Introduction to common marketing metrics

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Jill Rosok
Data Scientist

datacamp

Was the campaign successful?

Common metrics:

- Conversion rate
- Retention rate

Conversion rate

$$\text{Conversion rate} = \frac{\text{Number of people who convert}}{\text{Total number of people we marketed to}}$$

Calculating conversion rate using pandas

```
subscribers = marketing[marketing['converted'] == True]\
                       ['user_id'].nunique()
total = marketing['user_id'].nunique()
conv_rate = subscribers/total

print(round(conv_rate*100, 2), '%')
```

```
13.89 %
```

Retention rate

$$\text{Retention rate} = \frac{\text{Number of people who remain subscribed}}{\text{Total number of people who converted}}$$

Calculating retention rate

```
retained = marketing[marketing['is_retained'] == True]\
                    ['user_id'].nunique()
subscribers = marketing[marketing['converted'] == True]\
                     ['user_id'].nunique()
retention = retained/subscribers

print(round(retention*100, 2), '%')
```

```
84%
```

Let's practice!

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Customer segmentation

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Jill Rosok
Data Scientist

datacamp

Common ways to segment audiences

- Age
- Gender
- Location
- Past interaction(s) with the business
- Marketing channels users interacted with

Segmenting using pandas

```
# Subset to include only House Ads
house_ads = marketing\
          [marketing['subscribing_channel'] == 'House Ads']
retained = house_ads[house_ads['is_retained'] == True]\
                    ['user_id'].nunique()
subscribers = house_ads[house_ads['converted'] == True]\
                     ['user_id'].nunique()

retention_rate = retained/subscribers
print(round(retention_rate*100,2), '%')
```

```
58.05 %
```

There must be an easier way to segment!

Segmenting using pandas - groupby()

```
# Group by subscribing_channel and calculate retention
retained = marketing[marketing['is_retained'] == True]\
           .groupby(['subscribing_channel'])\
           ['user_id'].nunique()
print(retained)
```

```
subscribing_channel
Email         109
Facebook      152
House Ads     173
Instagram     158
Push           54
Name: user_id, dtype: int64
```

Segmenting using pandas - groupby()

```
# Group by subscribing_channel and calculate subscribers
subscribers = marketing[marketing['converted'] == True]\
            .groupby(['subscribing_channel'])\
            ['user_id'].nunique()
print(subscribers)
```

```
subscribing_channel
Email         125
Facebook      221
House Ads     298
Instagram     232
Push           77
Name: user_id, dtype: int64
```

Segmenting results

```
# Calculate the retention rate across the DataFrame
channel_retention_rate = (retained/subscribers)*100
print(channel_retention_rate)
```

```
subscribing_channel
Email         87.200000
Facebook      68.778281
House Ads     58.053691
Instagram     68.103448
Push          70.129870
Name: user_id, dtype: float64
```

Let's practice!

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Plotting campaign results (I)

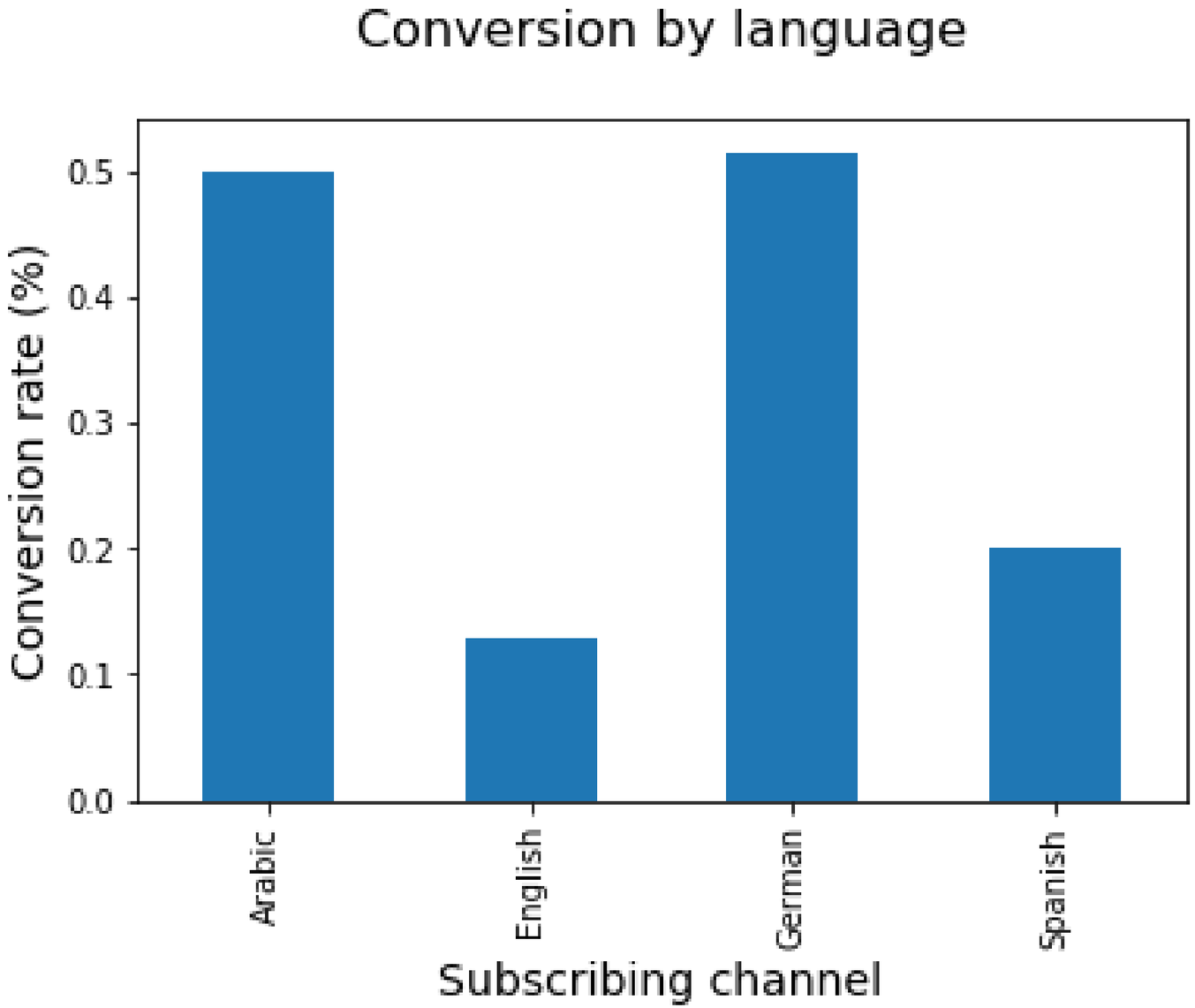
ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Jill Rosok
Data Scientist

datacamp

Comparing language conversion rates

```
import matplotlib.pyplot as plt


# Create a bar chart using channel retention DataFrame
language_conversion_rate.plot(kind = 'bar')
# Add a title and x and y-axis labels
plt.title('Conversion rate by language\n', size = 16)
plt.xlabel('Language', size = 14)
plt.ylabel('Conversion rate (%)', size = 14)
# Display the plot
plt.show()
```


Conversion by language
Conversion rate (%)
0.0
0.1
0.2
0.3
0.4
0.5
Arabic
English
German
Spanish
Subscribing channel

Calculating subscriber quality

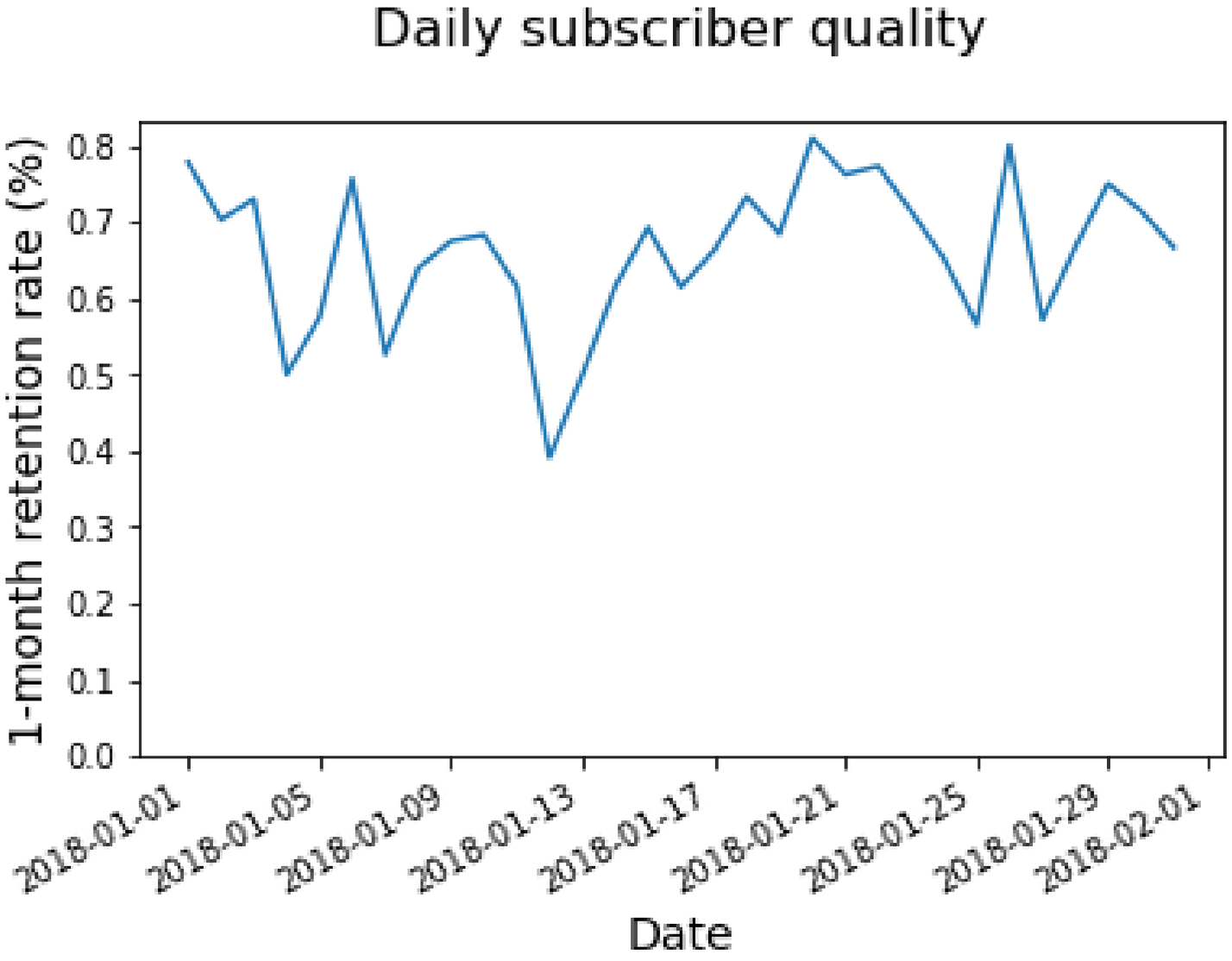
```
# Group by language_displayed and count unique users
total = marketing.groupby(['date_subscribed'])['user_id']\
                      .nunique()

# Group by language_displayed and sum conversions
retained = marketing[marketing['is_retained'] == True]\
                          .groupby(['date_subscribed'])\
                          ['user_id'].nunique()

# Calculate subscriber quality across dates
daily_retention_rate = retained/total
```

Preparing data to be plotted over time

```
# Reset index to turn the Series into a DataFrame
daily_retention_rate =
    pd.DataFrame(daily_retention_rate.reset_index())

# Rename columns
daily_retention_rate.columns = ['date_subscribed',
                                'retention_rate']
```

Visualizing data trended over time

```
# Create a line chart using the daily_retention DataFrame
daily_retention_rate.plot('date_subscribed',
                          'retention_rate')

# Add a title and x and y-axis labels
plt.title('Daily subscriber quality\n', size = 16)
plt.ylabel('1-month retention rate (%)', size = 14)
plt.xlabel('Date', size = 14)

# Set the y-axis to begin at 0
plt.ylim(0)
# Display the plot
plt.show()
```


Daily subscriber quality
1-month retention rate (%)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2018-01-01
2018-01-05
2018-01-09
2018-01-13
2018-01-17
2018-01-21
2018-01-25
2018-01-29
2018-02-01
Date

Let's practice!

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Plotting campaign results (II)

ANALYZING MARKETING CAMPAIGNS WITH PANDAS

Jill Rosok
Data Scientist

datacamp

Grouping by multiple columns

```
language = marketing.groupby(['date_served',
                              'language_preferred'])\
                              ['user_id'].count()


print(language.head())
```

```
date_served  preferred_language
2018-01-01   Arabic                3
             English             351
             German                5
             Spanish              11
2018-01-02   Arabic                4
Name: user_id, dtype: int64
```

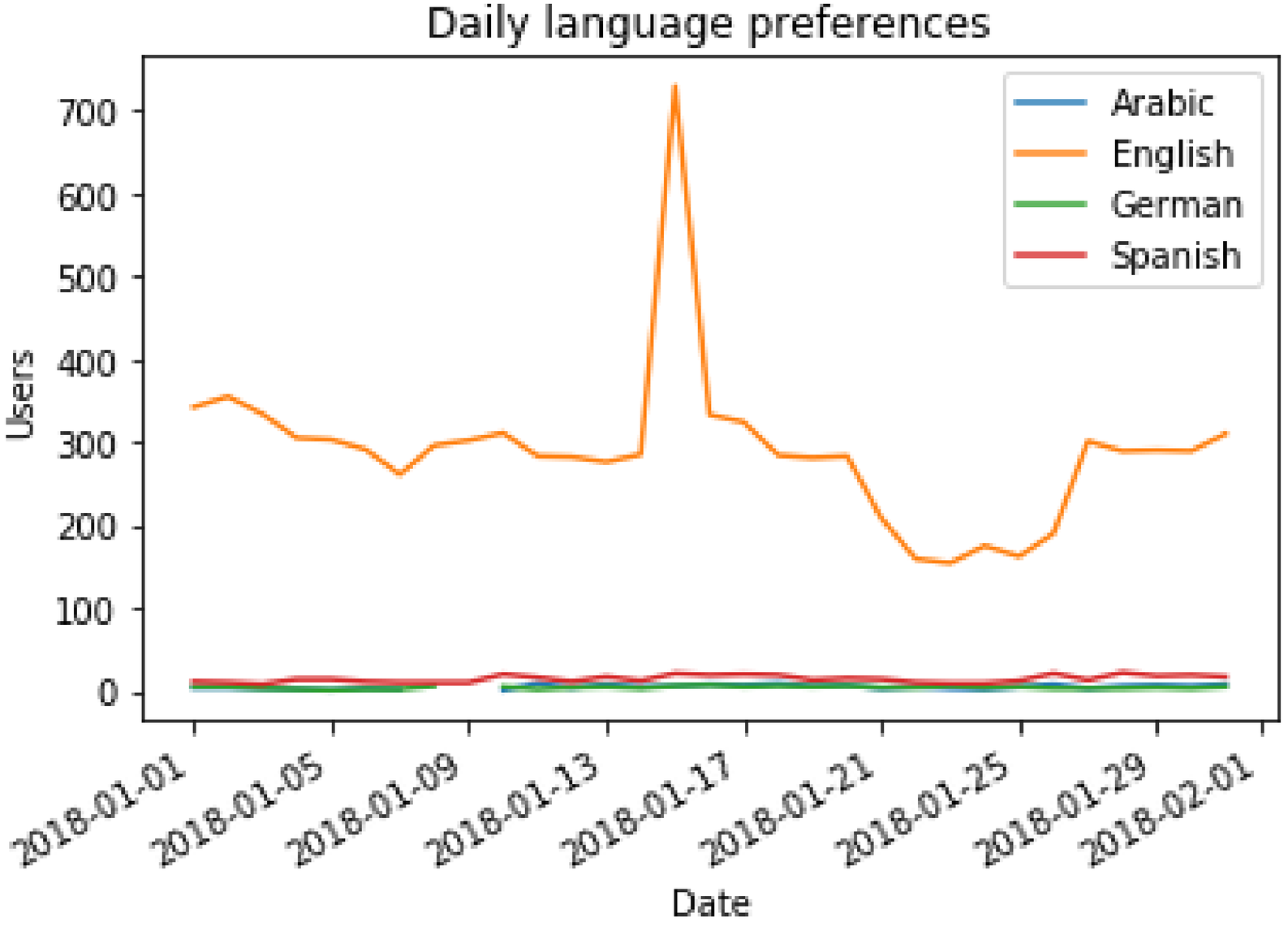
Unstacking after groupby

```
language = pd.DataFrame(language.unstack(level=1))

print(language.head())
```

```
preferred_language    Arabic    English    German    Spanish
date_served
2018-01-01            3.0        351.0     5.0       11.0
2018-01-02            4.0        369.0     6.0       10.0
2018-01-03            3.0        349.0     3.0       8.0
2018-01-04            2.0        313.0     2.0       14.0
2018-01-05            NaN        310.0     1.0       14.0
```

Plotting preferred language over time

```
language.plot()
plt.title('Daily language preferences')
plt.xlabel('Date')
plt.ylabel('Users')
plt.legend(loc = 'upper right',
           labels = language.columns.values)
plt.show()
```


Daily language preferences
Arabic
English
German
Spanish
Users
700
600
500
400
300
200
100
0
2018-01-01
2018-01-05
2018-01-09
2018-01-13
2018-01-17
2018-01-21
2018-01-25
2018-01-29
2018-02-01
Date

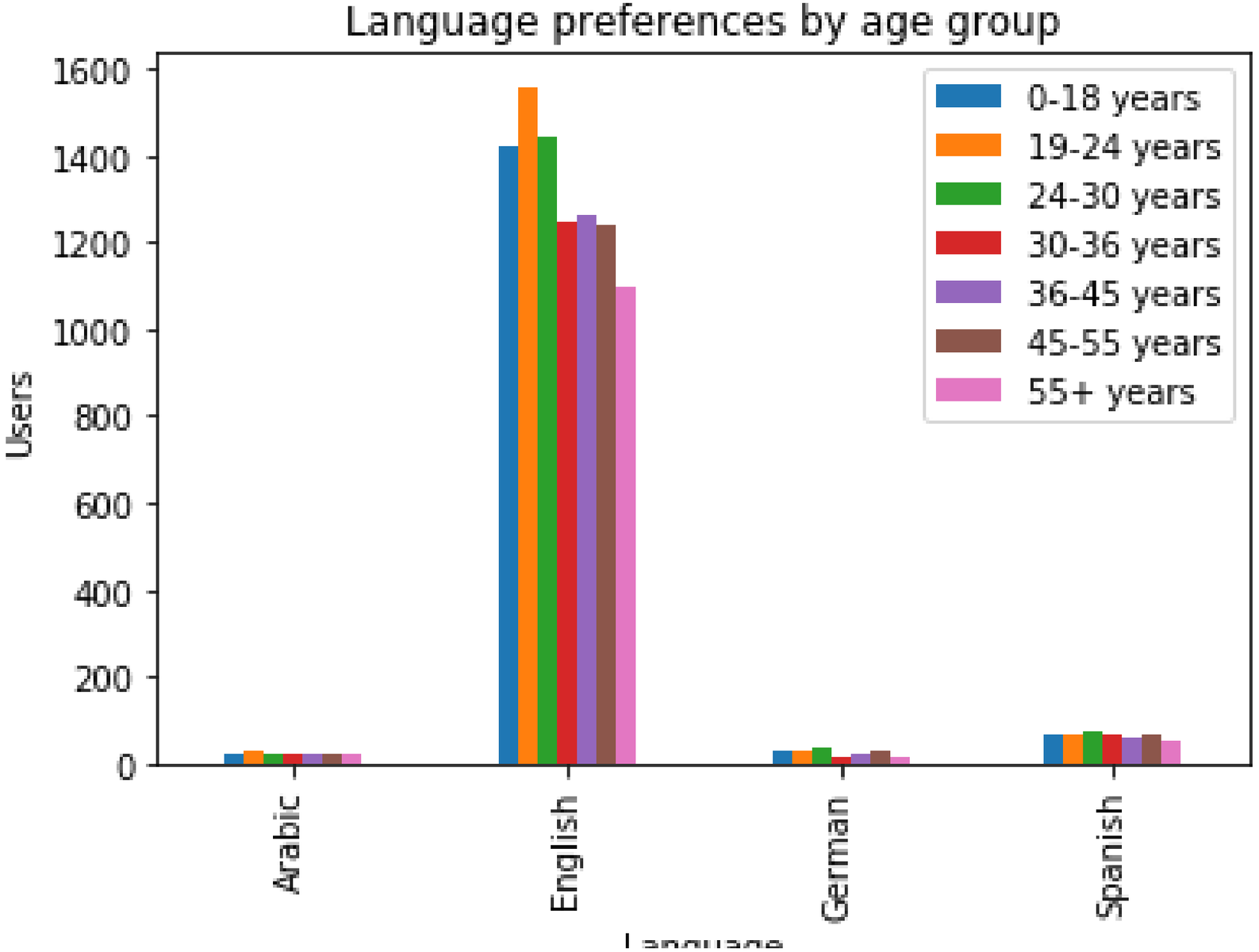
Creating grouped bar charts

```
# Create DataFrame grouped by age and language preference
language_age = marketing.groupby(['language_preferred',
                                  'age_group'])\
                                 ['user_id'].count()
language_age = pd.DataFrame(language_age.unstack(level=1))
print(language_age.head())
```

```
preferred_language  Arabic  English  German  Spanish
age_group
0-18 years              17     1409      20       66
19-24 years             25     1539      20       66
24-30 years             18     1424      18       71
30-36 years             19     1238      14       69
36-45 years             18     1251      17       55
```

Plotting language preferences by age group

```
language_age.plot(kind='bar')
plt.title('Language preferences by age group')
plt.xlabel('Language')
plt.ylabel('Users')
plt.legend(loc = 'upper right',
           labels = language_age.columns.values)
plt.show()
```


Language preferences by age group
1600
1400
1200
1000
800
600
400
200
0
Users
0-18 years
19-24 years
24-30 years
30-36 years
36-45 years
45-55 years
55+ years
Arabic
English
German
Spanish

Let's practice!

ANALYZING MARKETING CAMPAIGNS WITH PANDAS