```
%%writefile posts.csv
text
"Men's Wearhouse had great suit prices, but the tailoring took longer than prom
"Bought a tux at Men's Wearhouse for a wedding and the staff was extremely help
"Men's Wearhouse suits are affordable, but customer service really depends on t
"Had to come back twice because my suit wasn't altered correctly at Men's Wearh
"Really impressed with the selection at Men's Wearhouse—easy to find something
"Men's Wearhouse employee went above and beyond to help me last minute."
"Suit looked good but the pickup process at Men's Wearhouse was frustrating."
"Men's Wearhouse is my go-to for work suits—consistent quality for the price."
"Customer service at Men's Wearhouse felt rushed and inattentive."
"Great deals at Men's Wearhouse, especially during seasonal sales."
"My tailoring appointment at Men's Wearhouse ran late, but the final fit was sc
"Men's Wearhouse staff seemed understaffed when I visited."
"Happy with my purchase from Men's Wearhouse—good value overall."
"Not impressed with Men's Wearhouse alterations; had to get them redone elsewhe
"Men's Wearhouse made renting a suit for prom really easy."
"The fitting experience at Men's Wearhouse was quick and professional."
"Prices were fair at Men's Wearhouse, but checkout took forever."
"Men's Wearhouse quality is decent, though not luxury."
"Had a positive experience with Men's Wearhouse staff helping me choose a blaze
"Men's Wearhouse needs better communication on tailoring timelines."
```

```
Overwriting posts.csv
```

```
import pandas as pd
pd.read_csv("posts.csv").head()
```

	text
0	Men's Wearhouse had great suit prices, but the...
1	Bought a tux at Men's Wearhouse for a wedding ...
2	Men's Wearhouse suits are affordable, but cust...
3	Had to come back twice because my suit wasn't ...
4	Really impressed with the selection at Men's W...

```
df = pd.read_csv("posts.csv")
df.shape
```

```
(20, 1)
```

```
# Simple keyword-based theme tagging
def assign_theme(text):
    text = text.lower()
    if any(word in text for word in ['tailor', 'alter', 'fitting']):
        return 'tailoring'
    elif any(word in text for word in ['staff', 'service', 'associate']):
        return 'customer_service'
    elif any(word in text for word in ['price', 'deal', 'affordable', 'value']):
        return 'price_value'
    elif any(word in text for word in ['quality', 'luxury']):
        return 'product_quality'
    else:
        return 'store_experience'

df['theme'] = df['text'].apply(assign_theme)

df.head()
```

	text	theme
0	Men's Wearhouse had great suit prices, but the...	tailoring
1	Bought a tux at Men's Wearhouse for a wedding ...	customer_service
2	Men's Wearhouse suits are affordable, but cust...	customer_service
3	Had to come back twice because my suit wasn't ...	tailoring
4	Really impressed with the selection at Men's W...	store_experience

Next steps: Generate code with df | New interactive sheet

```
df.shape
```

```
(20, 2)
```

```
df.columns
```

```
Index(['text', 'theme'], dtype='object')
```

```
from textblob import TextBlob

# Create polarity if it's missing
if 'polarity' not in df.columns:
    df['polarity'] = df['text'].apply(lambda x: TextBlob(str(x)).sentiment.pola

# Now compute mean polarity by theme
```

```
df.groupby('theme')['polarity'].mean().sort_values()
```

theme	polarity
customer_service	0.060455
product_quality	0.166667
tailoring	0.189352
price_value	0.375000
store_experience	0.395833

dtype: float64

```
def assign_theme(text):
    text = str(text).lower()
    if any(word in text for word in ['tailor', 'alter', 'fitting']):
        return 'tailoring'
    elif any(word in text for word in ['staff', 'service', 'associate']):
        return 'customer_service'
    elif any(word in text for word in ['price', 'deal', 'affordable', 'value']):
        return 'price_value'
    elif any(word in text for word in ['quality', 'luxury']):
        return 'product_quality'
    else:
        return 'store_experience'

df['theme'] = df['text'].apply(assign_theme)
```

```
df.groupby('theme')['polarity'].mean().sort_values()
```

theme	polarity
customer_service	0.060455
product_quality	0.166667
tailoring	0.189352
price_value	0.375000
store_experience	0.395833

dtype: float64

```
df.groupby('theme')['polarity'].mean().sort_values().plot(kind='barh', title='Av
```

```
<Axes: title={'center': 'Average Sentiment by Theme'}, ylabel='theme'>
```