eda_previo

June 19, 2024

[19]:
```
import mplfinance as mpf
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1 Hagamos una visualizacion financiera de la empresa Starbucks

1.0.1 Veremos a traves de un grafico de velas, la informacion referente al cierre en Nasdaq , desde el 01-01-2016 al 01-01-2024

Debido a que Dunkin' Brands Group, Inc. (conocida por sus marcas Dunkin' Donuts y Baskin-Robbins) dejó de cotizar en el NASDAQ el 15 de diciembre de 2020, la data anterior a la fecha no existe en canales oficiales o regulares de Nasdaq

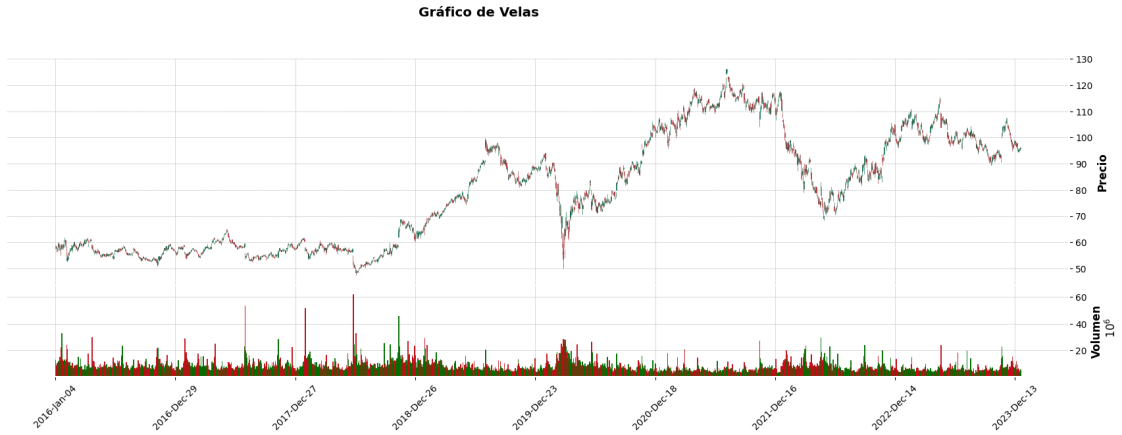
[3]:
```
# Carga el archivo .parquet en un DataFrame de Pandas
df_nasdaq_starbucks = pd.read_parquet('..//data//starbucks_nasdaq_data.parquet')
```

[ ]:
```
# Crear el gráfico de velas
mpf.plot(df_nasdaq_starbucks, type='candle', style='charles', volume=True,␣
 ↪title='Gráfico de Velas', ylabel='Precio', ylabel_lower='Volumen')
```

[6]:
```
# Ajustar el tamaño del gráfico
figratio = (26, 8)  # Proporción de la figura (ancho, alto)
figscale = 1.2  # Escala del tamaño general de la figura

mpf.plot(df_nasdaq_starbucks,
         type='candle',
         style='charles',
         volume=True,
         title='Gráfico de Velas',
         ylabel='Precio',
         ylabel_lower='Volumen',
         figratio=figratio,
         figscale=figscale)
```

```
c:\Users\jhcat\AppData\Local\Programs\Python\Python312\Lib\site-
packages\mplfinance\_arg_validators.py:84: UserWarning:

 ================================================================
```

```
   WARNING: YOU ARE PLOTTING SO MUCH DATA THAT IT MAY NOT BE
            POSSIBLE TO SEE DETAILS (Candles, Ohlc-Bars, Etc.)
   For more information see:
   - https://github.com/matplotlib/mplfinance/wiki/Plotting-Too-Much-Data

   TO SILENCE THIS WARNING, set `type='line'` in `mpf.plot()`
   OR set kwarg `warn_too_much_data=N` where N is an integer
   LARGER than the number of data points you want to plot.

 ================================================================
  warnings.warn('\n\n
================================================================ '+
```



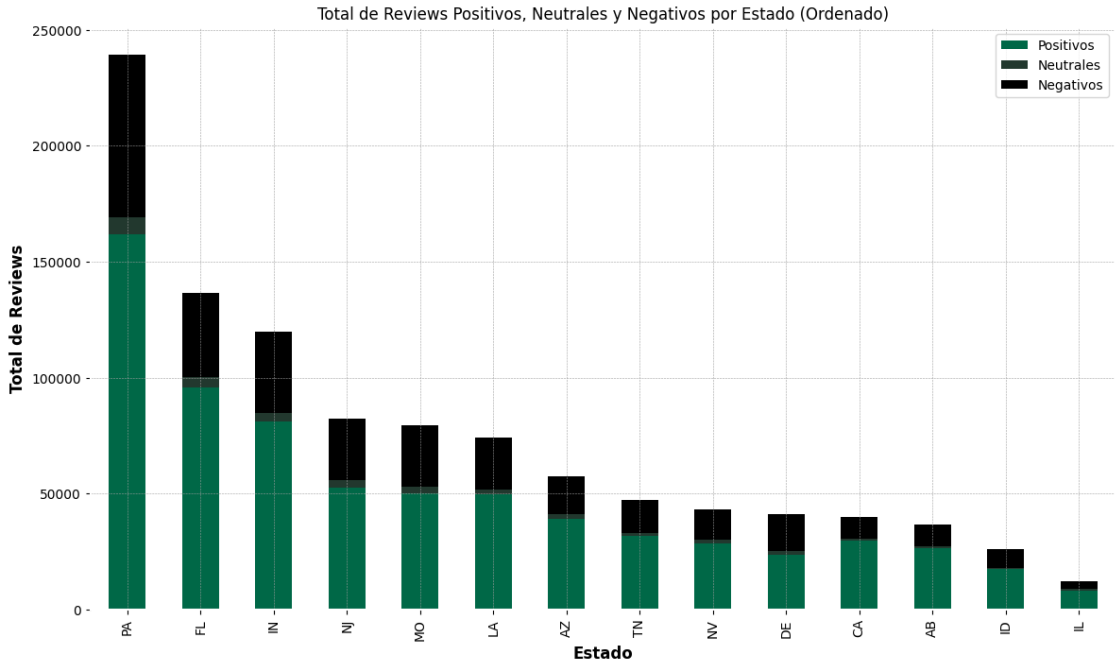
2 Hagamos una comparativa de Numero de Reviews x Estado, tanto para Starbucks como para su competidor (Dunkin), considerando discriminar por tipo de Review

2.0.1 Reviews de Starbucks por Estado

[26]:
```
# Cargar los archivos Parquet
df_business = pd.read_parquet('..//data//business.parquet')
df_review_sent_total = pd.read_parquet('..//data//review_sent_total.parquet')

# Unir los DataFrames en la columna business_id
df_merged = pd.merge(df_business, df_review_sent_total, on='business_id')

# Agrupar por 'state' y sumar las columnas 'positive_total', 'neutral_total',␣
 ↪'negative_total'
df_grouped = df_merged.groupby('state')[['positive_total', 'neutral_total',␣
 ↪'negative_total']].sum().reset_index()
```

```
# Calcular la columna 'total_reviews' como la suma de 'positive_total',␣
 ↪'neutral_total' y 'negative_total'
df_grouped['total_reviews'] = df_grouped['positive_total'] +␣
 ↪df_grouped['neutral_total'] + df_grouped['negative_total']

# Ordenar el DataFrame por 'total_reviews' de mayor a menor
df_grouped = df_grouped.sort_values(by='total_reviews', ascending=False)

# Definir los colores para las columnas
colors = ['#006847', '#22382E', '#000000']
```

[27]:
```
# Crear el gráfico de columnas apiladas
fig, ax = plt.subplots(figsize=(14, 8))

# Crear las columnas apiladas ordenadas
df_grouped.set_index('state')[['positive_total', 'neutral_total',␣
 ↪'negative_total']].plot(kind='bar', stacked=True, color=colors, ax=ax)

# Personalizar el gráfico
plt.title('Total de Reviews Positivos, Neutrales y Negativos por Estado␣
 ↪(Ordenado)')
plt.xlabel('Estado')
plt.ylabel('Total de Reviews')
plt.legend(['Positivos', 'Neutrales', 'Negativos'])

# Mostrar el gráfico
plt.show()
```

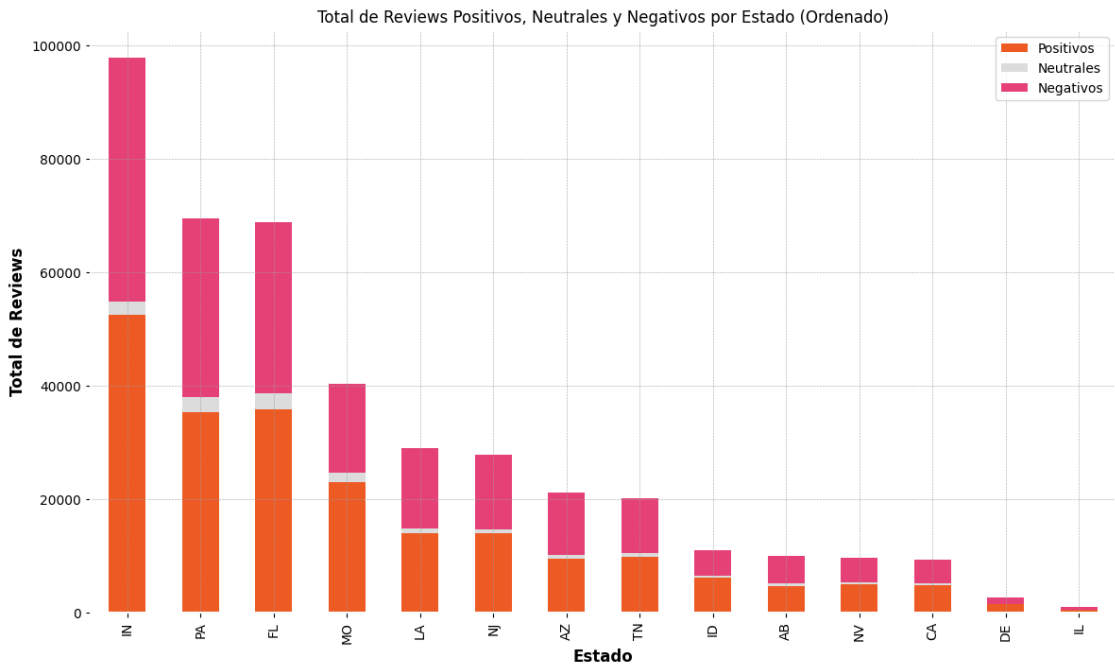


2.0.2 Reviews de Dunkin por Estado

[17]:
```
# Cargar los archivos Parquet
df_business_dunkin = pd.read_parquet('..//data//business_dunkin.parquet')
df_review_dunkin__sent_total = pd.read_parquet('..//data//
 ↪review_dunkin_sent_total.parquet')

# Unir los DataFrames en la columna business_id
df_merged = pd.merge(df_business_dunkin, df_review_dunkin__sent_total,␣
 ↪on='business_id')

# Agrupar por 'state' y sumar las columnas 'positive_total', 'neutral_total',␣
 ↪'negative_total'
df_grouped = df_merged.groupby('state')[['positive_total', 'neutral_total',␣
 ↪'negative_total']].sum().reset_index()

# Calcular la columna 'total_reviews' como la suma de 'positive_total',␣
 ↪'neutral_total' y 'negative_total'
df_grouped['total_reviews'] = df_grouped['positive_total'] +␣
 ↪df_grouped['neutral_total'] + df_grouped['negative_total']

# Ordenar el DataFrame por 'total_reviews' de mayor a menor
df_grouped = df_grouped.sort_values(by='total_reviews', ascending=False)

# Definir los colores para las columnas
```

```
colors = ['#ED5A24', '#DCDCDC', '#E64176']
```

[18]:
```
# Crear el gráfico de columnas apiladas
fig, ax = plt.subplots(figsize=(14, 8))

# Crear las columnas apiladas ordenadas
df_grouped.set_index('state')[['positive_total', 'neutral_total',␣
 ↪'negative_total']].plot(kind='bar', stacked=True, color=colors, ax=ax)

# Personalizar el gráfico
plt.title('Total de Reviews Positivos, Neutrales y Negativos por Estado␣
 ↪(Ordenado)')
plt.xlabel('Estado')
plt.ylabel('Total de Reviews')
plt.legend(['Positivos', 'Neutrales', 'Negativos'])

# Mostrar el gráfico
plt.show()
```

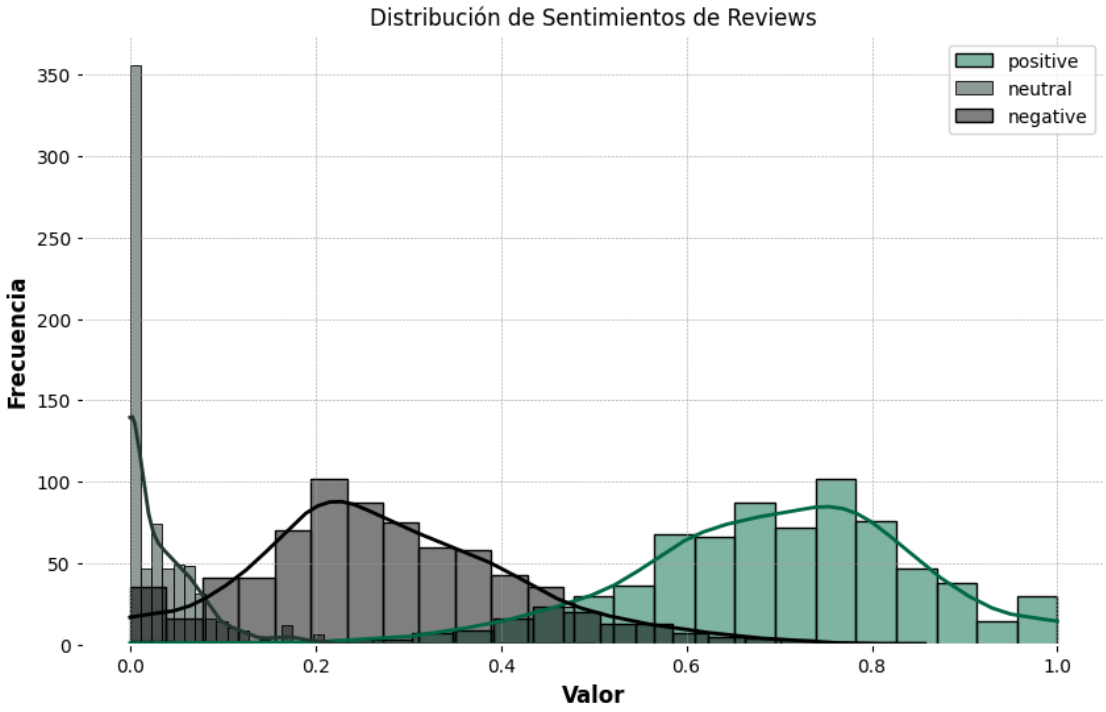


2.0.3 Visualizar Distribución de los Reviews para Starbucks

[28]:
```
# Cargar el archivo Parquet
df_review_sent_total = pd.read_parquet('..//data//review_sent_total.parquet')

# Definir los colores
```

```
colors = ['#006847', '#22382E', '#000000']
```

[29]:
```
# Crear el gráfico de distribución
plt.figure(figsize=(10, 6))
for i, col in enumerate(['positive', 'neutral', 'negative']):
    sns.histplot(df_review_sent_total[col], color=colors[i], label=col,␣
 ↪kde=True)

# Personalizar el gráfico
plt.title('Distribución de Sentimientos de Reviews')
plt.xlabel('Valor')
plt.ylabel('Frecuencia')
plt.legend()

# Mostrar el gráfico
plt.show()
```

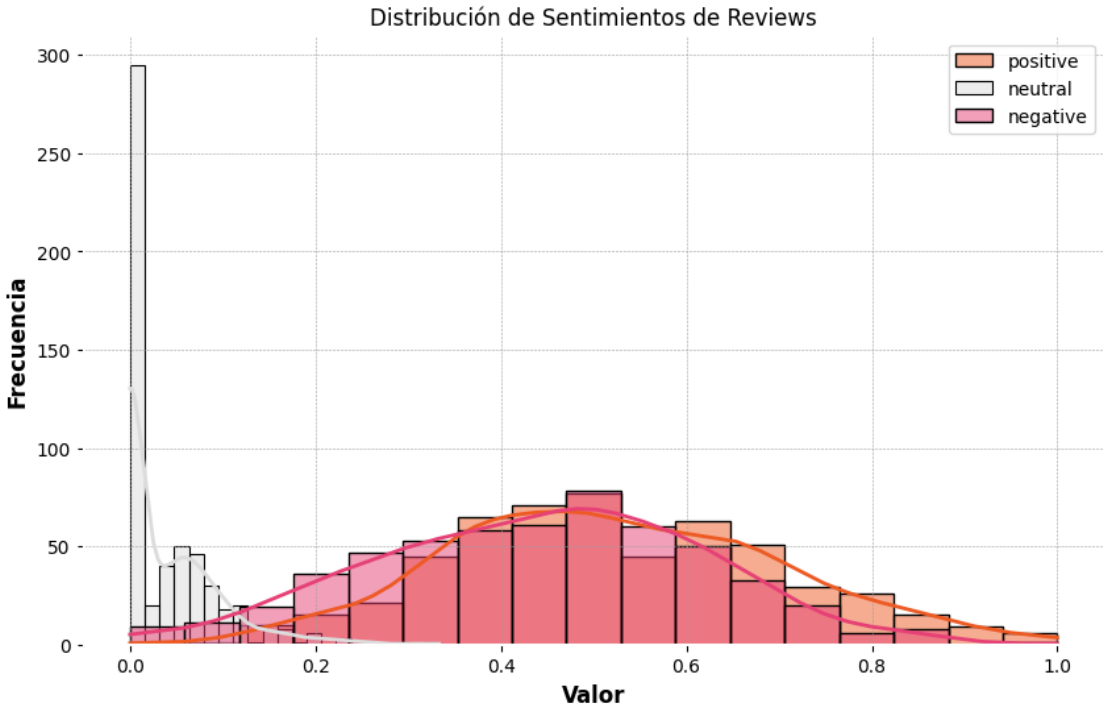


2.0.4 Visualizar Distribución de los Reviews para Dunkin

[30]:
```
# Cargar el archivo Parquet
df_review_sent_total = pd.read_parquet('..//data//review_dunkin_sent_total.
 ↪parquet')
```

```
# Definir los colores
colors = ['#ED5A24', '#DCDCDC', '#E64176']
```

[31]:
```
# Crear el gráfico de distribución
plt.figure(figsize=(10, 6))
for i, col in enumerate(['positive', 'neutral', 'negative']):
    sns.histplot(df_review_sent_total[col], color=colors[i], label=col,␣
 ↪kde=True)

# Personalizar el gráfico
plt.title('Distribución de Sentimientos de Reviews')
plt.xlabel('Valor')
plt.ylabel('Frecuencia')
plt.legend()

# Mostrar el gráfico
plt.show()
```



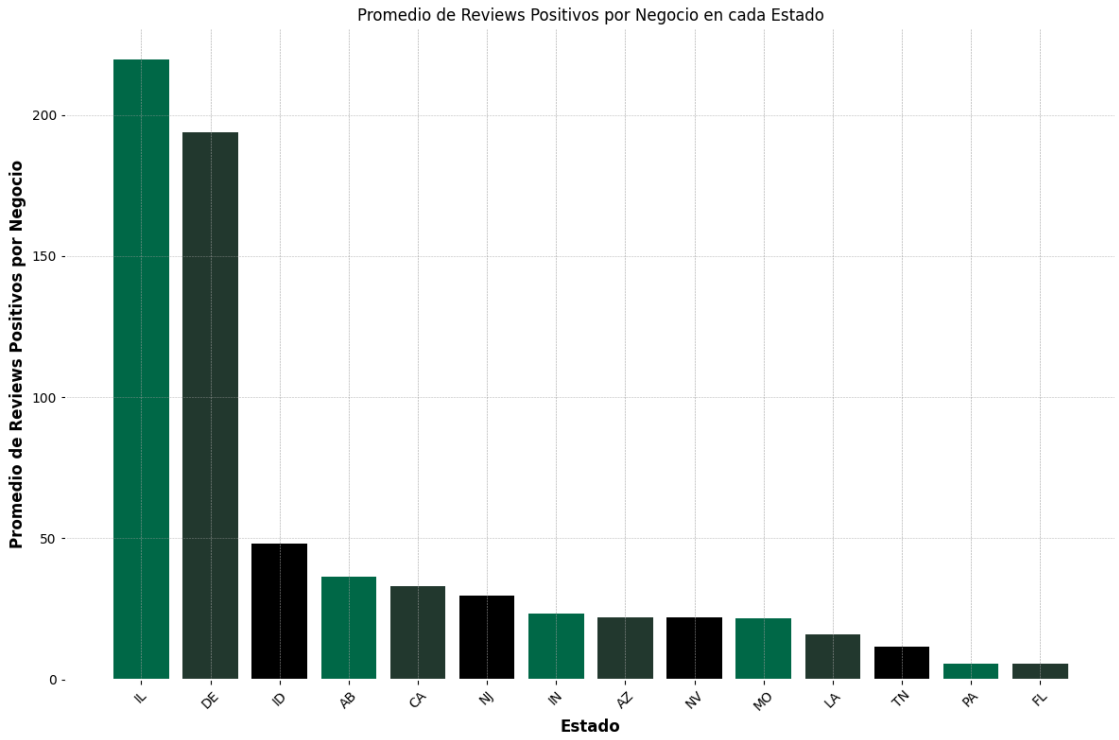
3 Definir la relación entre número de Reviews totales y la cantidad de locales por Estado

3.0.1 Los siguientes graficos nos representan, la cantidad de Reviews positivos de un Estado dividido entre el número total de locales en ese mismo Estado

Este grafico representa los Reviews Positivos Absolutos - Relación para Starbucks

[34]:
```
# Cargar los archivos Parquet
df_business = pd.read_parquet('..//data//business.parquet')
df_review_sent_total = pd.read_parquet('..//data//review_sent_total.parquet')

# Unir los DataFrames por la columna business_id
df_merged = pd.merge(df_business, df_review_sent_total, on='business_id')

# Calcular la cantidad promedio de reviews positivos por negocio en cada estado
df_merged['avg_positive_per_business'] = df_merged['positive_total'] /␣
 ↪df_merged.groupby('state')['business_id'].transform('nunique')

# Agrupar por estado y calcular el promedio de reviews positivos
df_state_avg_positive = df_merged.groupby('state')['avg_positive_per_business'].
 ↪mean().reset_index()

# Ordenar los estados de mayor a menor según el promedio de reviews positivos
df_state_avg_positive = df_state_avg_positive.
 ↪sort_values(by='avg_positive_per_business', ascending=False)

# Definir los colores para el gráfico
colors = ['#006847', '#22382E', '#000000']
```

[36]:
```
# Crear el gráfico de barras
plt.figure(figsize=(12, 8))
plt.bar(df_state_avg_positive['state'],␣
 ↪df_state_avg_positive['avg_positive_per_business'], color=colors)

# Personalizar el gráfico
plt.title('Promedio de Reviews Positivos por Negocio en cada Estado')
plt.xlabel('Estado')
plt.ylabel('Promedio de Reviews Positivos por Negocio')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

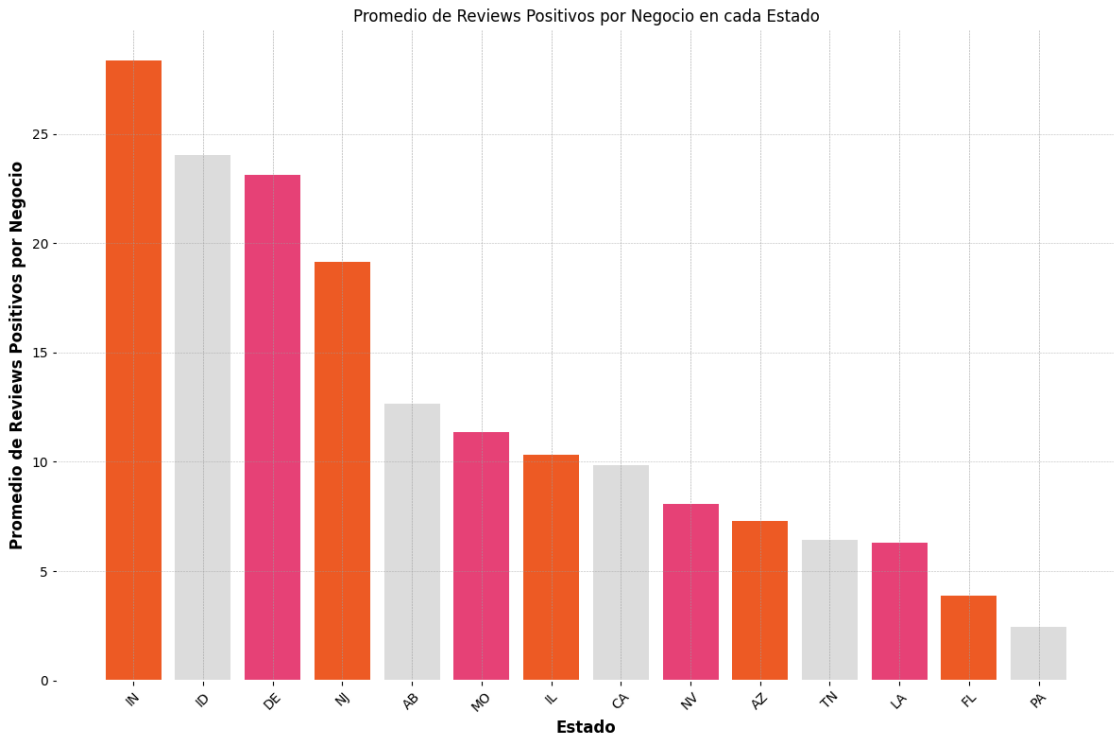


Este grafico representa los Reviews Positivos Absolutos - Relación para Dunkin

[37]:
```
# Cargar los archivos Parquet
df_business = pd.read_parquet('..//data//business_dunkin.parquet')
df_review_sent_total = pd.read_parquet('..//data//review_dunkin_sent_total.
 ↪parquet')

# Unir los DataFrames por la columna business_id
df_merged = pd.merge(df_business, df_review_sent_total, on='business_id')

# Calcular la cantidad promedio de reviews positivos por negocio en cada estado
df_merged['avg_positive_per_business'] = df_merged['positive_total'] /␣
 ↪df_merged.groupby('state')['business_id'].transform('nunique')

# Agrupar por estado y calcular el promedio de reviews positivos
df_state_avg_positive = df_merged.groupby('state')['avg_positive_per_business'].
 ↪mean().reset_index()

# Ordenar los estados de mayor a menor según el promedio de reviews positivos
df_state_avg_positive = df_state_avg_positive.
 ↪sort_values(by='avg_positive_per_business', ascending=False)

# Definir los colores para el gráfico
```

```
colors = ['#ED5A24', '#DCDCDC', '#E64176']
```

[38]:
```
# Crear el gráfico de barras
plt.figure(figsize=(12, 8))
plt.bar(df_state_avg_positive['state'],␣
 ↪df_state_avg_positive['avg_positive_per_business'], color=colors)

# Personalizar el gráfico
plt.title('Promedio de Reviews Positivos por Negocio en cada Estado')
plt.xlabel('Estado')
plt.ylabel('Promedio de Reviews Positivos por Negocio')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```

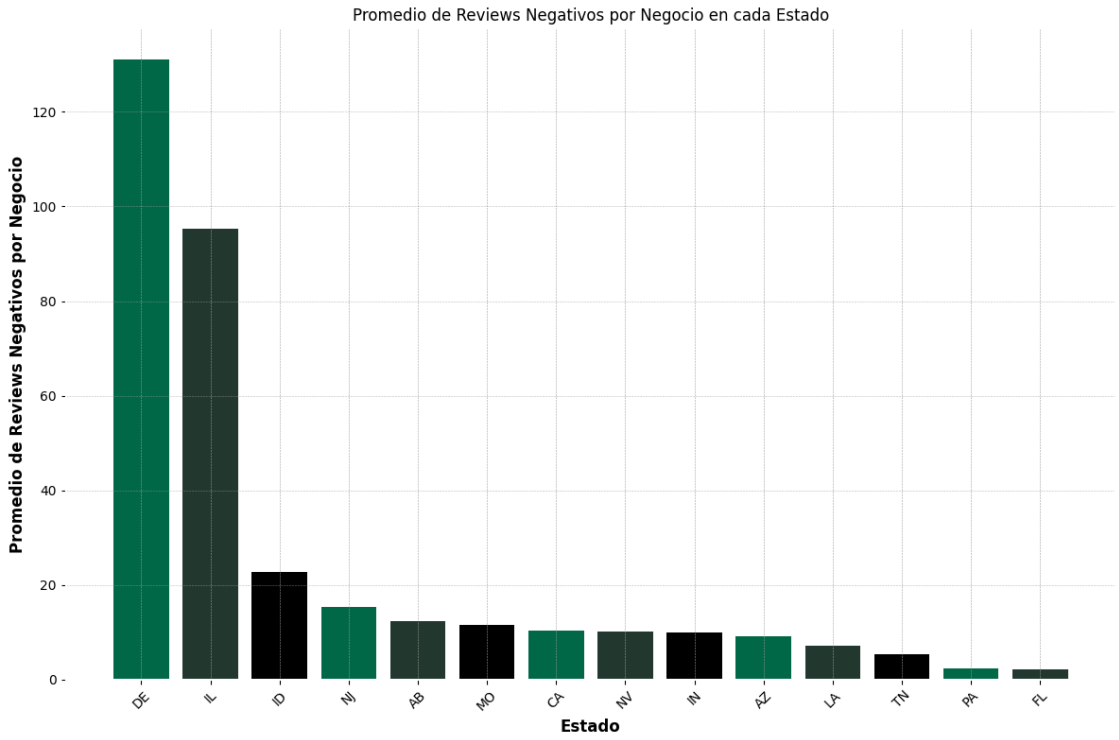


3.0.2 Los siguientes graficos nos representan, la cantidad de Reviews negativos de un Estado dividido entre el número total de locales en ese mismo Estado

Este grafico representa los Reviews Negativos Absolutos - Relación para Starbucks

[39]:
```
# Cargar los archivos Parquet
df_business = pd.read_parquet('..//data//business.parquet')
df_review_sent_total = pd.read_parquet('..//data//review_sent_total.parquet')

# Unir los DataFrames por la columna business_id
df_merged = pd.merge(df_business, df_review_sent_total, on='business_id')

# Calcular la cantidad promedio de reviews negativos por negocio en cada estado
df_merged['avg_negative_per_business'] = df_merged['negative_total'] /␣
 ↪df_merged.groupby('state')['business_id'].transform('nunique')

# Agrupar por estado y calcular el promedio de reviews negativos
df_state_avg_negative = df_merged.groupby('state')['avg_negative_per_business'].
 ↪mean().reset_index()

# Ordenar los estados de mayor a menor según el promedio de reviews negativos
df_state_avg_negative = df_state_avg_negative.
 ↪sort_values(by='avg_negative_per_business', ascending=False)

# Definir los colores para el gráfico
colors = ['#006847', '#22382E', '#000000']
```

[40]:
```
# Crear el gráfico de barras
plt.figure(figsize=(12, 8))
plt.bar(df_state_avg_negative['state'],␣
 ↪df_state_avg_negative['avg_negative_per_business'], color=colors)

# Personalizar el gráfico
plt.title('Promedio de Reviews Negativos por Negocio en cada Estado')
plt.xlabel('Estado')
plt.ylabel('Promedio de Reviews Negativos por Negocio')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```



Este grafico representa los Reviews Negativos Absolutos - Relación para Dunkin

[41]:
```
# Cargar los archivos Parquet
df_business = pd.read_parquet('..//data//business_dunkin.parquet')
df_review_sent_total = pd.read_parquet('..//data//review_dunkin_sent_total.
 ↪parquet')

# Unir los DataFrames por la columna business_id
df_merged = pd.merge(df_business, df_review_sent_total, on='business_id')

# Calcular la cantidad promedio de reviews negativos por negocio en cada estado
df_merged['avg_negative_per_business'] = df_merged['negative_total'] /␣
 ↪df_merged.groupby('state')['business_id'].transform('nunique')

# Agrupar por estado y calcular el promedio de reviews negativos
df_state_avg_negative = df_merged.groupby('state')['avg_negative_per_business'].
 ↪mean().reset_index()

# Ordenar los estados de mayor a menor según el promedio de reviews negativos
df_state_avg_negative = df_state_avg_negative.
 ↪sort_values(by='avg_negative_per_business', ascending=False)

# Definir los colores para el gráfico
```

```
colors = ['#ED5A24', '#DCDCDC', '#E64176']
```

[42]:
```
# Crear el gráfico de barras
plt.figure(figsize=(12, 8))
plt.bar(df_state_avg_negative['state'],␣
  ↪df_state_avg_negative['avg_negative_per_business'], color=colors)

# Personalizar el gráfico
plt.title('Promedio de Reviews Negativos por Negocio en cada Estado')
plt.xlabel('Estado')
plt.ylabel('Promedio de Reviews Negativos por Negocio')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Mostrar el gráfico
plt.tight_layout()
plt.show()
```