

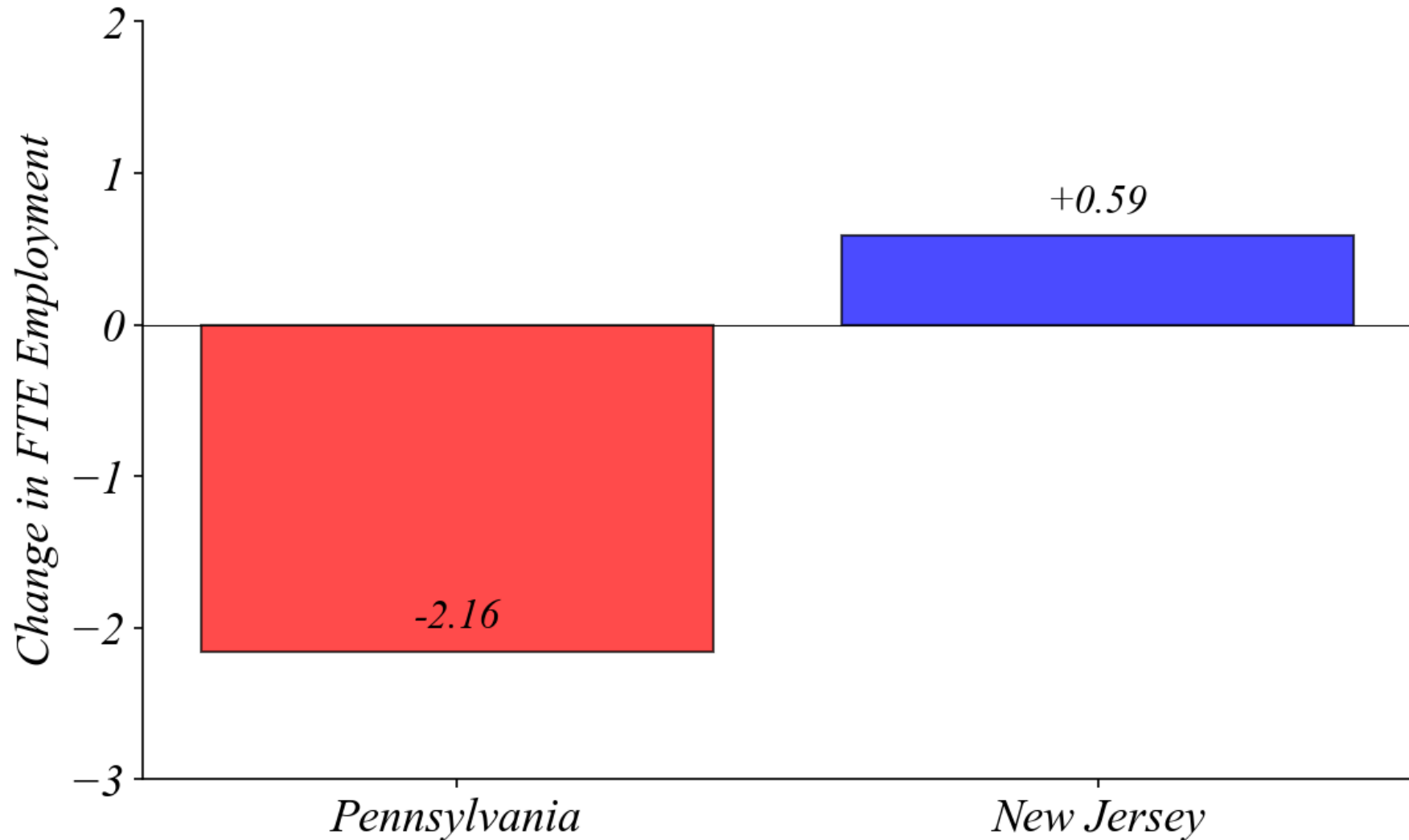
# ECON 0150 | Economic Data Analysis

*The economist's data analysis skillset.*

## *Part 1.4 | Panel Data (Long Format)*

# Part 0 | Minimum Wage Study

*NJ raised its minimum wage. Employment did not fall.*



# They Measured Changes

*The same 410 stores, surveyed twice*

Store	Wave	State	FTE
1	1	NJ	20.5
1	2	NJ	24.0
2	1	PA	18.0
2	2	PA	17.5
...	...	...	...

> *same stores, two time periods, tracking each store's change*

# Panel Data

*Data with repeated observations across entities AND time*

- ***Panel data** tracks the same units (people, countries, stores) over time*
- *Examples: Card & Krueger's minimum wage study, GDP by country by year, sales by store by hour*
- *Can be stored in two formats: **Long** or **Wide***

# Long Format Panel Data

*Each observation is a separate row*

Shop	Hours	Quantity
Shop A	12	1
Shop A	15	2
Shop A	14	2
Shop A	16	2
Shop A	19	1
...	...	...

> *Shop (i) and Hours (t) are indexes; Quantity (x) is the variable*

# Labor Demand: Hiring a Barista

*Use Coffee\_Sales\_Receipts.csv to help understand where to hire a new barista.*

- *You manage three coffee shops and are considering where to hire a new barista.*
- *You have a dataset containing information about the transactions taking place at all three coffee shops throughout the day.*
- *Lets consider how to use this data to inform our decision.*

# Hiring a Barista

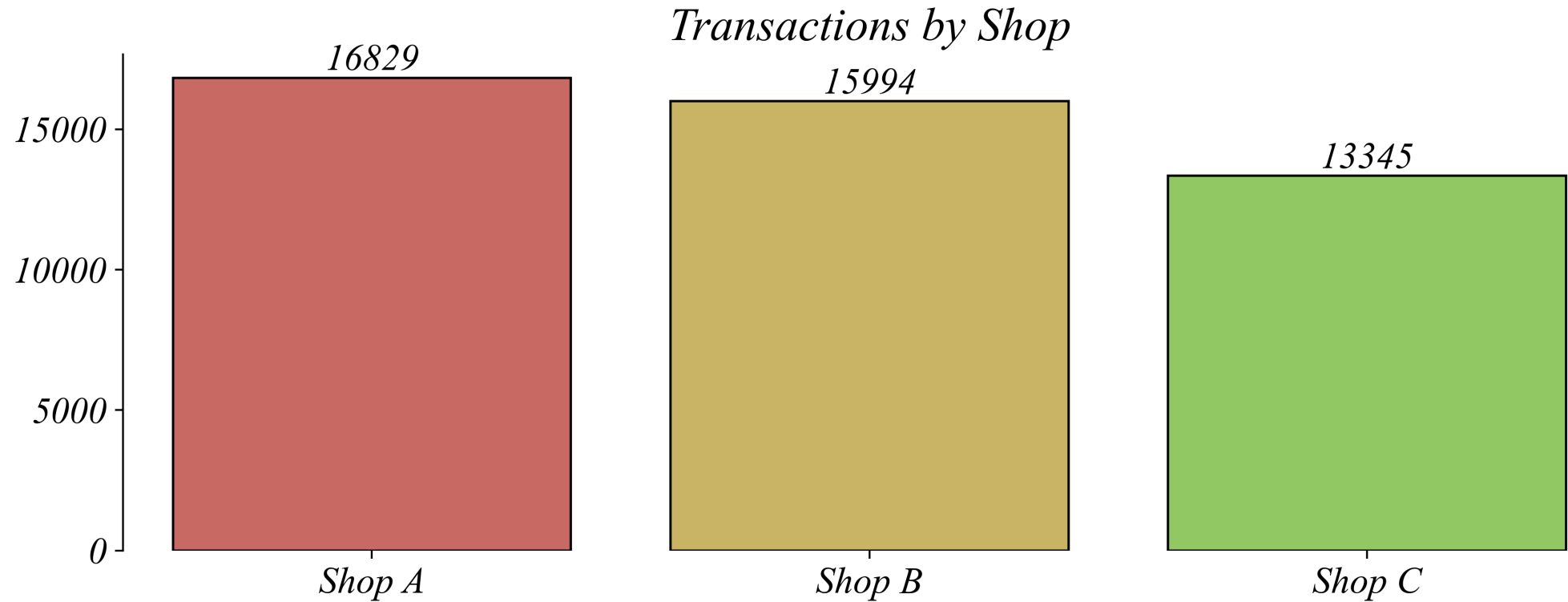
*Q. Which coffee shop is the busiest?*

Shop	Hours	Quantity
Shop A	12	1
Shop A	15	2
Shop A	14	2
Shop A	16	2
Shop A	19	1
...	...	...

> *as is usually the case, it's difficult to know without summarization*

# Hiring a Barista: Bar Graphs Compare Shops

*Q. Which coffee shop is the busiest?*



*> a bar chart makes it easy to compare shops' busyness*

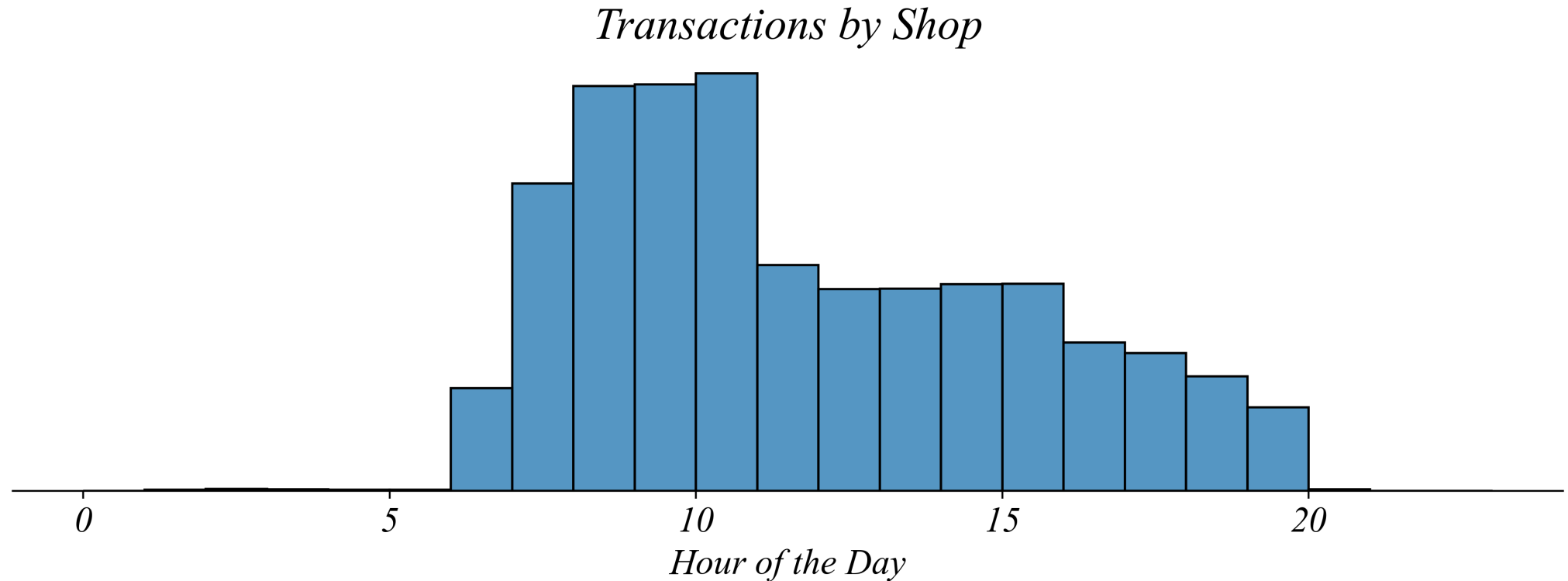


# Hiring a Barista

*Q. What time of day is the busiest?*

# Hiring a Barista: Histograms Can Compare Times

*Q. What time of day is the busiest?*



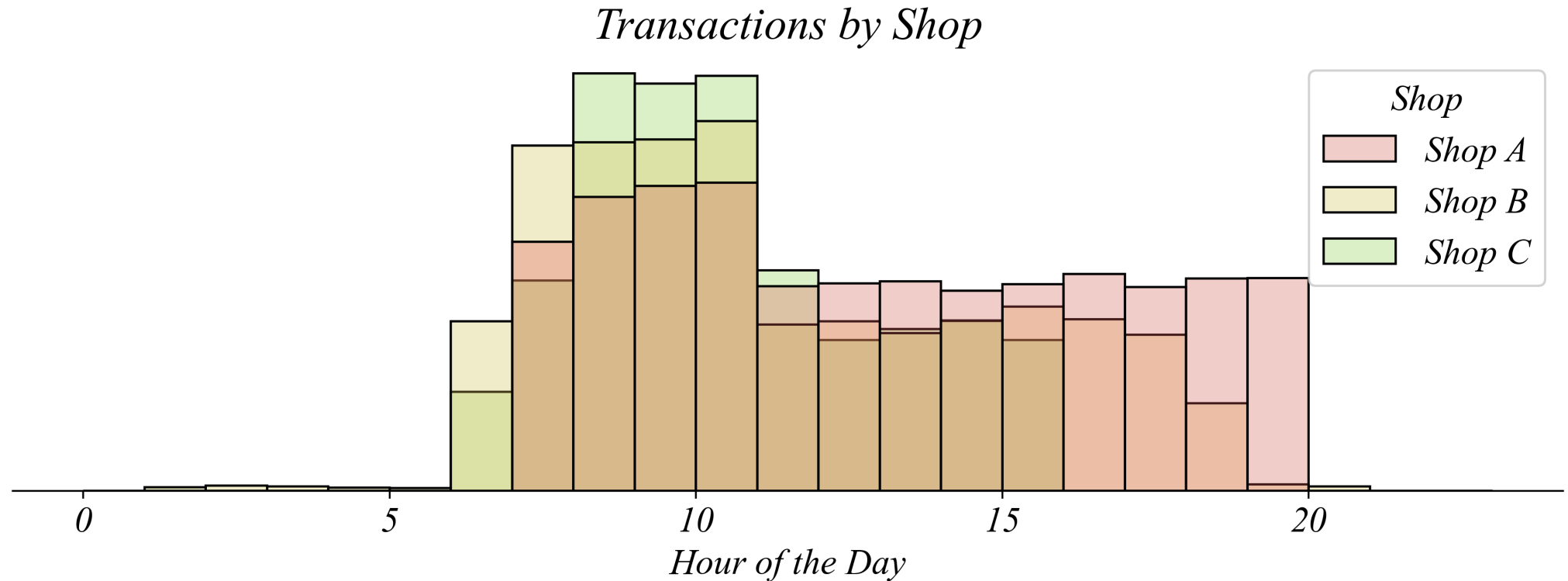
- > *a histogram makes it easy to compare transactions by time of day*
- > *does this mean the morning shift at Shop A is the busiest?*

# Hiring a Barista

*Q. Which shift is the busiest?*

# Hiring a Barista: Transactions by Shop

*Q. Which shift is the busiest?*

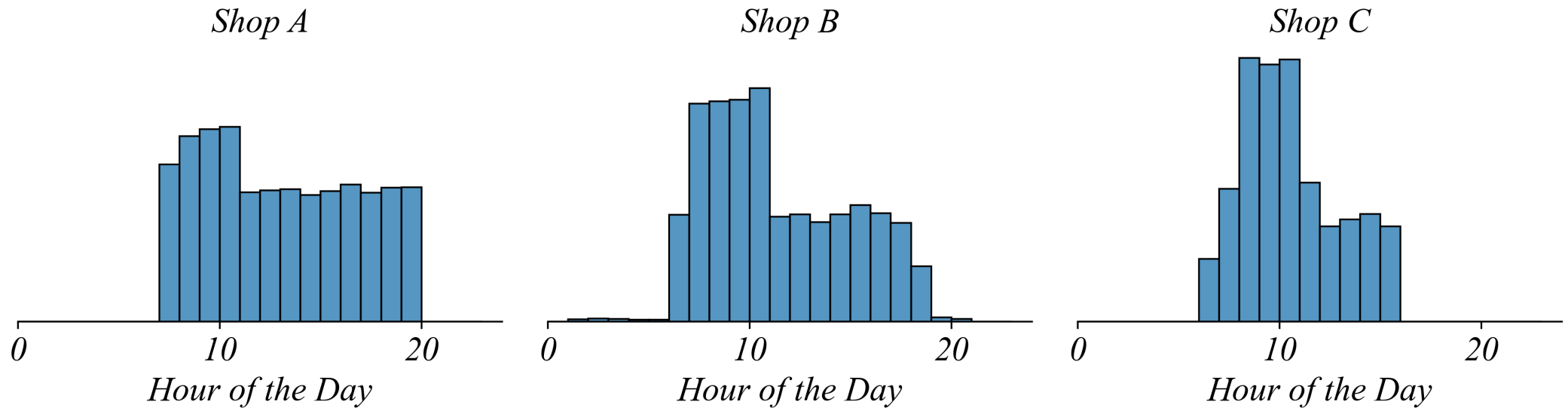


- > *an overlaid histogram can show all three groups*
- > *does this show the data clearly?*

# Hiring a Barista: Faceting

*Each shop gets its own panel*

*Transactions by Shop*

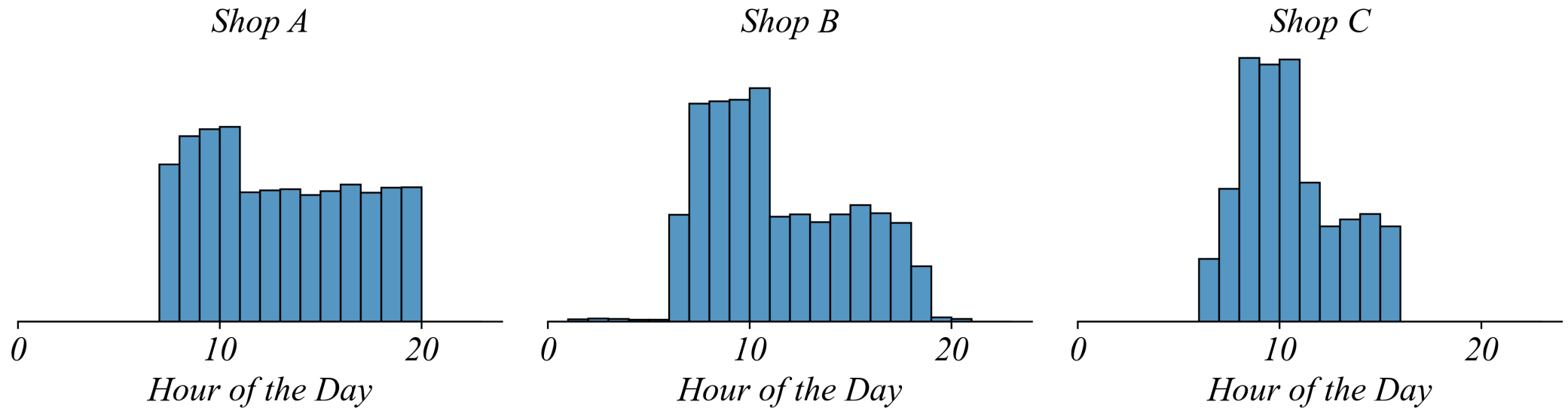


> *same data, but now each shop has its own histogram*

# Hiring a Barista: Faceting

*Q. Which shop has the most consistent traffic throughout the day?*

*Transactions by Shop*

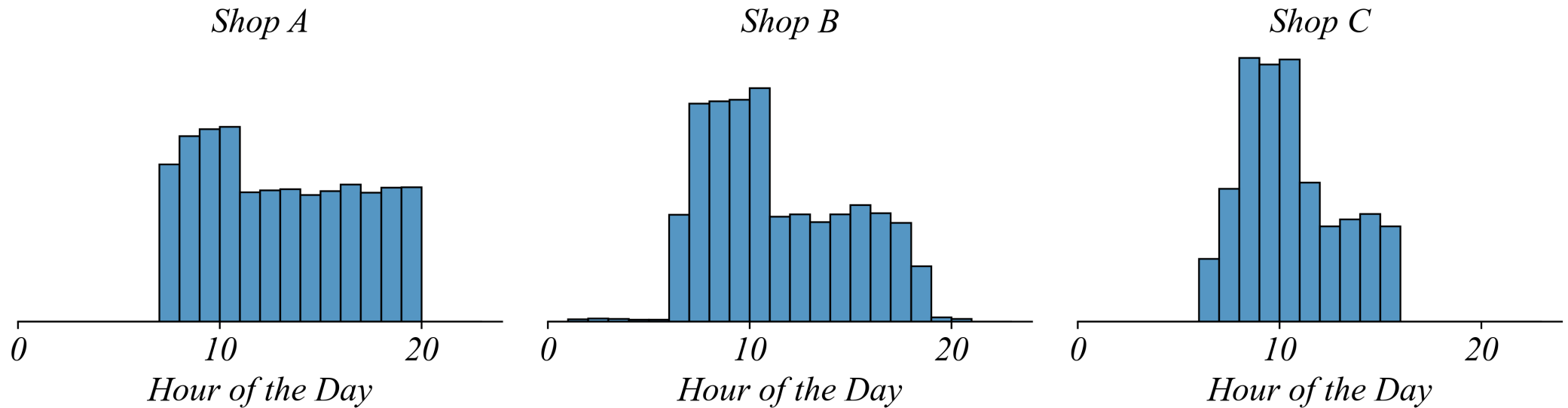


> *Shop A — the distribution is relatively flat*

# Hiring a Barista: Faceting

*Q. Which shop is busiest during the morning rush?*

*Transactions by Shop*

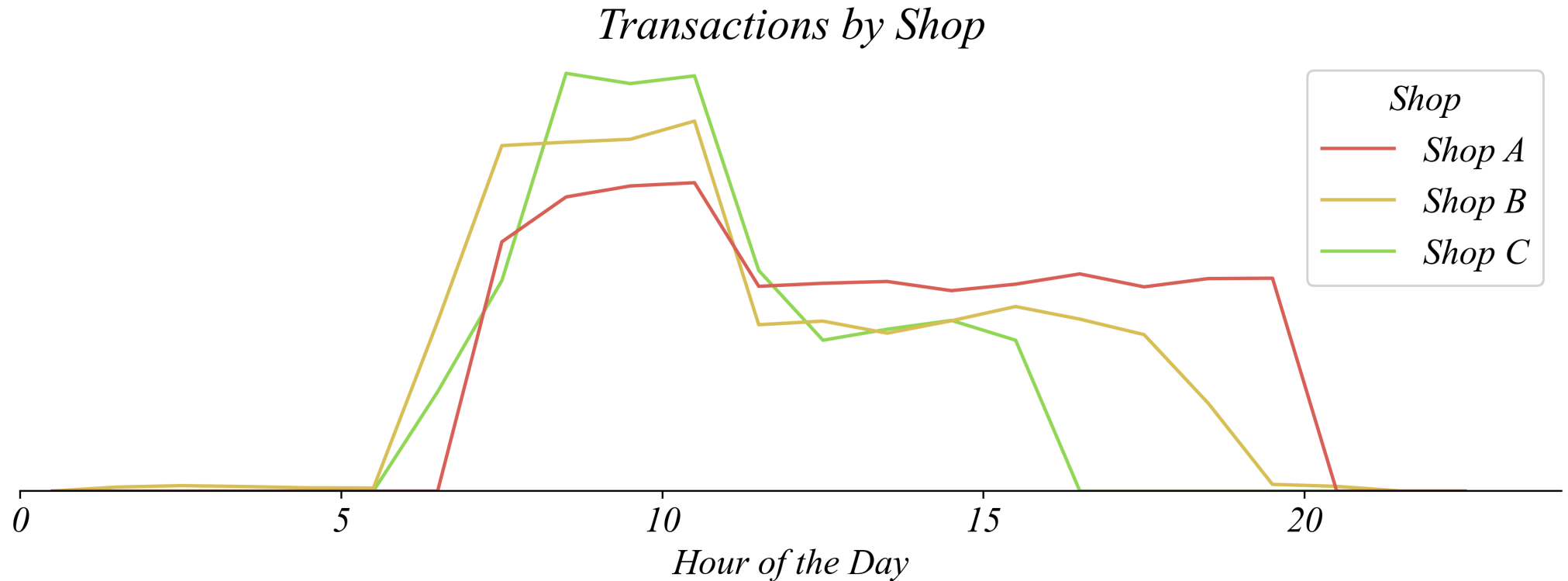


> *Shop C — compare the 8-10am peaks across panels*

> *but since the histograms are separated it's not as easy to make the comparison*

# Hiring a Barista: Line Graphs

*Q. Which shop is busiest during the morning rush?*



- > *line graphs can also show comparisons between groups clearly*
- > *Shop C — easier to see the 8-10am peak across shops*



# Part 1.4 | Panel Data (Long Format)

## Summary

- *Panel data* tracks repeated observations across groups or time
- *Long format:* Each observation is a row; a column identifies the group
- *Faceting* separates each group into its own panel for comparison
- *Line graphs* simplify multiple categories on a single plot

# S-T-E for Panel Data (Long Format)

*What we just did*

Step	Action
SELECT	All transactions from three coffee shops
TRANSFORM	Group by shop and hour; count transactions
ENCODE	Hour → x-position; Count → y-position; Shop → color/facet

# Exercise 1.4 | Coffee Shop Transactions

*Use Coffee\_Sales\_Receipts.csv to help inform where to hire a barista.*

```
1 # Load Dataset
2 sales = pd.read_csv(file_path + 'Coffee_Sales_Receipts.csv')
3 sales.head()
```

Shop	Hours	Quantity
Shop A	12	1
Shop A	15	2
Shop A	14	2
Shop A	16	2
Shop A	19	1
...	...	...

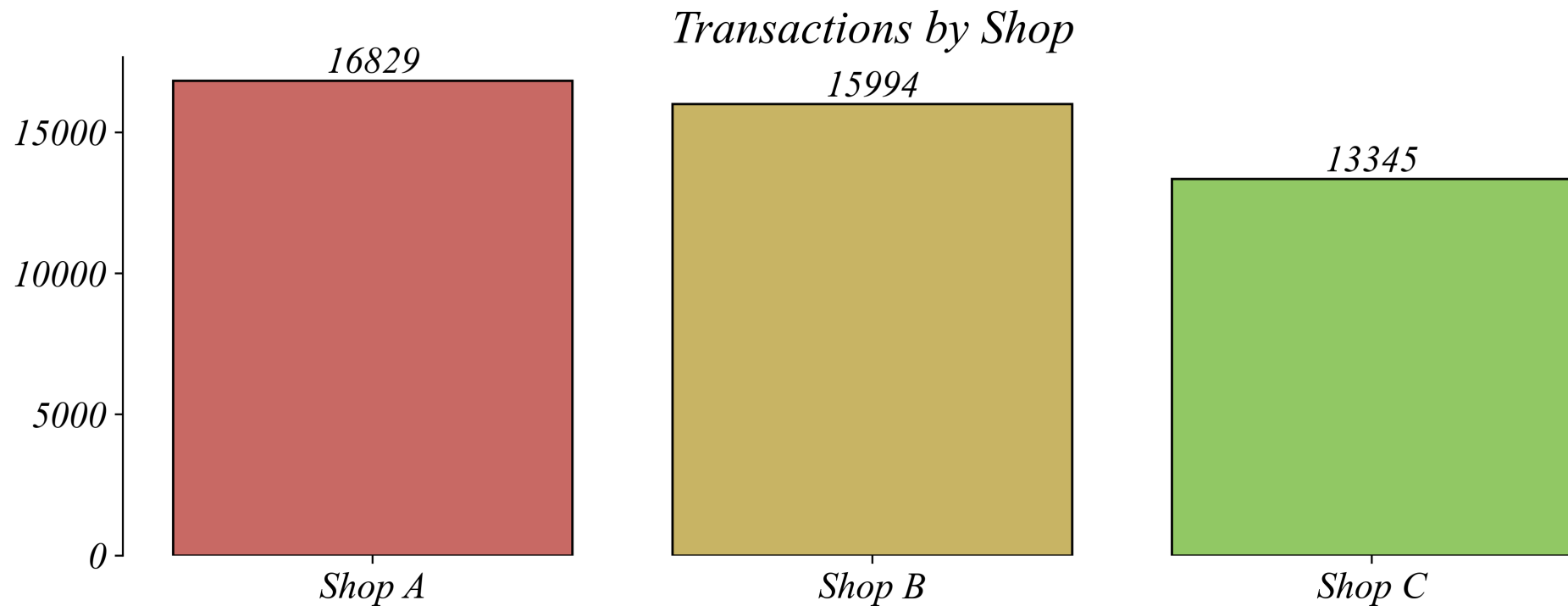
> *Shop (i) and Hours (t) are indexes; Quantity (x) is the variable*

> *this is **Long-Format Panel Data**: transactions are all in the same column*

# Exercise 1.4 | Bar Chart

*Use Coffee\_Sales\_Receipts.csv to help inform where to hire a barista.*

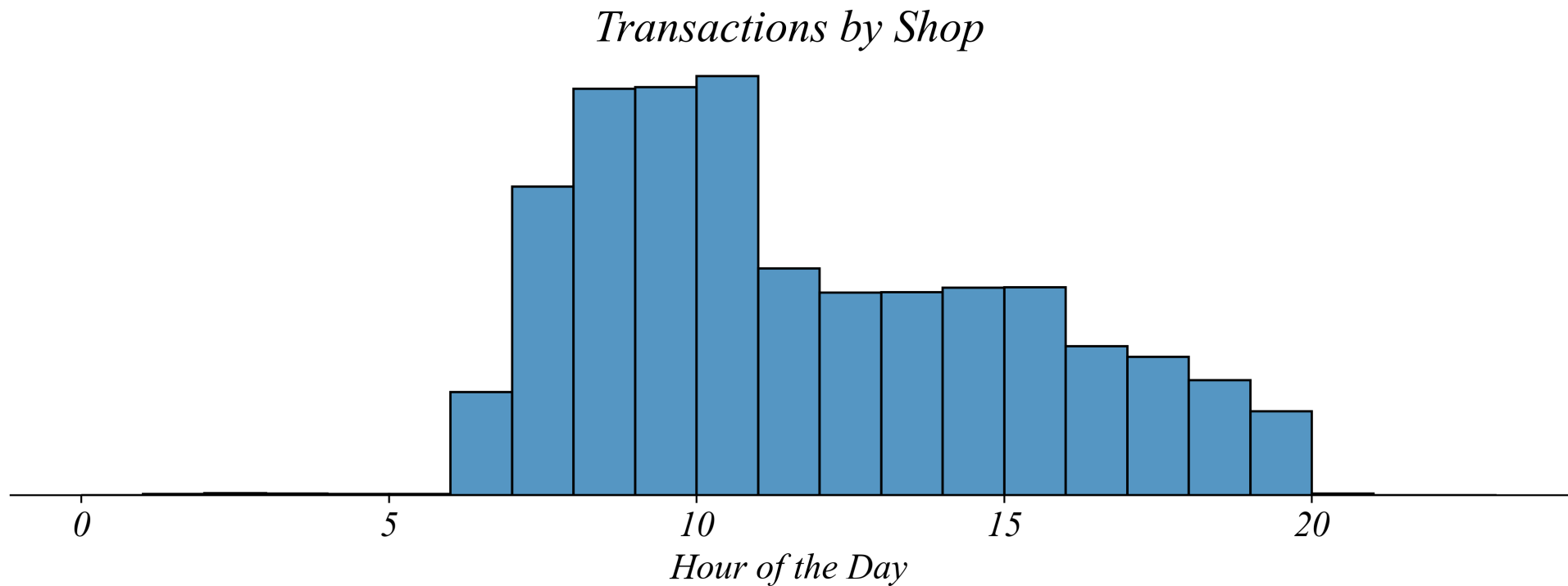
```
1 # Bar graph  
2 sns.countplot(sales, x='Shop', hue='Shop')
```



# Exercise 1.4 | Histogram

*Use Coffee\_Sales\_Receipts.csv to help inform where to hire a barista.*

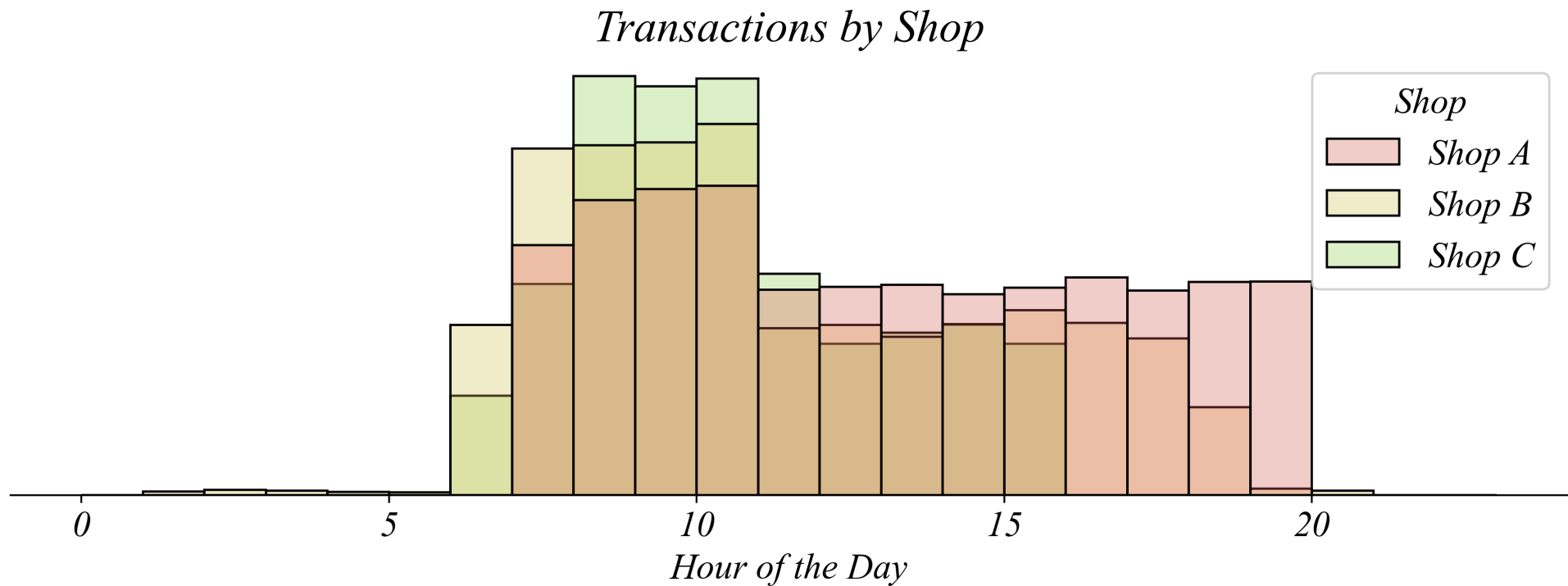
```
1 # Create a histogram
2 sns.histplot(sales, x='Hours', bins=range(0,24,1))
```



# Exercise 1.4 | Multi-Histogram

*Use Coffee\_Sales\_Receipts.csv to help inform where to hire a barista.*

```
1 # Create a multi-histogram
2 sns.histplot(sales, x='Hours', hue='Shop', bins=range(0,24,1))
```

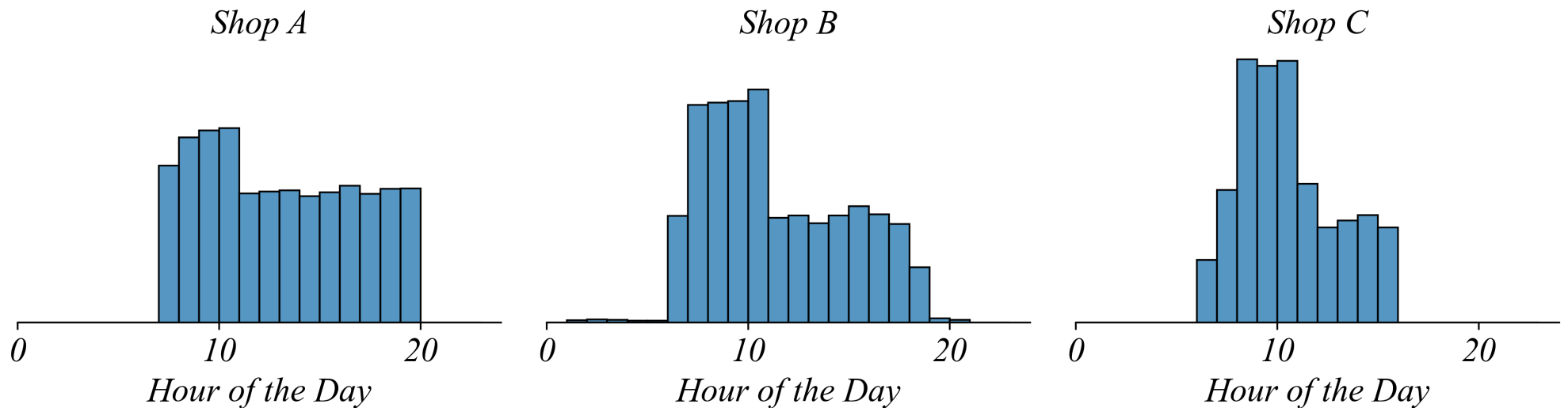


# Exercise 1.4 | Faceted Histogram

*Use faceting to give each shop its own panel.*

```
1 # Create a faceted histogram  
2 sns.displot(sales, x='Hours', col='Shop', bins=range(0,24,1))
```

*Transactions by Shop*



# Exercise 1.4 | Multiple Line Graph

*The Groupby Approach: Create a summary table, then plot*

```
1 # Create a summary table
2 counts = sales.groupby(['Shop', 'Hours']).size().reset_index(name='Count')
3 counts.head()
```

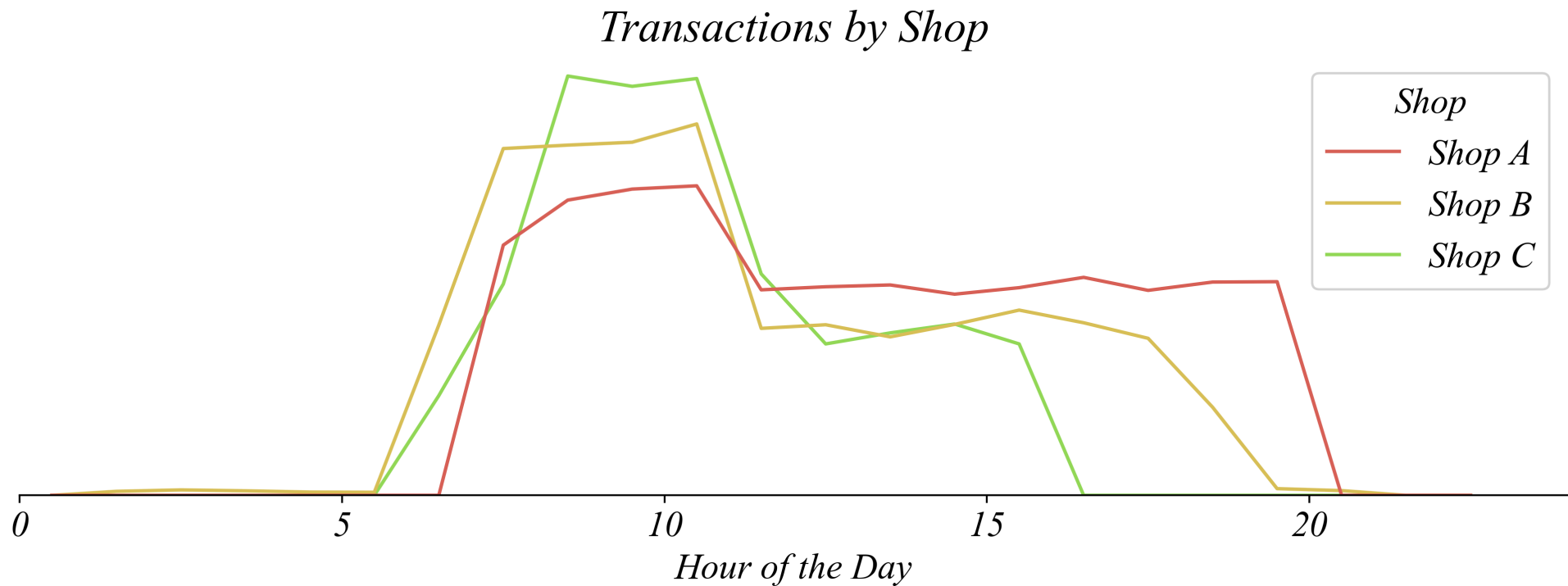
Shop	Hours	Count
Shop A	6	12
Shop A	7	45
Shop A	8	67
Shop A	9	58
Shop A	10	42
...	...	...



# Exercise 1.4 | Multiple Line Graph

*The Groupby Approach: Create a summary table, then plot*

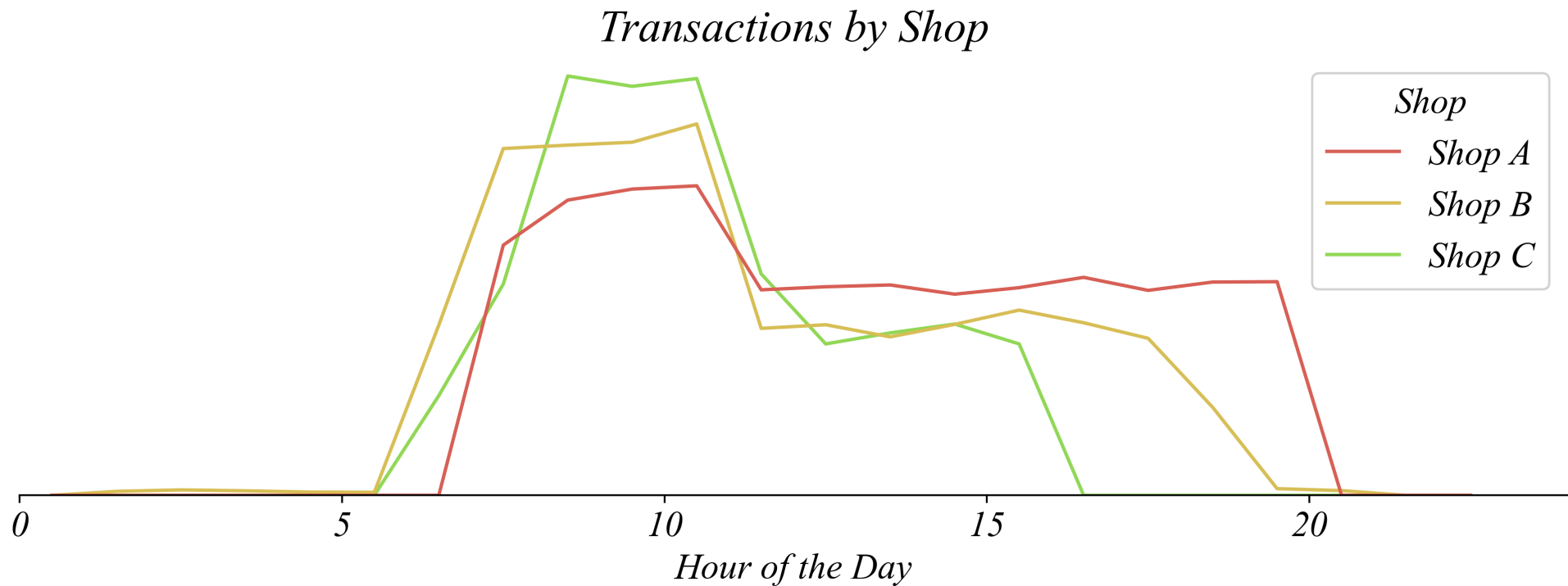
```
1 # Multiple-Line Graph using lineplot
2 sns.lineplot(counts, x='Hours', y='Count', hue='Shop')
```



# Exercise 1.4 | Multiple Line Graph (Shortcut)

*The Shortcut: Let histplot do the counting for you*

```
1 # Multiple-Line Graph using histplot shortcut
2 sns.histplot(sales, x='Hours', hue='Shop', bins=range(0,24,1), element='poly', fill=False)
```



# Looking Ahead: Part 1.5

*Sometimes we need data in a different shape*

Shop	Hours	Quantity
Shop A	12	1
Shop A	15	2
Shop A	14	2
Shop A	16	2
Shop A	19	1
...	...	...

- *What if we want to compare Hour 10 to Hour 26?*
- *Long format makes this awkward - we'd need to filter and merge*
- *Wide format makes this easy by using a column for each time period*