product_ID	Category	Price	In_Stock
P001	Electronics	299	True
P002	Clothing	49	False
P003	Electronics	89	True
P004	Food	12	True
P005	Clothing	79	True

Filter: `(Price < 100) AND (In_Stock == True)`

Answer: P003, P004

Practice 2: Multi-Step Operations

Track data through multiple transformations

Sale_ID	Store	Amount
S001	North	120
S002	South	80
S003	North	150
S004	South	90
S005	North	100

Operations:

1. Filter for Amount >= 100

2. Group by Store

3. Calculate mean Amount

Answer: North, 125

Practice 3: Data Cleaning Decisions

What cleaning is needed for each entry?

Response_ID	Duration
R001	“5 minutes”
R002	“180”
R003	“about 3 min”
R004	“N/A”

For each entry, select ALL that apply:

R001: [Extract number] [Remove text] [Convert type] [Handle missing] [Already clean]

R002: [Extract number] [Remove text] [Convert type] [Handle missing] [Already clean]

R003: [Extract number] [Remove text] [Convert type] [Handle missing] [Already clean]

Practice 4: Build Complex Filters

Construct the correct boolean logic

Goal: Find all employees who:

- *Work in either Tech or Sales departments*
- *AND have been with company more than 2 years*
- *AND earn less than $70,000*

Use these components to construct a filter:

1. (Department == ‘Tech’)

2. (Years > 2)

3. (Department == ‘Sales’)

4. (Salary < 70000)

Answer: (1 OR 3) AND 2 AND 4

Practice 5: Choose the Right Transformation

Why transform data?

Scenario: Comparing test scores across different schools where class sizes vary dramatically (10-50 students)

You have:

- *Total_Points_Earned (all students combined)*
- *Number_of_Students*

Which transformation makes schools comparable?

a. Total_Points_Earned + Number_of_Students
b. Total_Points_Earned - Number_of_Students
c. Total_Points_Earned / Number_of_Students
*d. Total_Points_Earned * Number_of_Students*

Answer: c) Creates average score per student

Practice 6: Predict Grouping Output

What will the grouped data look like?

Order_ID	Customer	Amount	Region
O001	Alice	50	East
O002	Bob	30	West
O003	Alice	70	East
O004	Charlie	40	East
O005	Bob	60	West

We’ve `Grouped by Customer` then `Summed by Amount`.

How many rows in output? _____

What’s the sum for Bob? _____

Which customer has highest total? _____

Answers: 3 rows, 90, Alice (120)

Tips for MiniExam 2

Key concepts to remember

Filtering:

- *AND: both conditions must be true*
- *OR: at least one condition must be true*

Grouping:

- *Output has one row per group*
- *Choose the right aggregation (sum, mean, count, etc.)*

Tips for MiniExam 2

Key concepts to remember

Transformations:

- *Division normalizes for fair comparison*
- *Log transformation helps with different scales*

Data Cleaning:

- *Text → Number needs type conversion*
- *Missing values: drop or fill (not both!)*
- *Consistent format before analysis*