ECON 0150 | Economic Data Analysis

The economist's data analysis pipeline.

Part 1.6 | Grouping Data

Example 1.6 | Starbucks Offers

In starbucks_promotions.csv, which offers are most effective?

```python
# Import packages
import pandas as pd

# Load data
data = pd.read_csv("starbucks_offers.csv")
```

Starbucks Offers | The Original Table

Which offers are most effective?

We have a table of events …

>not straightforward to see which offers are most effective

Starbucks Offers | Grouping and Summing

Which offers are most effective?

Summarize total revenue by `Offer ID`:

1. Filter (if needed; keep all rows for now)

```
# Filter (no filter here yet)
#data = data[filter]
```

2. Group by `Offer ID`

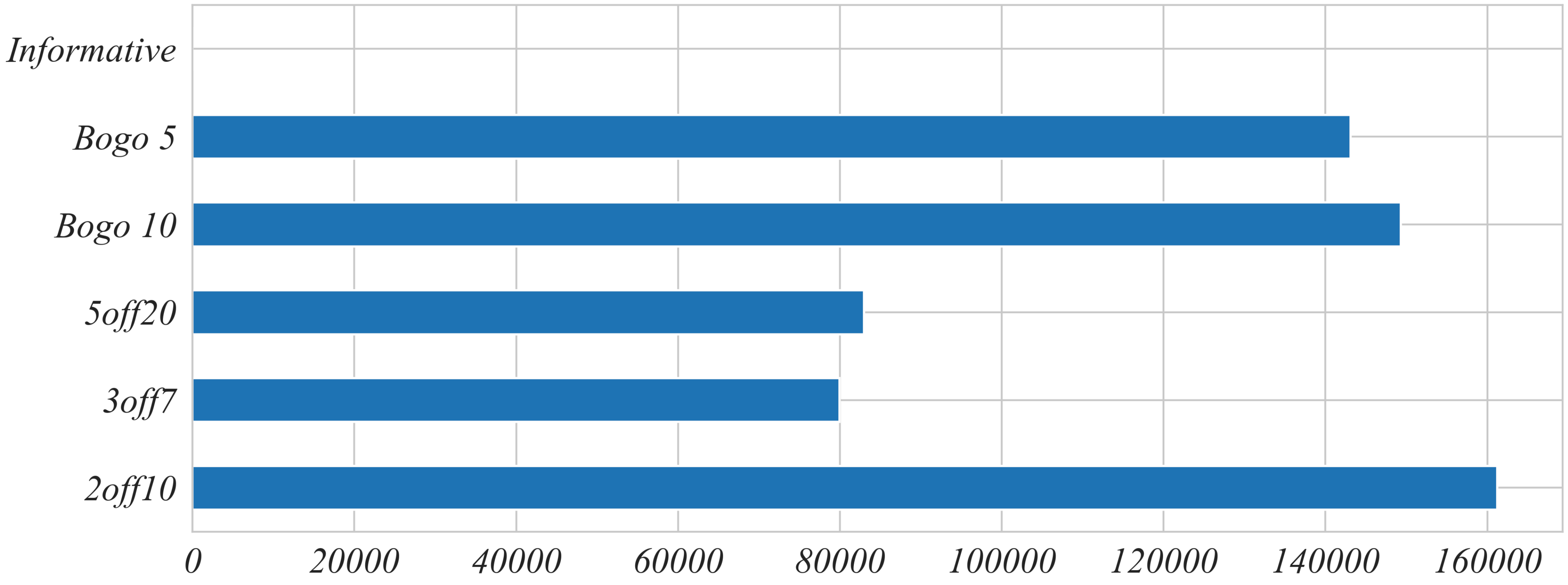
```
# Group by ID
grouped_by_id = data.groupby("Offer ID")
```

3. Sum revenue by group

```
# Sum revenue by group
grouped_revenue = grouped_by_id["Revenue"].sum()
```

Starbucks Offers | Grouping and Summing

Which offers are most effective?

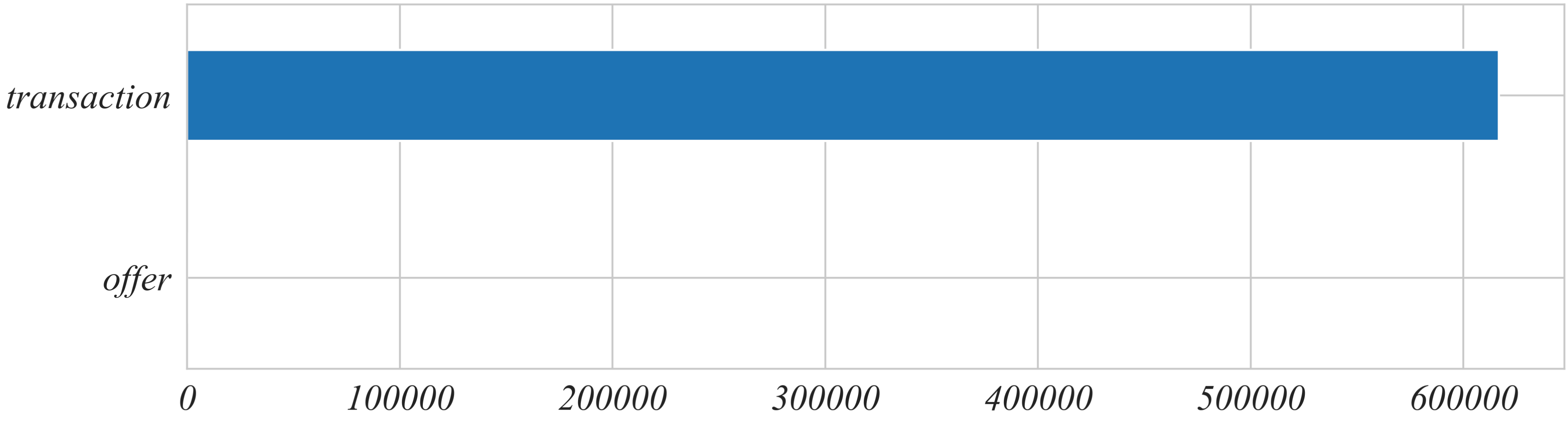


Starbucks Offers | Grouping

Which offers are most effective?

We can group on any cateogrical variable, like `Event`:

```python
# Summarize total revenue by 'Event'
event_summary = data.groupby("Event")["Revenue"].sum()
```



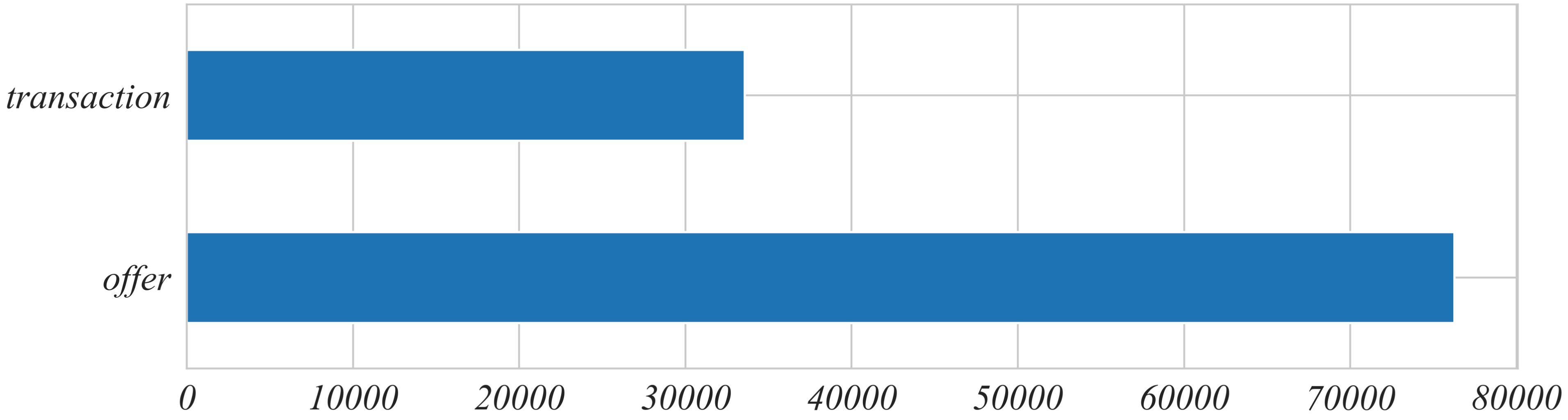
> “Offer” and “Offer Completed” events have 0 revenue, so you’ll see zeros for those rows

Starbucks Offers | Use Grouping to Count

But how many offers are there per group?

Instead of summing, count how many rows there are for each event type:

```
# Count number of each event
event_count = data.groupby("Event")["Event"].count()
```



Starbucks Offers | Filtering + Grouping

What is the average transaction amount per offer type?

Mean Revenue per Transaction

1. Filter `Event == "transaction"` *(exclude zero-revenue “Offer” rows)*

```python
# Filter for transactions only
transactions_only = data[data["Event"] == "transaction"]
```

2. Group by Offer ID

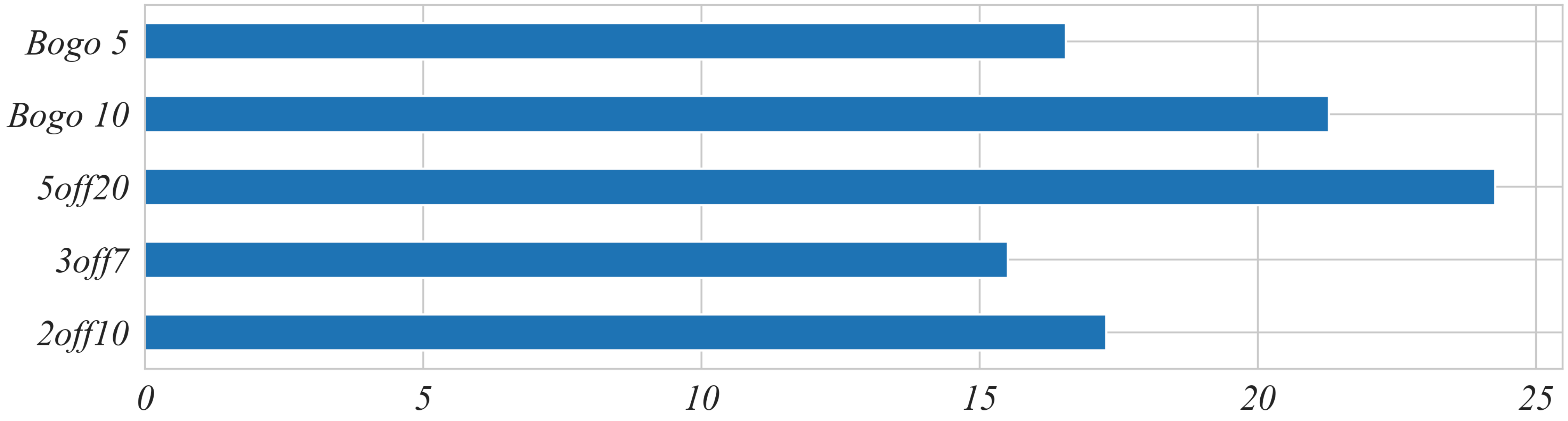
```python
# Group by Offer ID
transaction_groups = transactions_only.groupby("Offer ID")
```

3. Take the mean of the revenue column

```python
# Take the mean revenue
mean_revenue = transaction_groups["Revenue"].mean()
```

Starbucks Offers | Filtering + Grouping

What is the average transaction amount per offer type?



> this often gives a better picture of how much people spend per transaction when they use the offer

Starbucks Offers | Drawing Conclusions

Which offers are truly most effective?

1. How many times was each offer sent?

```python
# Count offers by Offer ID
offers_only = data[data["Event"] == "offer"] # Filter for Offer
offers_count = offers_only.groupby("Offer ID")["Event"].count()
```

2. How many times was each offer actually used?

```python
# Count transactions by Offer ID
transactions_only = data[data["Event"] == "transaction"] # Filter for Transaction
transactions_count = transactions_only.groupby("Offer ID")["Event"].count()
```

3. Total revenue or average revenue from those used offers.

```python
# Sum revenue by Offer ID
grouped_revenue = data.groupby("Offer ID")["Revenue"].sum()
```

Starbucks Offers | Combining Results

Which offers are truly most effective?

Combine into a single data frame:

- *`offers_count`*
- *`transactions_count`*
- *`grouped_revenue`*

```
summary = pd.DataFrame({
    "Offers": offers_count,
    "Transactions": transactions_count,
    "Revenue": grouped_revenue
})
```

Create new columns

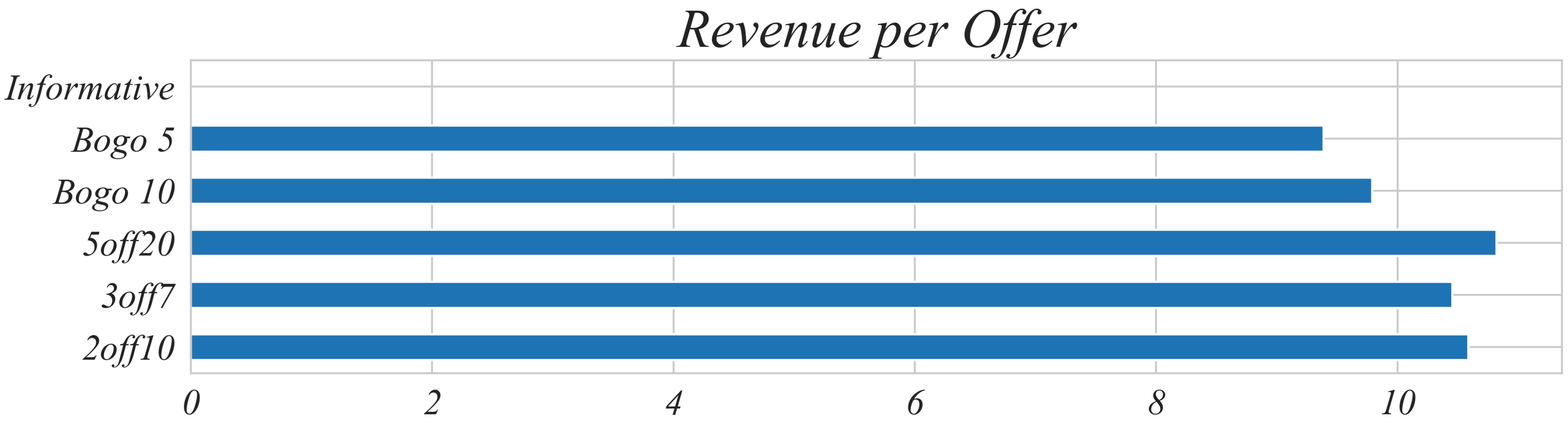
```
# Create a "Revenue per Offer" column
summary["Revenue_per_Offer"] = summary["Revenue"] / summary["Offers"]

# Create a "Transactions per Offer" column
summary["Transactions_per_Offer"] = summary["Transactions"] / summary["Offers"]
```

Starbucks Offers | Revenue Figure

Which offers are truly most effective?

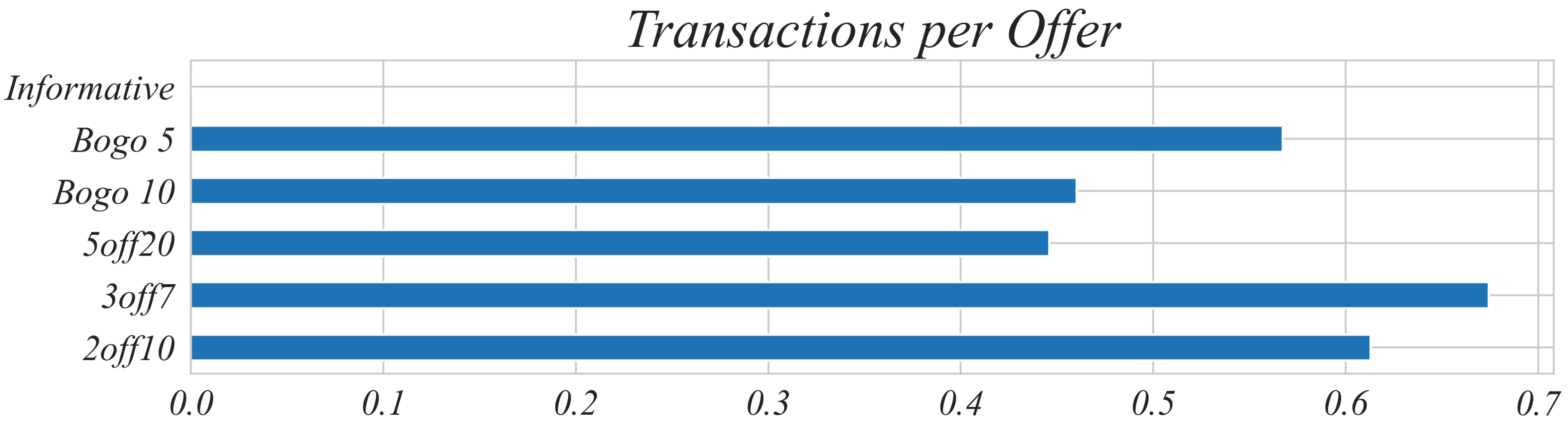
```
# Plot revenue per offer
summary_df["Revenue_per_Offer"].plot(kind='barh', title="Revenue per Offer")
```



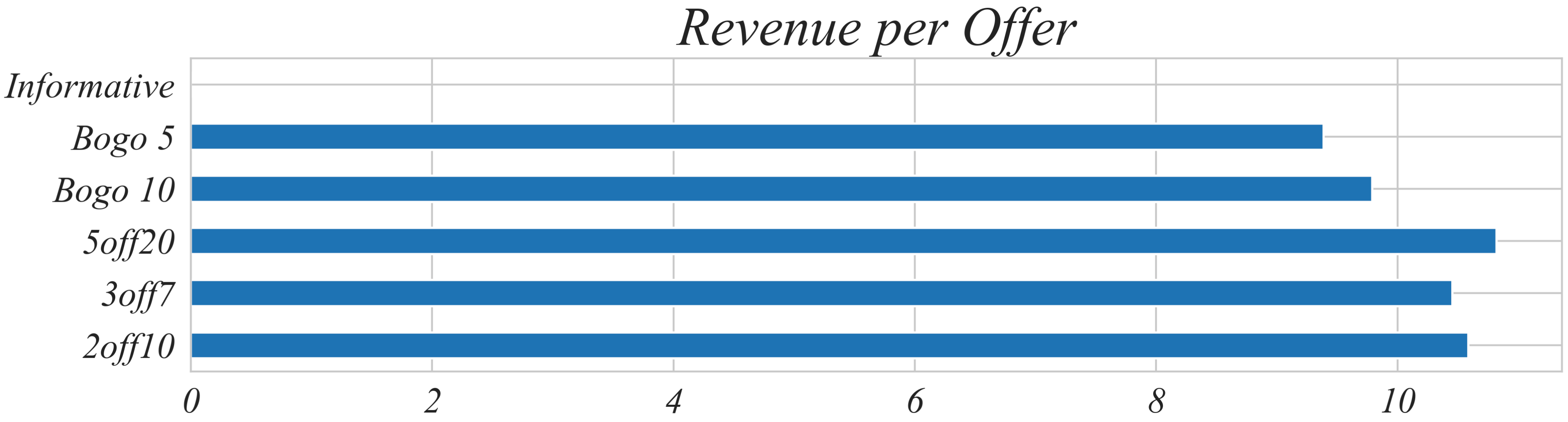
Starbucks Offers | Transaction Figure

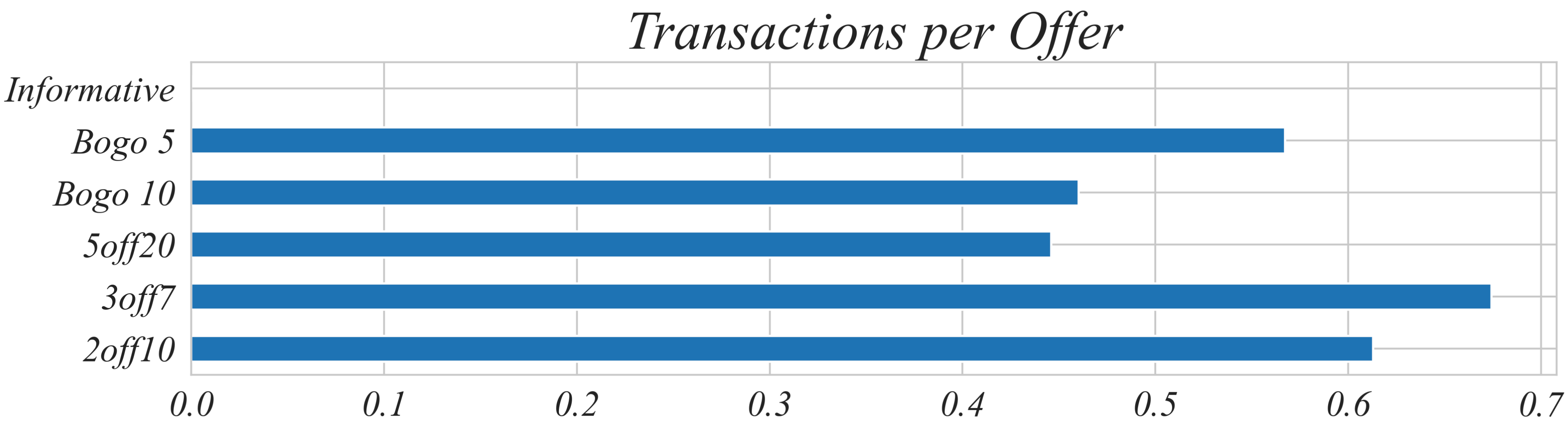
Which offers are truly most effective?

```
# Plot transactions per offer
summary_df["Transactions_per_Offer"].plot(kind='barh', title="Transactions per Offer")
```



Starbucks Offers | Both Figures




Transactions per Offer
Informative
Bogo 5
Bogo 10
5off20
3off7
2off10
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7

Starbucks Offers | Interpretation

Which offers are most effective?

- *The offer 5off20 has the highest* ***revenue*** *but a lower* ***redemption rate****.*
- *The offer 3off7 has a high* ***redemption rate*** *but the discount may be costly to Starbucks.*
- *The offer 2off10 lands solidly in the top on both metrics and represents a more modest discount.*

Part 1.6 | Summary

- ***Group and Aggregate**: Group by relevant columns to quickly summarize data*
- ***Filtering Matters**: Filter out irrelevant rows before grouping*
- ***Common Aggregations**: Use summaries like **sum**, **count**, **mean**, or **max***
- ***Widespread Use**: This technique is core to data analysis in nearly every field*
- ***Next Steps**: Combine grouping and filtering with joins, pivots, or merges for even richer analysis and visualization.*