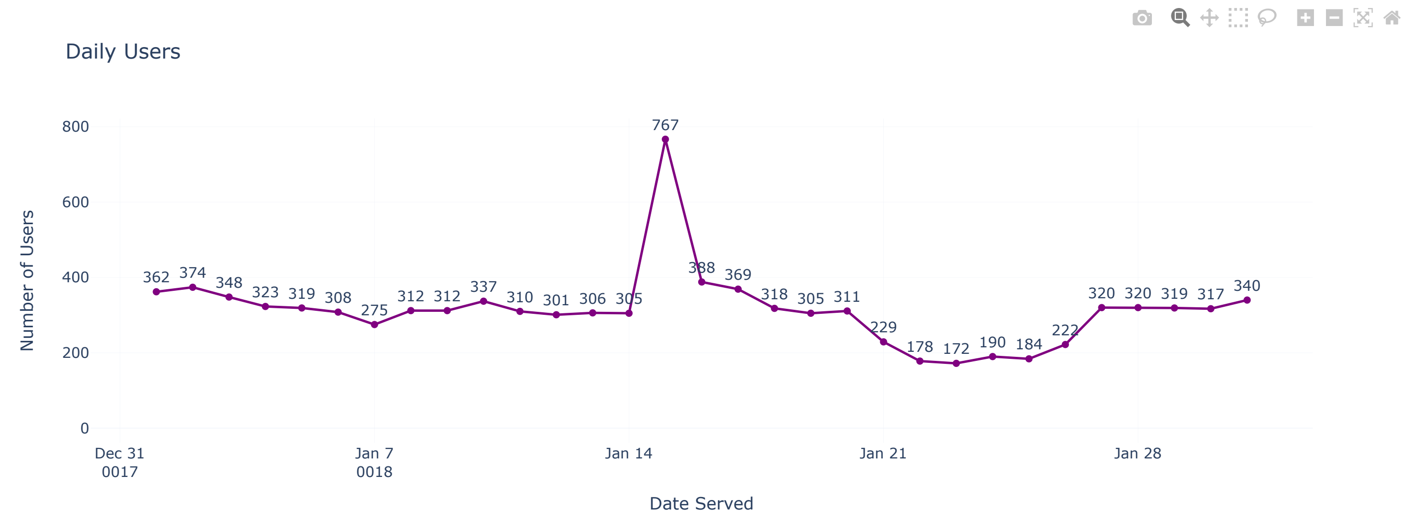
```python
import polars as pl
import numpy as np
import plotly.express as px

marketing = pl.read_csv('marketing.csv')
marketing = marketing.with_columns(pl.col(["date_served", "date_subscribed","date_canceled"]).str.to_date("%m/%d/%Y"))
print(marketing.describe())
```

```
shape: (9, 13)
┌───────────┬───────────┬───────────┬───────────┬───┬───────────┬───────────┬───────────┬───────────┐
│ statistic ┆ user_id   ┆ date_serv ┆ marketing ┆ … ┆ date_subs ┆ date_canc ┆ subscribi ┆ is_retai  │
│ ---       ┆ ---       ┆ ed        ┆ _channel  ┆   ┆ cribed    ┆ eled      ┆ ng_channe ┆ ned       │
│ str       ┆ str       ┆ ---       ┆ ---       ┆   ┆ ---       ┆ ---       ┆ l         ┆ ---       │
│           ┆           ┆ str       ┆ str       ┆   ┆ str       ┆ str       ┆ ---       ┆ f64       │
│           ┆           ┆           ┆           ┆   ┆           ┆           ┆ str       ┆           │
╞═══════════╪═══════════╪═══════════╪═══════════╪═══╪═══════════╪═══════════╪═══════════╪═══════════╡
│ count     ┆ 10037     ┆ 10021     ┆ 10022     ┆ … ┆ 1856      ┆ 577       ┆ 1856      ┆ 1856.0    │
│ null_coun ┆ 0         ┆ 16        ┆ 15        ┆ … ┆ 8181      ┆ 9460      ┆ 8181      ┆ 8181.0    │
│ t         ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆           │
│ mean      ┆ null      ┆ 0018-01-1 ┆ null      ┆ … ┆ 0018-01-1 ┆ 0018-03-0 ┆ null      ┆ 0.689116  │
│           ┆           ┆ 6         ┆           ┆   ┆ 5         ┆ 4         ┆           ┆           │
│ std       ┆ null      ┆ null      ┆ null      ┆ … ┆ null      ┆ null      ┆ null      ┆ null      │
│ min       ┆ a10000000 ┆ 0018-01-0 ┆ Email     ┆ … ┆ 0018-01-0 ┆ 0018-01-0 ┆ Email     ┆ 0.0       │
│           ┆ 1         ┆ 1         ┆           ┆   ┆ 1         ┆ 5         ┆           ┆           │
│ 25%       ┆ null      ┆ 0018-01-0 ┆ null      ┆ … ┆ 0018-01-0 ┆ 0018-02-0 ┆ null      ┆ null      │
│           ┆           ┆ 8         ┆           ┆   ┆ 7         ┆ 7         ┆           ┆           │
│ 50%       ┆ null      ┆ 0018-01-1 ┆ null      ┆ … ┆ 0018-01-1 ┆ 0018-03-0 ┆ null      ┆ null      │
│           ┆           ┆ 5         ┆           ┆   ┆ 5         ┆ 4         ┆           ┆           │
│ 75%       ┆ null      ┆ 0018-01-2 ┆ null      ┆ … ┆ 0018-01-1 ┆ 0018-04-0 ┆ null      ┆ null      │
│           ┆           ┆ 2         ┆           ┆   ┆ 9         ┆ 1         ┆           ┆           │
│ max       ┆ a10009245 ┆ 0018-01-3 ┆ Push      ┆ … ┆ 0018-01-3 ┆ 0018-05-0 ┆ Push      ┆ 1.0       │
│           ┆           ┆ 1         ┆           ┆   ┆ 1         ┆ 9         ┆           ┆           │
└───────────┴───────────┴───────────┴───────────┴───┴───────────┴───────────┴───────────┴───────────┘
```

```
daily_users = marketing[['date_served','user_id']].sort('date_served').group_by(['date_served']).agg(pl.col('user_id').n_unique().alias("users_num"))
print(daily_users.head())

fig = px.line(
    daily_users,
    x='date_served',
    y='users_num',
    title='Daily Users',
    template='plotly_white',
    labels={'date_served': 'Date Served', 'users_num': 'Number of Users'},
    markers=True,
    text='users_num'
)
fig.update_traces({'line_color':'purple','textposition':'top center'})
fig.update_layout(yaxis=dict(range=[0, None]))
fig.show()
```

```
shape: (5, 2)
┌─────────────┬───────────┐
│ date_served ┆ users_num │
│ ---         ┆ ---       │
│ date        ┆ u32       │
╞═════════════╪═══════════╡
│ null        ┆ 16        │
│ 0018-01-01  ┆ 362       │
│ 0018-01-02  ┆ 374       │
│ 0018-01-03  ┆ 348       │
│ 0018-01-04  ┆ 323       │
└─────────────┴───────────┘
```



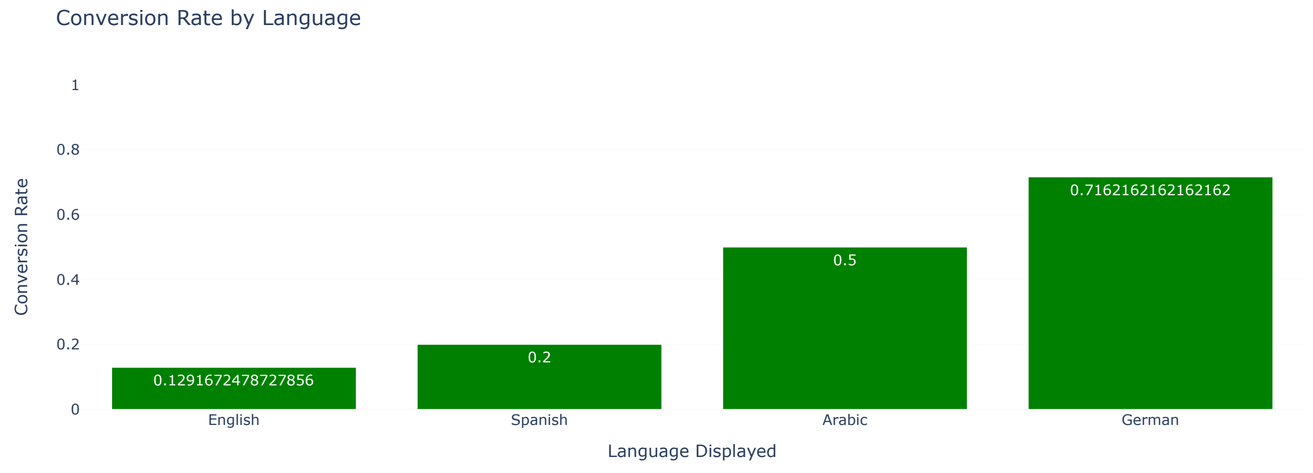
```
total = marketing['user_id'].n_unique()
subscribers = marketing.filter(pl.col('converted')==True)['user_id'].n_unique()
conversion_rate = subscribers/total
print("Conversion rate", round(conversion_rate*100, 2), "%",sep=" ")

retained = marketing.filter(pl.col('is_retained')==True)['user_id'].n_unique()
retention_rate = retained/subscribers
print("Retention rate", round(retention_rate*100, 2), "%",sep=" ")
```

```
Conversion rate 13.89 %
Retention rate 66.8 %
```

```
def conversion_rate(dataframe, column_names):
    column_conv = dataframe.filter(pl.col('converted')==True).group_by(column_names).agg(pl.col('user_id').n_unique().alias("users_converted"))
    column_total = dataframe.group_by(column_names).agg(pl.col('user_id').n_unique().alias("users_total"))

    conversion_df = column_conv.join(column_total, on=column_names, how='inner')
    conversion_df = conversion_df.with_columns(((pl.col("users_converted")/pl.col("users_total")).fill_nan(0)).alias("conversion_rate"))
    return conversion_df
```

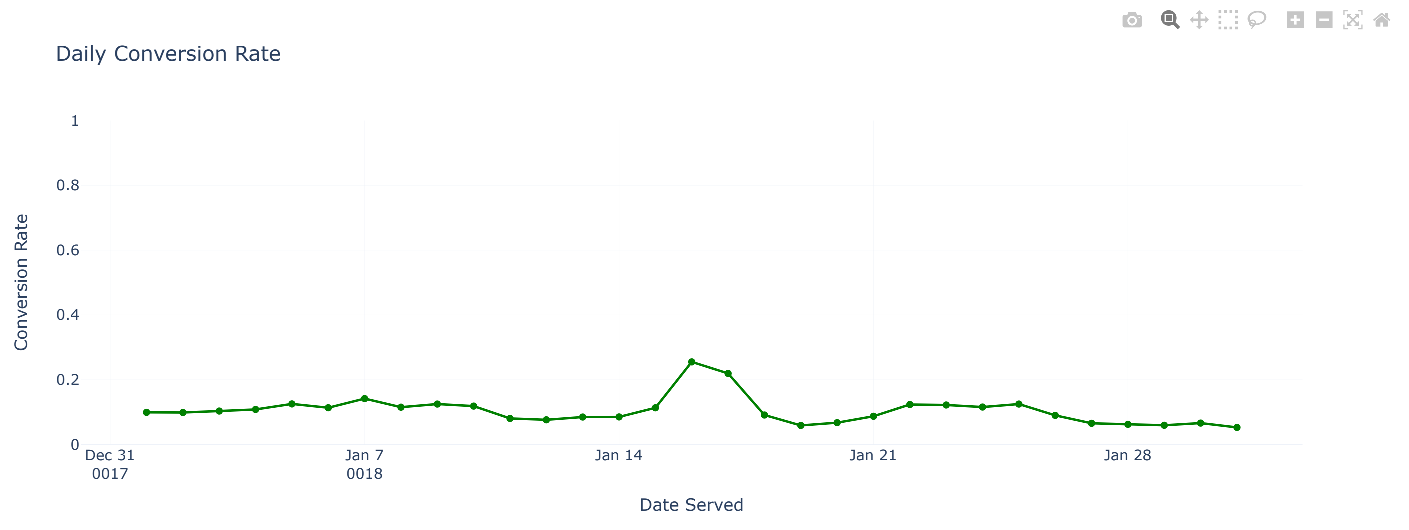
```python
language_conversion_rate = conversion_rate(marketing,'language_displayed')
print('Speaker conversion rate by language: ',language_conversion_rate, sep="\n")

fig = px.bar(
    language_conversion_rate.sort("conversion_rate"),
    x='language_displayed',
    y='conversion_rate',
    color_discrete_sequence=['green'],
    title='Conversion Rate by Language',
    template='plotly_white',
    labels={'language_displayed': 'Language Displayed', 'conversion_rate': 'Conversion Rate'},
    text="conversion_rate"
)
fig.update_layout(yaxis=dict(range=[0, 1]))
fig.show()
```

```
Speaker conversion rate by language:
shape: (4, 4)
┌────────────────────┬─────────────────┬─────────────┬─────────────────┐
│ language_displayed ┆ users_converted ┆ users_total ┆ conversion_rate │
│ ---                ┆ ---             ┆ ---         ┆ ---             │
│ str                ┆ u32             ┆ u32         ┆ f64             │
╞════════════════════╪═════════════════╪═════════════╪═════════════════╡
│ Arabic             ┆ 12              ┆ 24          ┆ 0.5             │
│ English            ┆ 926             ┆ 7169        ┆ 0.129167        │
│ German             ┆ 53              ┆ 74          ┆ 0.716216        │
│ Spanish            ┆ 24              ┆ 120         ┆ 0.2             │
└────────────────────┴─────────────────┴─────────────┴─────────────────┘
```



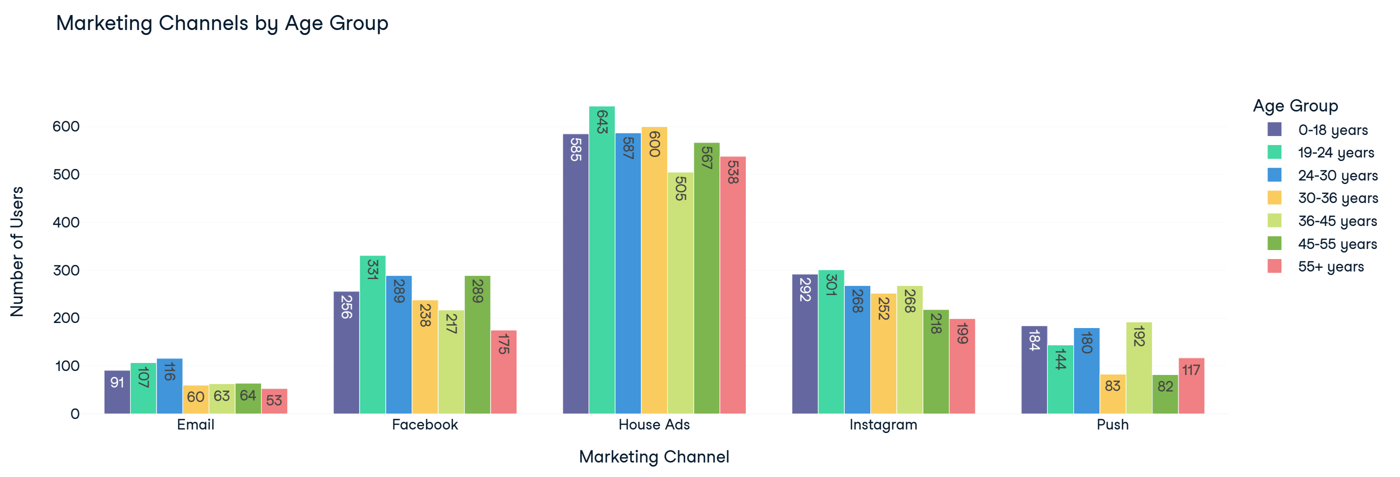
```python
daily_conversion_rate = conversion_rate(marketing,'date_served')
print("Daily Conversion Rate: ", daily_conversion_rate, sep="\n")

fig = px.line(
    daily_conversion_rate.sort('date_served'),
    x='date_served',
    y='conversion_rate',
    title='Daily Conversion Rate',
    template='plotly_white',
    labels={'date_served': 'Date Served', 'conversion_rate': 'Conversion Rate'},
    markers=True,
)
fig.update_traces({'line_color':'green'})
fig.update_layout(yaxis=dict(range=[0, 1]))
fig.show()
```

```
Daily Conversion Rate:
shape: (31, 4)
┌─────────────┬─────────────────┬─────────────┬─────────────────┐
│ date_served ┆ users_converted ┆ users_total ┆ conversion_rate │
│ ---         ┆ ---             ┆ ---         ┆ ---             │
│ date        ┆ u32             ┆ u32         ┆ f64             │
╞═════════════╪═════════════════╪═════════════╪═════════════════╡
│ 0018-01-05  ┆ 40              ┆ 319         ┆ 0.125392        │
│ 0018-01-11  ┆ 25              ┆ 310         ┆ 0.080645        │
│ 0018-01-19  ┆ 18              ┆ 305         ┆ 0.059016        │
│ 0018-01-25  ┆ 23              ┆ 184         ┆ 0.125           │
│ 0018-01-08  ┆ 36              ┆ 312         ┆ 0.115385        │
│ …           ┆ …               ┆ …           ┆ …               │
│ 0018-01-07  ┆ 39              ┆ 275         ┆ 0.141818        │
│ 0018-01-15  ┆ 87              ┆ 767         ┆ 0.113429        │
│ 0018-01-06  ┆ 35              ┆ 308         ┆ 0.113636        │
│ 0018-01-03  ┆ 36              ┆ 348         ┆ 0.103448        │
│ 0018-01-13  ┆ 26              ┆ 306         ┆ 0.084967        │
└─────────────┴─────────────────┴─────────────┴─────────────────┘
```



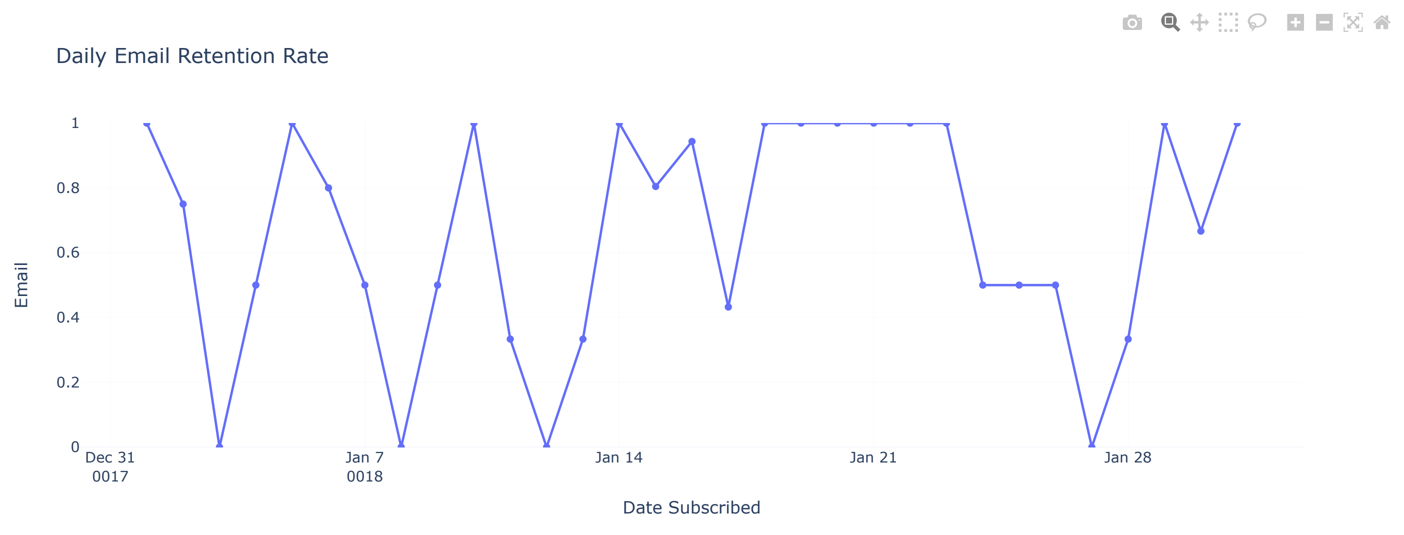
```python
channel_age = marketing.group_by(['marketing_channel', 'age_group']).agg(pl.col('user_id').n_unique().alias("users_num"))
print(channel_age.head())

fig = px.bar(
    channel_age.sort(['marketing_channel','age_group']),
    x="marketing_channel",
    y="users_num",
    color="age_group",
    barmode="group",
    title="Marketing Channels by Age Group",
    labels={"marketing_channel": "Marketing Channel", "users_num": "Number of Users", 'age_group':"Age Group"},
    text="users_num"
)
fig.show()
```

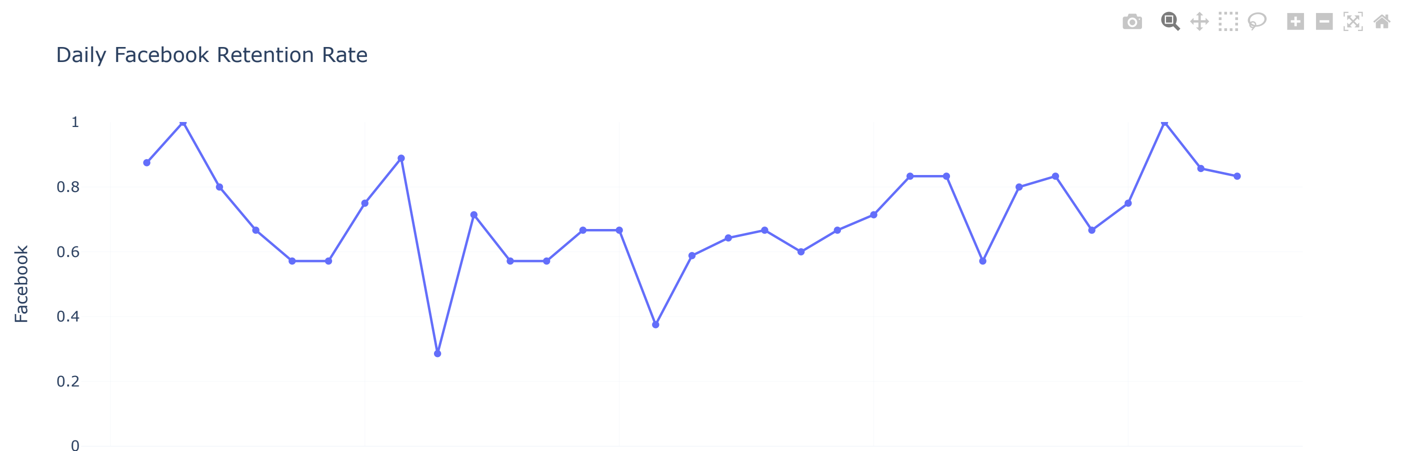
```
shape: (5, 3)
┌───────────────────┬─────────────┬───────────┐
│ marketing_channel ┆ age_group   ┆ users_num │
│ ---               ┆ ---         ┆ ---       │
│ str               ┆ str         ┆ u32       │
╞═══════════════════╪═════════════╪═══════════╡
│ Email             ┆ 55+ years   ┆ 53        │
│ null              ┆ 24-30 years ┆ 2         │
│ Facebook          ┆ 24-30 years ┆ 289       │
│ null              ┆ 19-24 years ┆ 3         │
│ House Ads         ┆ 30-36 years ┆ 600       │
└───────────────────┴─────────────┴───────────┘
```



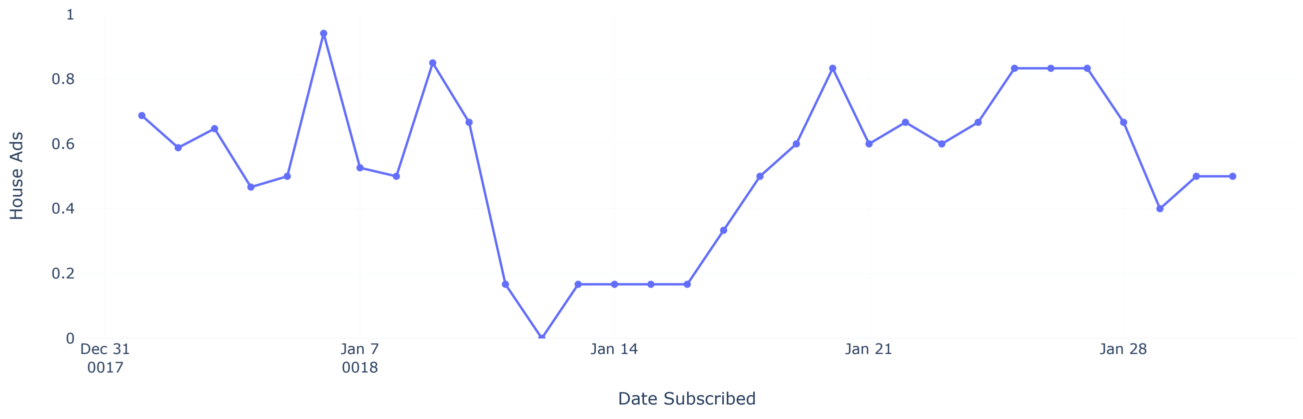
```python
sub_total = marketing.group_by(['date_subscribed', 'subscribing_channel']).agg(pl.col('user_id').n_unique().alias('sub_num'))
retention_subs =
marketing.filter(pl.col('is_retained')==True).group_by(['date_subscribed','subscribing_channel']).agg(pl.col('user_id').n_unique().alias("users_retained"))
retention_df = retention_subs.join(sub_total,on=['date_subscribed', 'subscribing_channel'], how='inner')
retention_df = retention_df.with_columns((pl.col("users_retained")/pl.col("sub_num")).alias("retention_rate"))
retention_df = retention_df.pivot(values='retention_rate', index='date_subscribed', columns='subscribing_channel')
retention_df = retention_df.fill_nan(0).fill_null(0)
columns = sorted(retention_df.columns)
columns.remove('date_subscribed')
print(retention_df)

for column in columns:
    fig = px.line(
        retention_df.sort('date_subscribed'),
        x='date_subscribed',
        y=column,
        title=f'Daily {column} Retention Rate',
        template='plotly_white',
        labels={'date_subscribed': 'Date Subscribed', 'retention_rate': 'Retention Rate'},
        markers=True
    )
    fig.update_layout(yaxis=dict(range=[0, 1]))
    fig.show()
```

```
shape: (31, 6)
┌─────────────────┬──────────┬───────────┬──────────┬───────────┬──────────┐
│ date_subscribed ┆ Facebook ┆ Instagram ┆ Push     ┆ House Ads ┆ Email    │
│ ---             ┆ ---      ┆ ---       ┆ ---      ┆ ---       ┆ ---      │
│ date            ┆ f64      ┆ f64       ┆ f64      ┆ f64       ┆ f64      │
╞═════════════════╪══════════╪═══════════╪══════════╪═══════════╪══════════╡
│ 0018-01-22      ┆ 0.833333 ┆ 0.75      ┆ 1.0      ┆ 0.666667  ┆ 1.0      │
│ 0018-01-30      ┆ 0.857143 ┆ 1.0       ┆ 0.5      ┆ 0.5       ┆ 0.666667 │
│ 0018-01-15      ┆ 0.375    ┆ 0.875     ┆ 1.0      ┆ 0.166667  ┆ 0.804348 │
│ 0018-01-18      ┆ 0.666667 ┆ 0.9       ┆ 0.0      ┆ 0.5       ┆ 1.0      │
│ 0018-01-28      ┆ 0.75     ┆ 0.666667  ┆ 1.0      ┆ 0.666667  ┆ 0.333333 │
│ …               ┆ …        ┆ …         ┆ …        ┆ …         ┆ …        │
│ 0018-01-31      ┆ 0.833333 ┆ 0.666667  ┆ 0.5      ┆ 0.5       ┆ 1.0      │
│ 0018-01-24      ┆ 0.571429 ┆ 0.666667  ┆ 1.0      ┆ 0.666667  ┆ 0.5      │
│ 0018-01-17      ┆ 0.642857 ┆ 0.894737  ┆ 0.9      ┆ 0.333333  ┆ 0.432432 │
│ 0018-01-27      ┆ 0.666667 ┆ 0.4       ┆ 0.333333 ┆ 0.833333  ┆ 0.0      │
│ 0018-01-06      ┆ 0.571429 ┆ 0.5       ┆ 0.5      ┆ 0.941176  ┆ 0.8      │
└─────────────────┴──────────┴───────────┴──────────┴───────────┴──────────┘
```

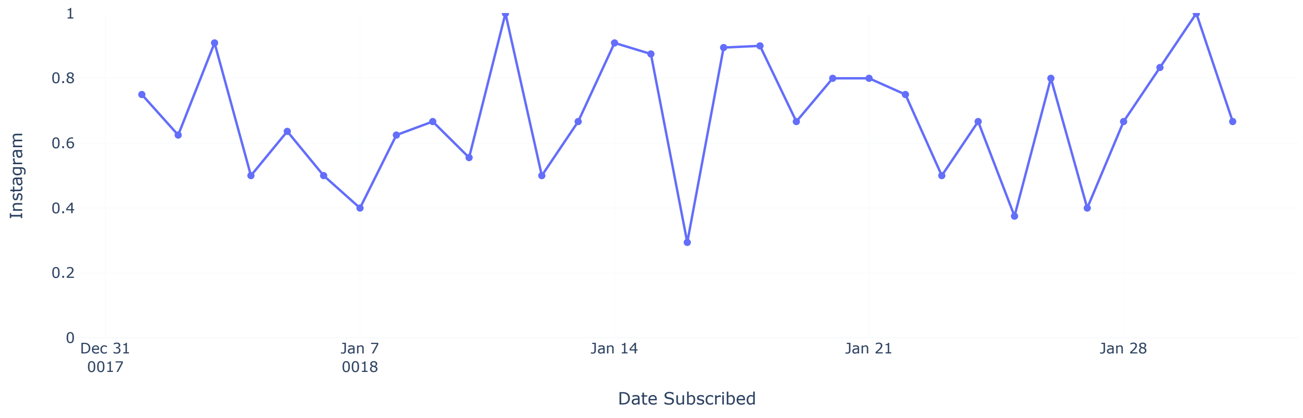




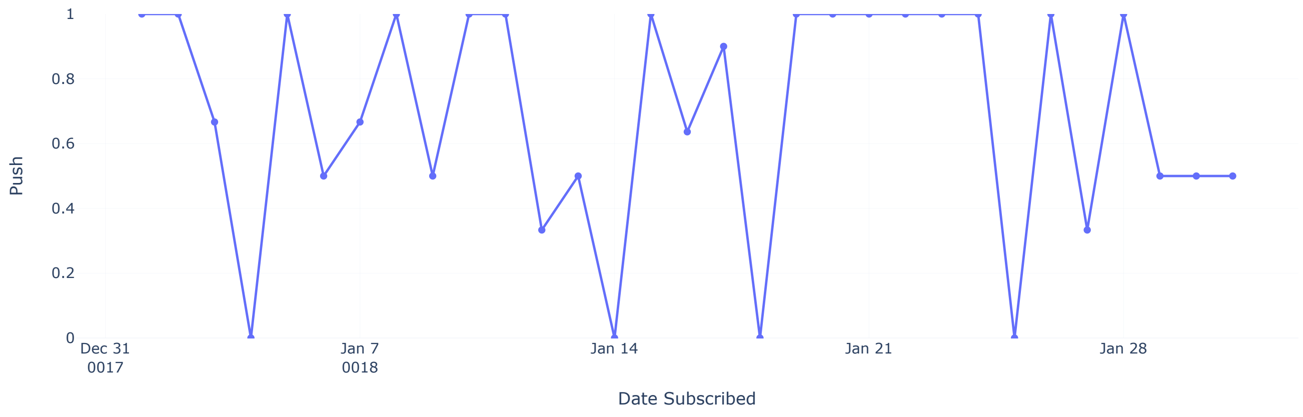
Daily House Ads Retention Rate



Daily Instagram Retention Rate



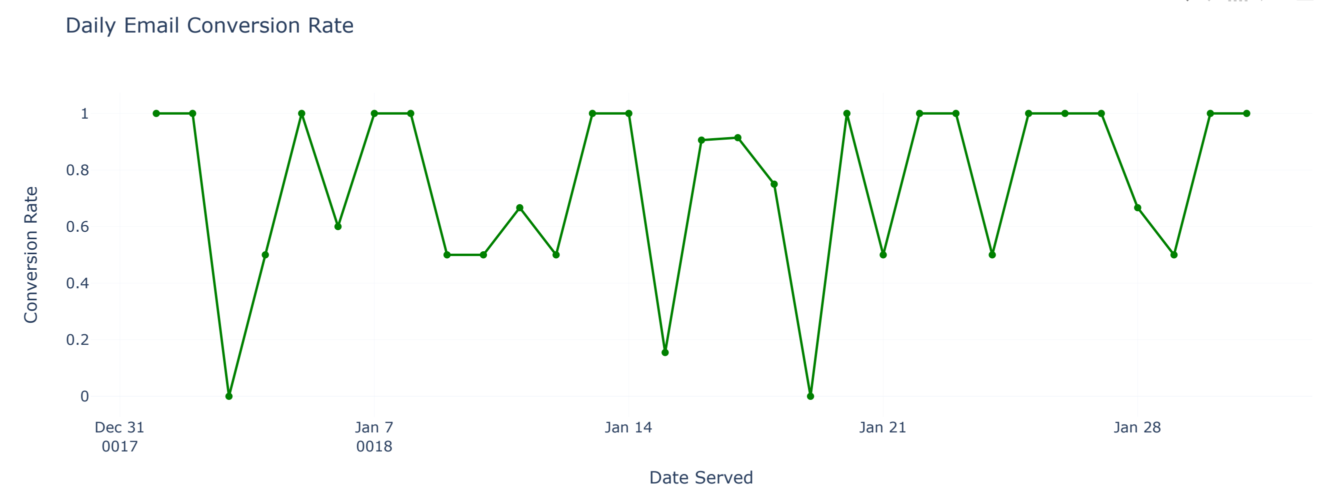
Daily Push Retention Rate

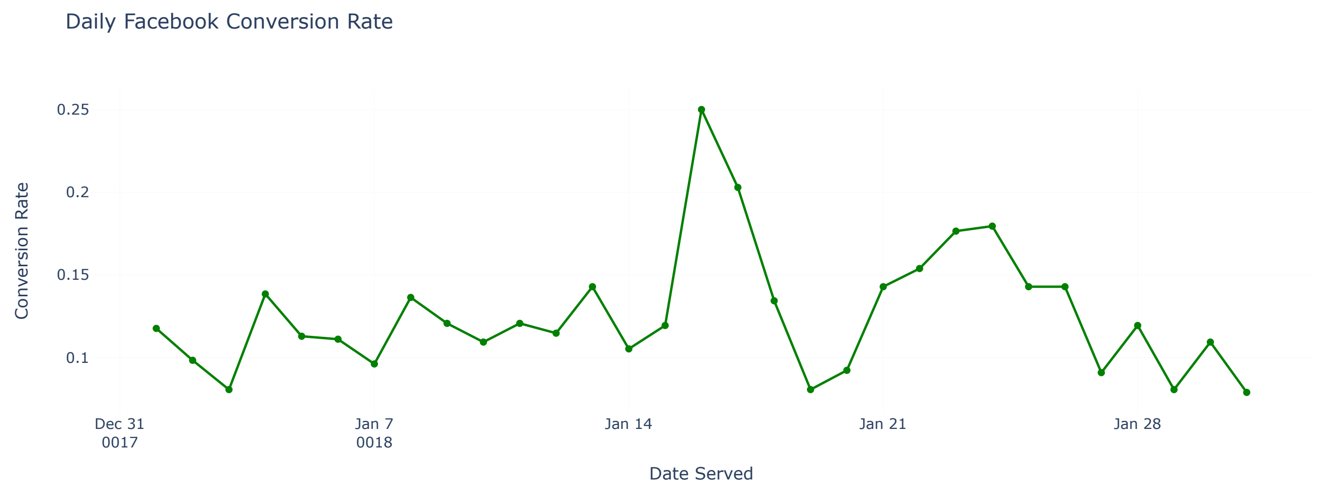


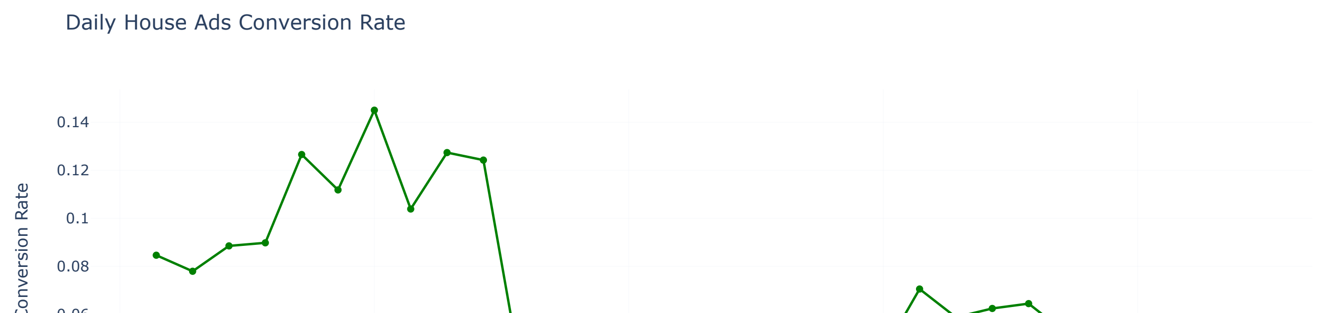
```python
def plotting_conv(dataframe):
    columns = sorted(dataframe.columns)
    columns.remove('date_served')
    for column in columns:
        fig = px.line(
            dataframe,
            x=dataframe.get_column('date_served'),
            y=column,
            title=f'Daily {column} Conversion Rate',
            labels={'date_served': 'Date served', column: 'Conversion Rate'},
            markers=True

        )
        fig.update_layout(template='plotly_white', hovermode='x', yaxis=dict(range=[0, None]),xaxis_title="Date Served")
        fig.update_traces(line_color='green')
        fig.show()
```

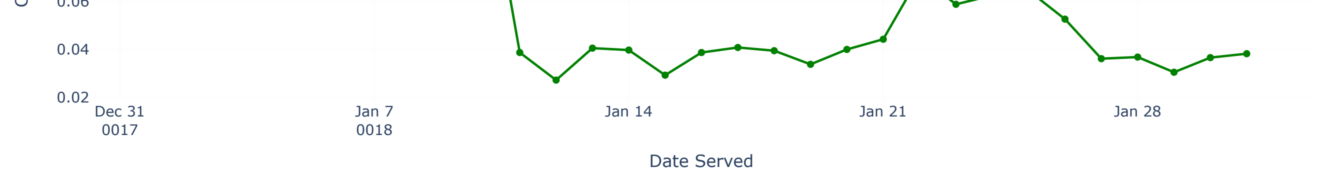
```
marketing_channel_conv = conversion_rate(marketing, ['date_served', 'marketing_channel'])
marketing_channel_df = marketing_channel_conv.pivot(index='date_served', columns='marketing_channel', values='conversion_rate').fill_null(0)
print(marketing_channel_df)

plotting_conv(marketing_channel_df.sort('date_served'))
```

```
shape: (31, 6)
┌─────────────┬───────────┬──────────┬──────────┬──────────┬───────────┐
│ date_served ┆ House Ads ┆ Push     ┆ Facebook ┆ Email    ┆ Instagram │
│ ---         ┆ ---       ┆ ---      ┆ ---      ┆ ---      ┆ ---       │
│ date        ┆ f64       ┆ f64      ┆ f64      ┆ f64      ┆ f64       │
╞═════════════╪═══════════╪══════════╪══════════╪══════════╪═══════════╡
│ 0018-01-16  ┆ 0.03871   ┆ 0.261905 ┆ 0.25     ┆ 0.90566  ┆ 0.239437  │
│ 0018-01-04  ┆ 0.08982   ┆ 0.058824 ┆ 0.138462 ┆ 0.5      ┆ 0.126984  │
│ 0018-01-07  ┆ 0.145038  ┆ 0.088235 ┆ 0.096154 ┆ 1.0      ┆ 0.175439  │
│ 0018-01-08  ┆ 0.103896  ┆ 0.064516 ┆ 0.136364 ┆ 1.0      ┆ 0.125     │
│ 0018-01-03  ┆ 0.088542  ┆ 0.083333 ┆ 0.080645 ┆ 0.0      ┆ 0.171875  │
│ …           ┆ …         ┆ …        ┆ …        ┆ …        ┆ …         │
│ 0018-01-14  ┆ 0.039735  ┆ 0.058824 ┆ 0.105263 ┆ 1.0      ┆ 0.171875  │
│ 0018-01-17  ┆ 0.040816  ┆ 0.232558 ┆ 0.202899 ┆ 0.914286 ┆ 0.246753  │
│ 0018-01-23  ┆ 0.058824  ┆ 0.125    ┆ 0.176471 ┆ 1.0      ┆ 0.166667  │
│ 0018-01-29  ┆ 0.030488  ┆ 0.058824 ┆ 0.080645 ┆ 0.5      ┆ 0.095238  │
│ 0018-01-21  ┆ 0.044248  ┆ 0.1      ┆ 0.142857 ┆ 0.5      ┆ 0.104167  │
└─────────────┴───────────┴──────────┴──────────┴──────────┴───────────┘
```

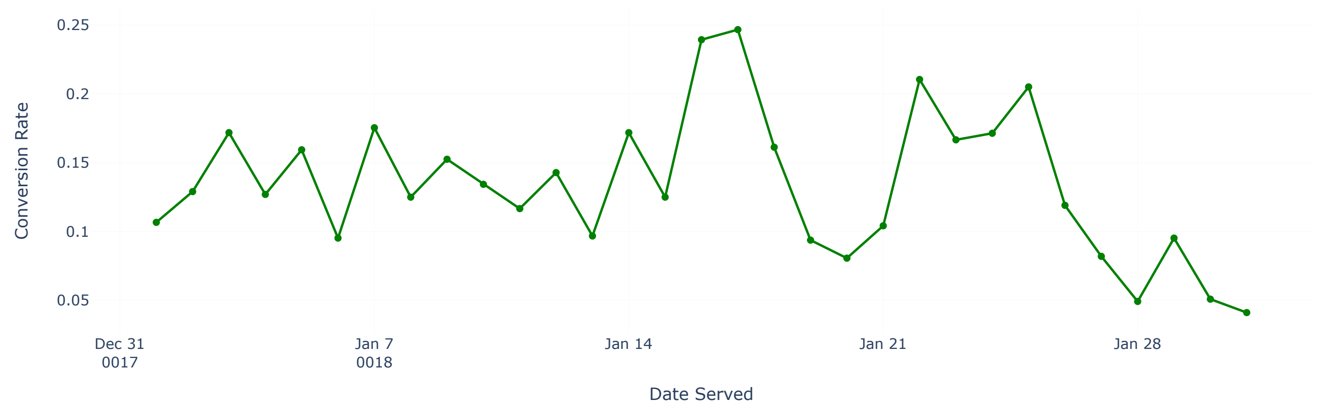




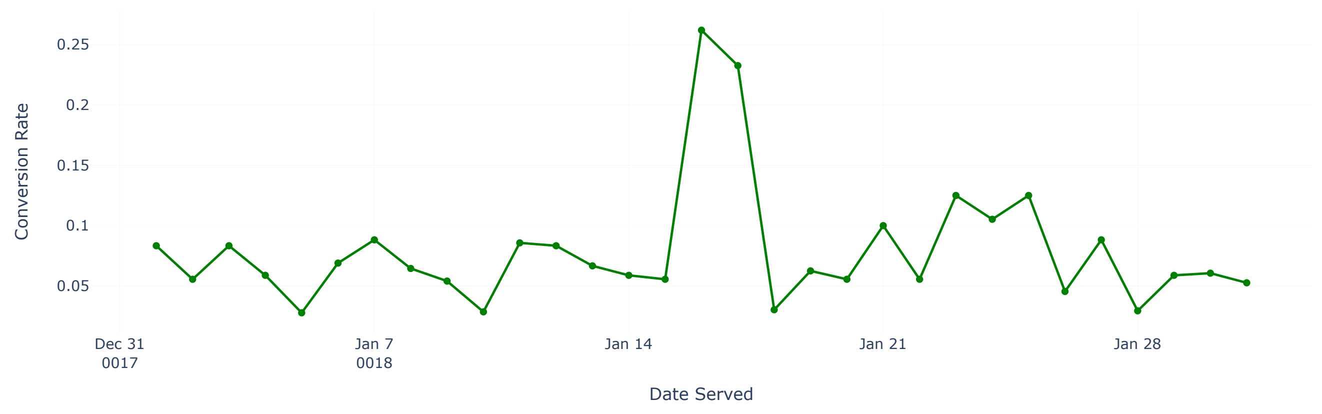




Daily Instagram Conversion Rate

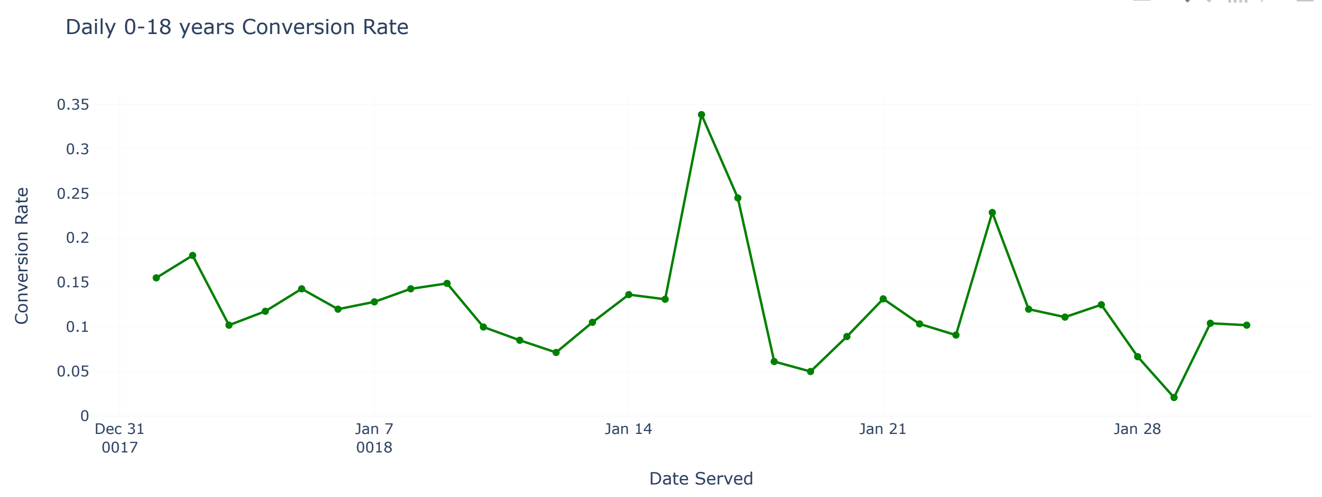


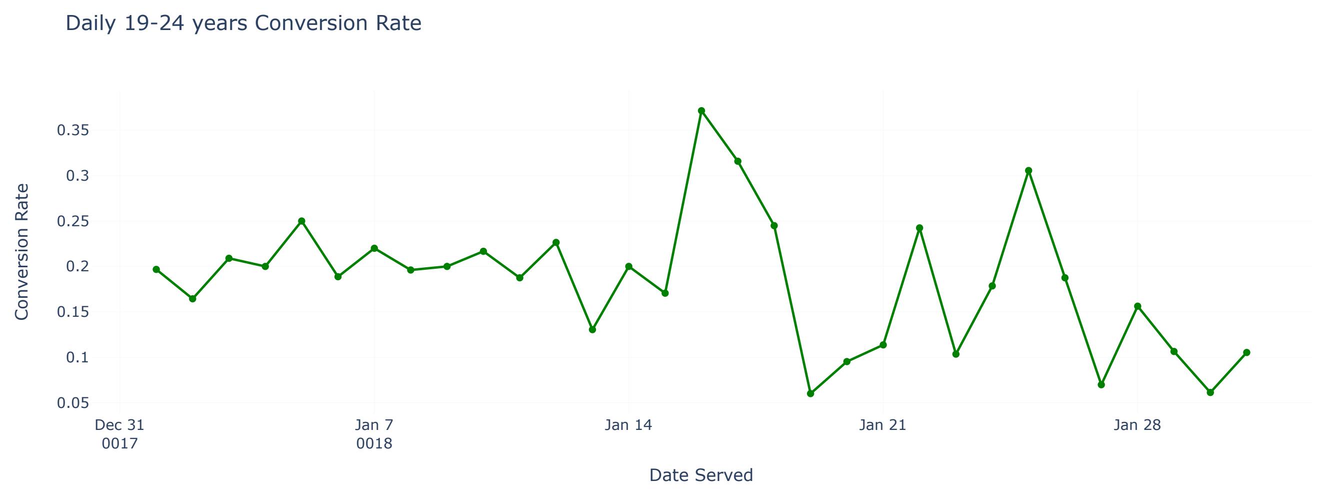
Daily Push Conversion Rate

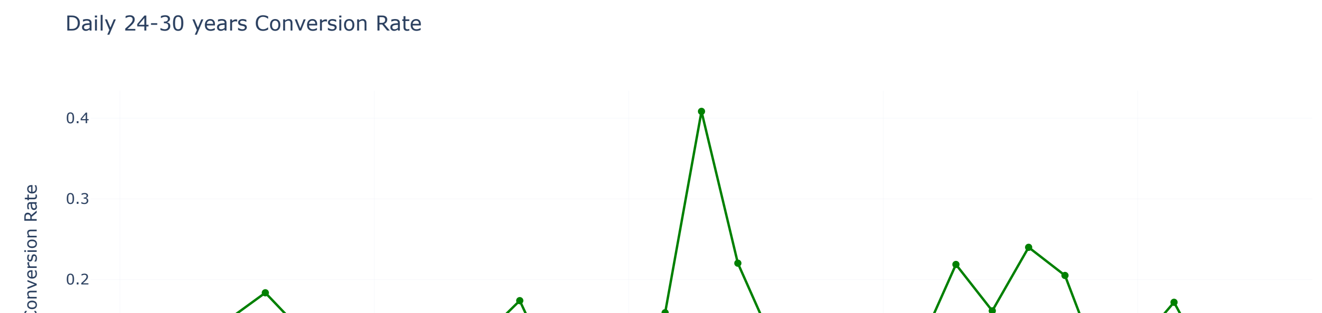


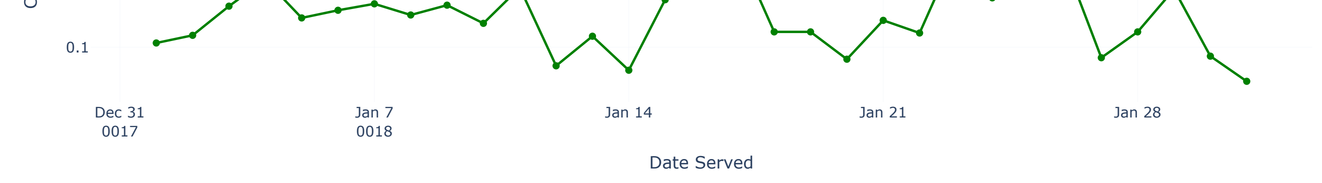
```
age_group_conv = conversion_rate(marketing, ['date_served', 'age_group'])
age_group_df = age_group_conv.pivot(index='date_served', columns='age_group', values='conversion_rate').fill_null(0)
print(age_group_df)
plotting_conv(age_group_df.sort('date_served'))
```

```
shape: (31, 8)
┌────────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┬──────────┐
│ date_serve ┆ 0-18 years ┆ 45-55  ┆ 19-24    ┆ 55+ years ┆ 36-45  ┆ 24-30    ┆ 30-36    │
│ d          ┆ ---      ┆ years    ┆ years    ┆ ---      ┆ years    ┆ years    ┆ years    │
│ ---        ┆ f64      ┆ ---      ┆ ---      ┆ f64      ┆ ---      ┆ ---      ┆ ---      │
│ date       ┆          ┆ f64      ┆ f64      ┆          ┆ f64      ┆ f64      ┆ f64      │
╞════════════╪══════════╪══════════╪══════════╪══════════╪══════════╪══════════╪══════════╡
│ 0018-01-12 ┆ 0.071429 ┆ 0.0      ┆ 0.226415 ┆ 0.045455 ┆ 0.0      ┆ 0.076923 ┆ 0.075    │
│ 0018-01-06 ┆ 0.12     ┆ 0.068182 ┆ 0.188679 ┆ 0.078947 ┆ 0.073171 ┆ 0.145833 ┆ 0.078947 │
│ 0018-01-14 ┆ 0.136364 ┆ 0.073171 ┆ 0.2      ┆ 0.0      ┆ 0.025641 ┆ 0.071429 ┆ 0.025    │
│ 0018-01-15 ┆ 0.131148 ┆ 0.074468 ┆ 0.170543 ┆ 0.035294 ┆ 0.061224 ┆ 0.15894  ┆ 0.090909 │
│ 0018-01-13 ┆ 0.105263 ┆ 0.023256 ┆ 0.130435 ┆ 0.052632 ┆ 0.1      ┆ 0.113636 ┆ 0.052632 │
│ …          ┆ …        ┆ …        ┆ …        ┆ …        ┆ …        ┆ …        ┆ …        │
│ 0018-01-28 ┆ 0.066667 ┆ 0.0      ┆ 0.15625  ┆ 0.0      ┆ 0.042553 ┆ 0.119048 ┆ 0.0      │
│ 0018-01-24 ┆ 0.228571 ┆ 0.0      ┆ 0.178571 ┆ 0.055556 ┆ 0.09375  ┆ 0.16129  ┆ 0.0      │
│ 0018-01-29 ┆ 0.020833 ┆ 0.041667 ┆ 0.106383 ┆ 0.0      ┆ 0.0      ┆ 0.171875 ┆ 0.0      │
│ 0018-01-03 ┆ 0.102041 ┆ 0.047619 ┆ 0.208955 ┆ 0.043478 ┆ 0.06     ┆ 0.150943 ┆ 0.042553 │
│ 0018-01-18 ┆ 0.061224 ┆ 0.074074 ┆ 0.244898 ┆ 0.088235 ┆ 0.02439  ┆ 0.119048 ┆ 0.020408 │
└────────────┴──────────┴──────────┴──────────┴──────────┴──────────┴──────────┴──────────┘
```

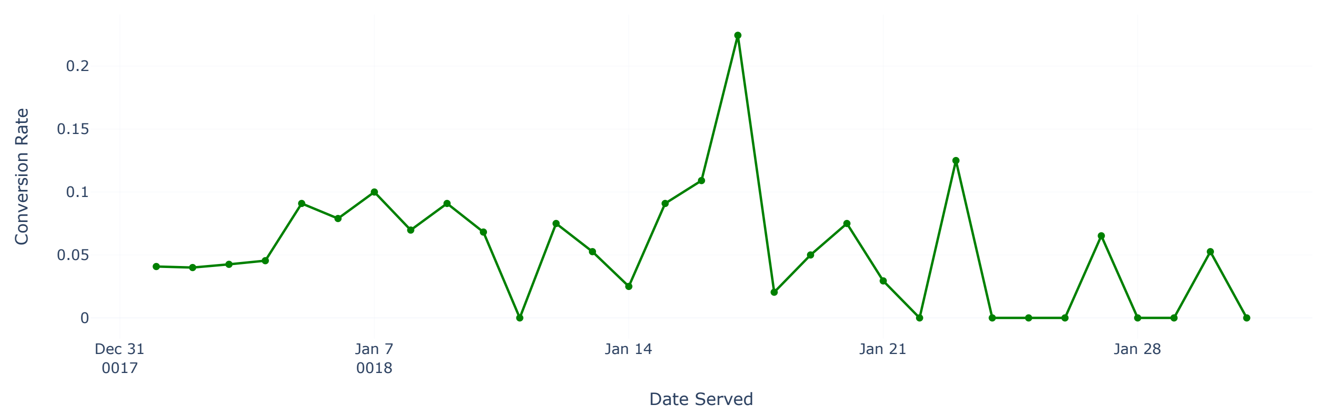




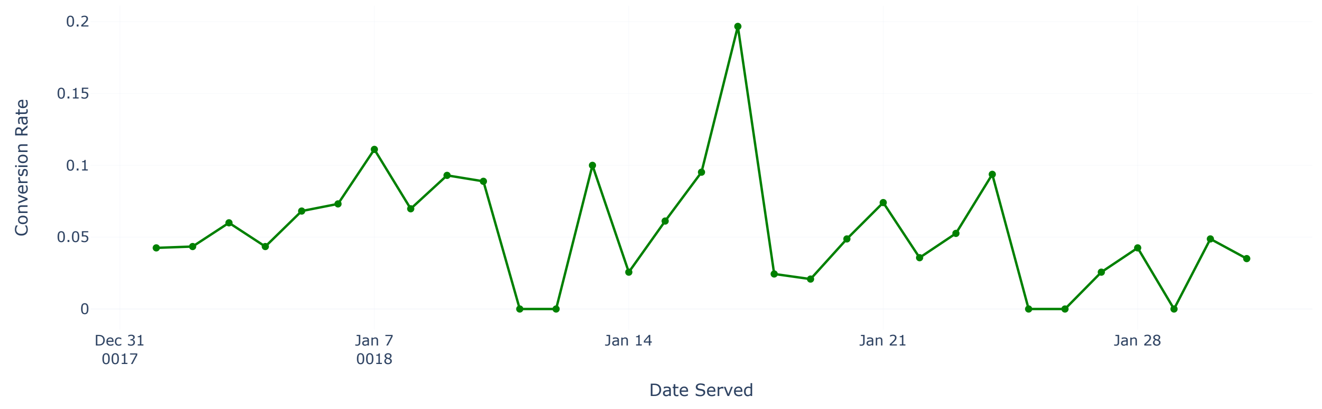




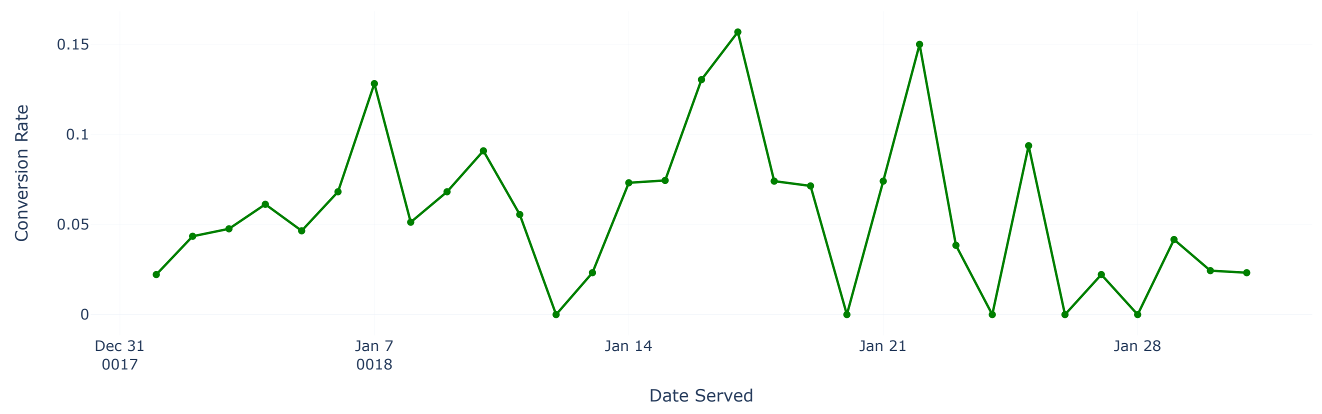
Daily 30-36 years Conversion Rate



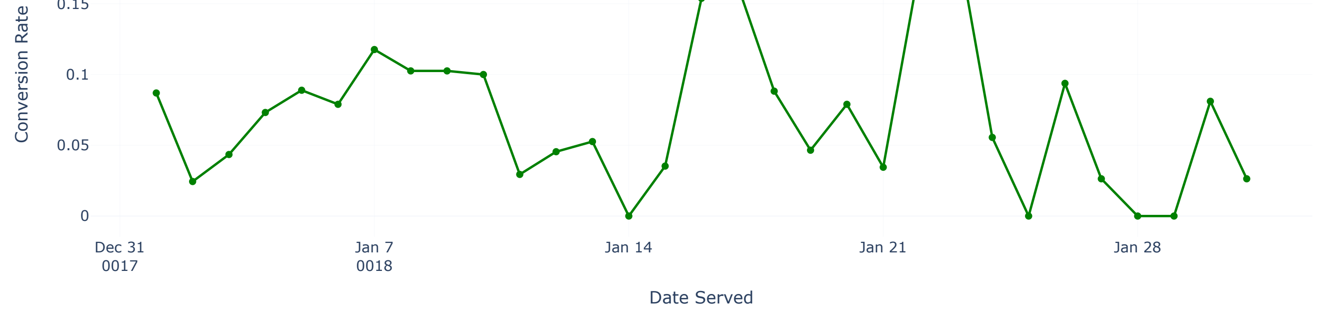
Daily 36-45 years Conversion Rate



Daily 45-55 years Conversion Rate

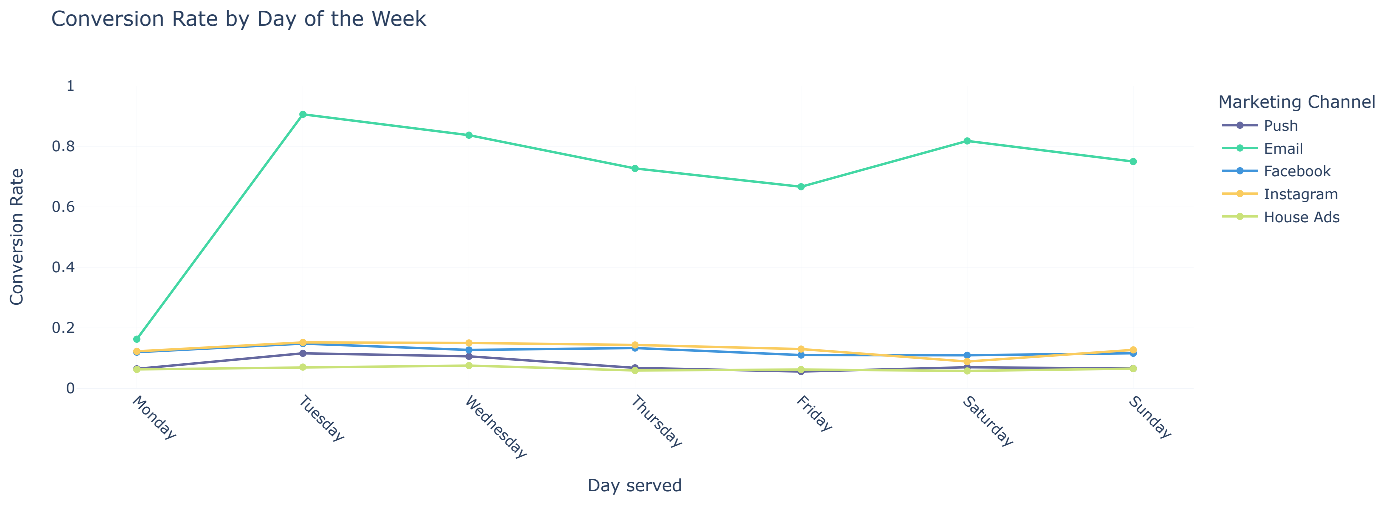


Daily 55+ years Conversion Rate



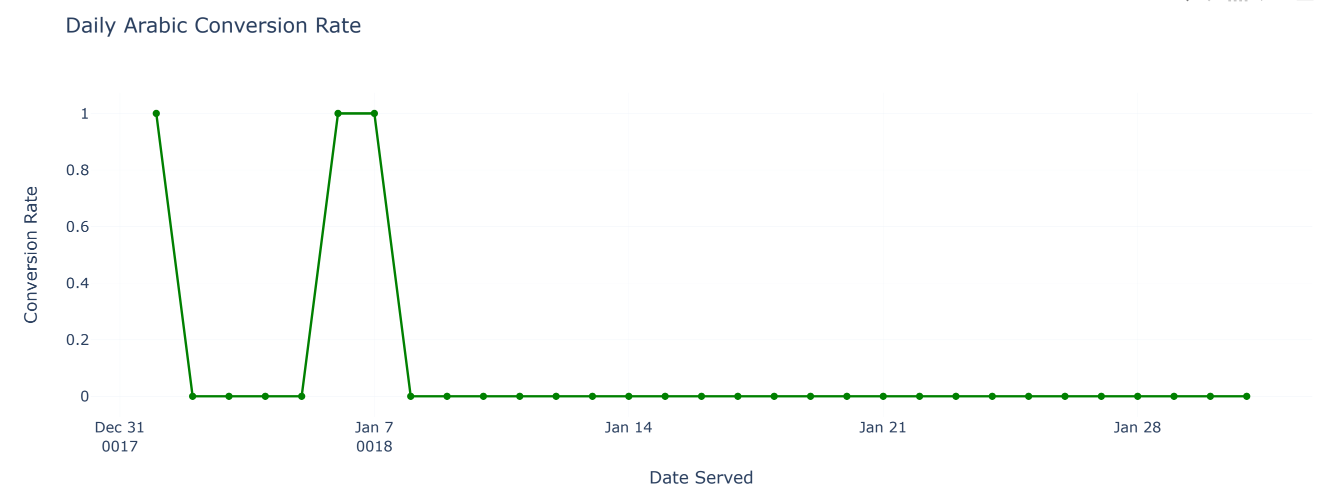
```python
marketing = marketing.with_columns((pl.col("date_served").dt.weekday()).alias("DoW_served"))
DoW_conversion = conversion_rate(marketing, ['DoW_served','marketing_channel'] )
DoW_conversion_df = DoW_conversion.pivot(columns='marketing_channel', index='DoW_served', values='conversion_rate').fill_null(0)
DoW_conversion_df = DoW_conversion_df.sort('DoW_served')
print(DoW_conversion_df)

fig = px.line(
        DoW_conversion_df,
        x='DoW_served',
        y= DoW_conversion_df.columns,
        title='Conversion Rate by Day of the Week',
        labels={'variable': 'Marketing Channel', 'DoW_served':'Day served','value':'Conversion Rate'},
        markers=True
)
fig.update_layout(
        xaxis=dict(tickangle=45),
        yaxis=dict(range=[0, 1]),
        template='plotly_white',

)
fig.update_xaxes(
    tickvals=[1,2,3,4,5,6,7],
    ticktext= ["Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"]
)
fig.show()
```

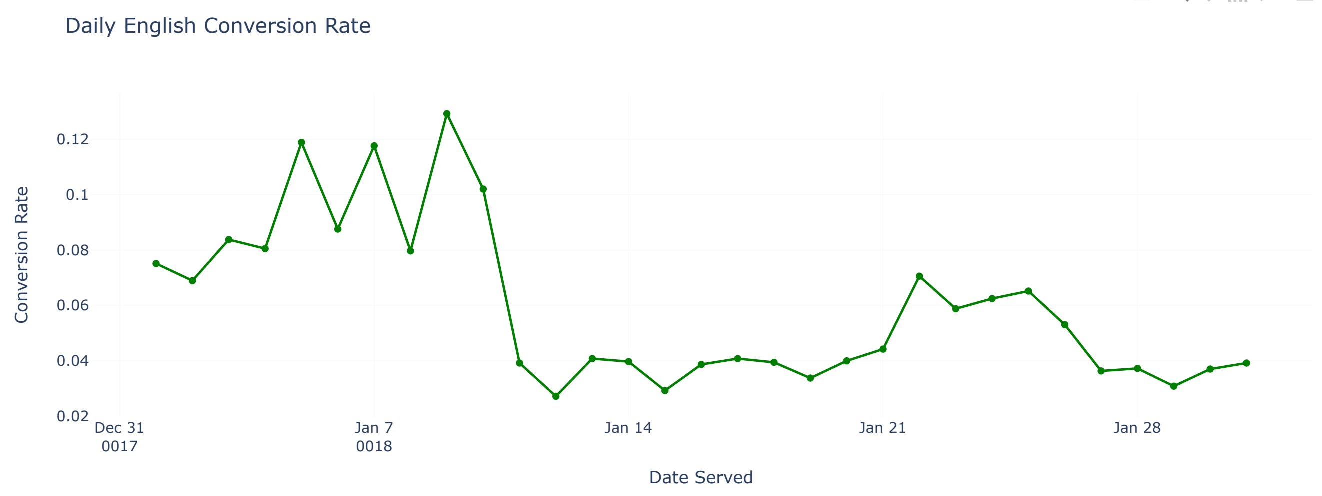
```
shape: (7, 6)
┌────────────┬──────────┬──────────┬──────────┬───────────┬───────────┐
│ DoW_served ┆ Push     ┆ Email    ┆ Facebook ┆ Instagram ┆ House Ads │
│ ---        ┆ ---      ┆ ---      ┆ ---      ┆ ---       ┆ ---       │
│ i8         ┆ f64      ┆ f64      ┆ f64      ┆ f64       ┆ f64       │
╞════════════╪══════════╪══════════╪══════════╪═══════════╪═══════════╡
│ 1          ┆ 0.064516 ┆ 0.162621 ┆ 0.119601 ┆ 0.122517  ┆ 0.06266   │
│ 2          ┆ 0.115854 ┆ 0.90625  ┆ 0.147887 ┆ 0.151943  ┆ 0.0703125 │
│ 3          ┆ 0.105882 ┆ 0.837209 ┆ 0.127036 ┆ 0.15016   ┆ 0.075269  │
│ 4          ┆ 0.067797 ┆ 0.727273 ┆ 0.133333 ┆ 0.143498  ┆ 0.059034  │
│ 5          ┆ 0.055556 ┆ 0.666667 ┆ 0.110132 ┆ 0.12987   ┆ 0.062278  │
│ 6          ┆ 0.069767 ┆ 0.818182 ┆ 0.109375 ┆ 0.08871   ┆ 0.057566  │
│ 7          ┆ 0.065574 ┆ 0.75     ┆ 0.116071 ┆ 0.127193  ┆ 0.065217  │
└────────────┴──────────┴──────────┴──────────┴───────────┴───────────┘
```

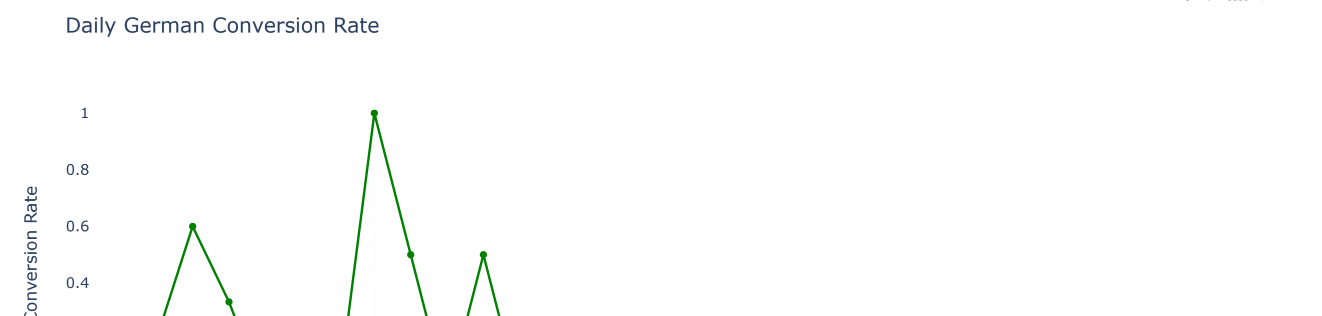


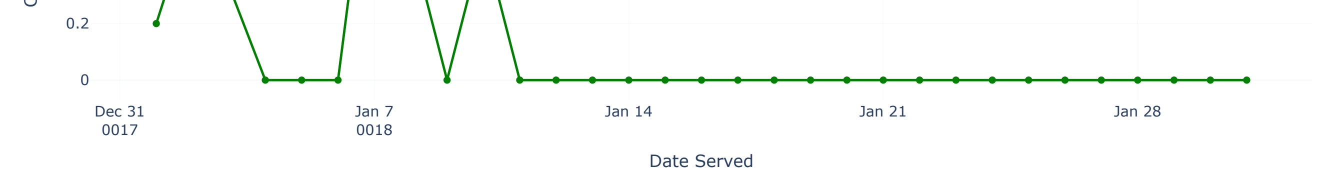
```
house_ads = marketing.filter(pl.col('marketing_channel')=='House Ads')
conv_lang_channel = conversion_rate(house_ads,['date_served','language_displayed'])
conv_lang_df = conv_lang_channel.pivot(columns='language_displayed', index='date_served', values='conversion_rate').fill_null(0)
print(conv_lang_df)
plotting_conv(conv_lang_df.sort('date_served'))
```

```
shape: (31, 5)
┌─────────────┬──────────┬────────┬──────────┬──────────┐
│ date_served ┆ English  ┆ Arabic ┆ German   ┆ Spanish  │
│ ---         ┆ ---      ┆ ---    ┆ ---      ┆ ---      │
│ date        ┆ f64      ┆ f64    ┆ f64      ┆ f64      │
╞═════════════╪══════════╪════════╪══════════╪══════════╡
│ 0018-01-31  ┆ 0.039216 ┆ 0.0    ┆ 0.0      ┆ 0.0      │
│ 0018-01-07  ┆ 0.117647 ┆ 1.0    ┆ 1.0      ┆ 0.125    │
│ 0018-01-03  ┆ 0.083799 ┆ 0.0    ┆ 0.333333 ┆ 0.125    │
│ 0018-01-25  ┆ 0.065217 ┆ 0.0    ┆ 0.0      ┆ 0.0      │
│ 0018-01-06  ┆ 0.087591 ┆ 1.0    ┆ 0.0      ┆ 0.2      │
│ …           ┆ …        ┆ …      ┆ …        ┆ …        │
│ 0018-01-05  ┆ 0.118881 ┆ 0.0    ┆ 0.0      ┆ 0.214286 │
│ 0018-01-21  ┆ 0.044248 ┆ 0.0    ┆ 0.0      ┆ 0.0      │
│ 0018-01-26  ┆ 0.053097 ┆ 0.0    ┆ 0.0      ┆ 0.0      │
│ 0018-01-16  ┆ 0.03871  ┆ 0.0    ┆ 0.0      ┆ 0.0      │
│ 0018-01-12  ┆ 0.027211 ┆ 0.0    ┆ 0.0      ┆ 0.0      │
└─────────────┴──────────┴────────┴──────────┴──────────┘
```

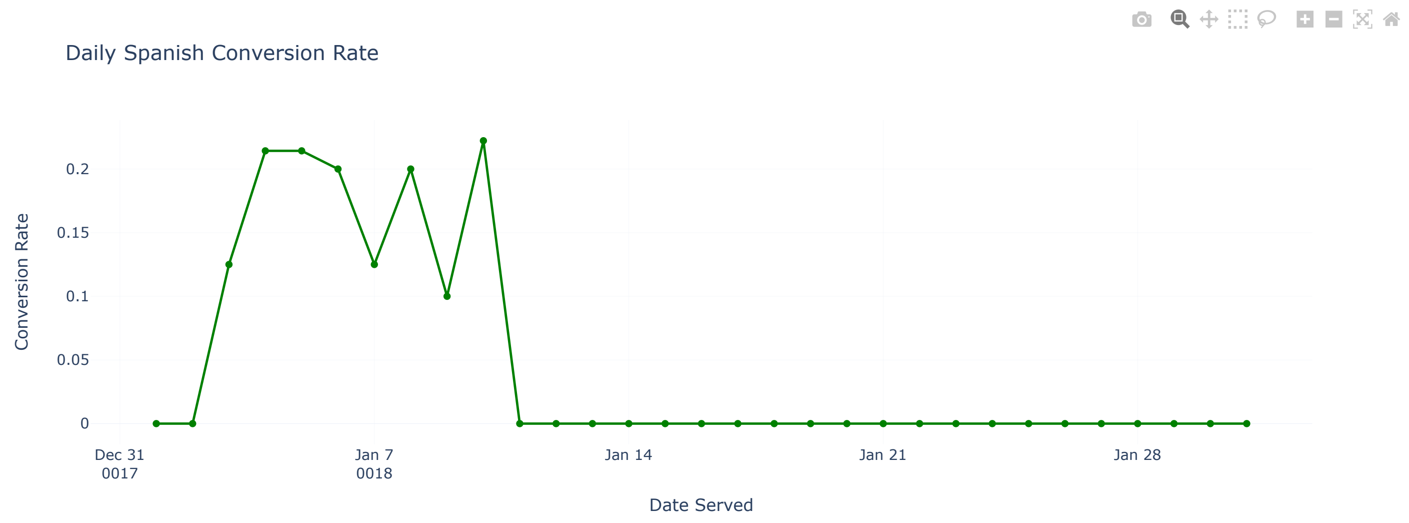








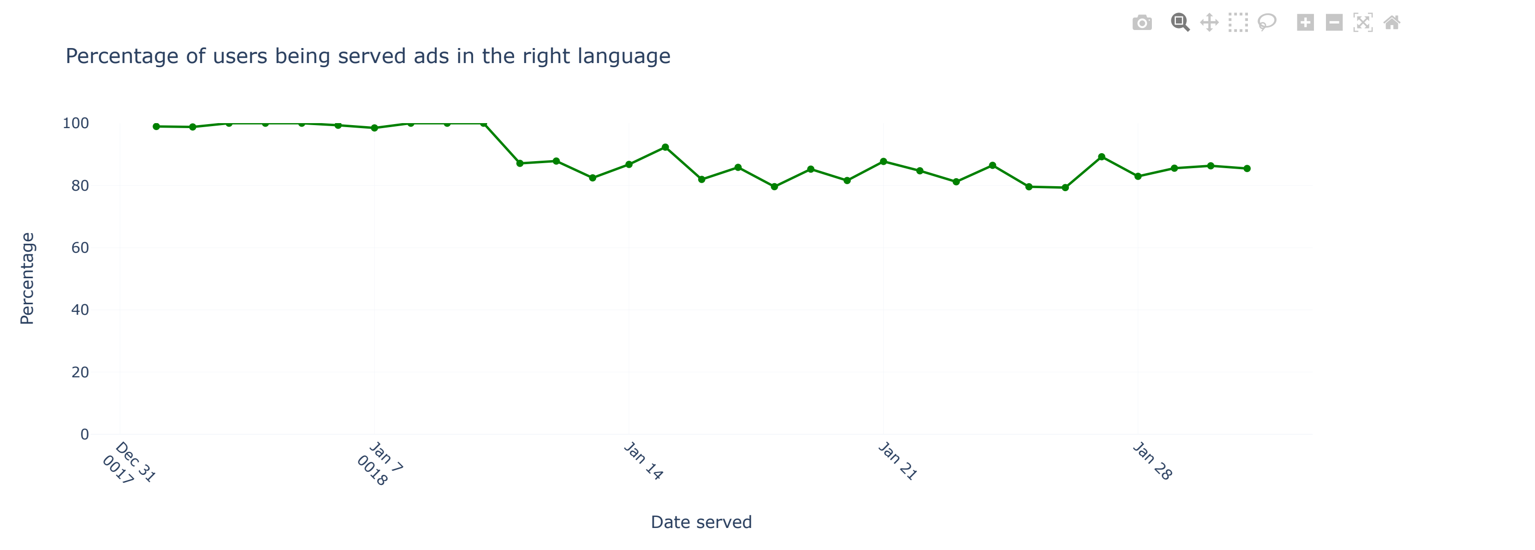
0.2
0
Dec 31
0017
Jan 7
0018
Jan 14
Jan 21
Jan 28
Date Served



Daily Spanish Conversion Rate
Conversion Rate
0.2
0.15
0.1
0.05
0
Dec 31
0017
Jan 7
0018
Jan 14
Jan 21
Jan 28
Date Served

```python
house_ads = house_ads.with_columns(pl.when(pl.col("language_displayed") ==
pl.col("language_preferred")).then(pl.lit('Yes')).otherwise(pl.lit('No')).alias("is_correct_lang"))
language_check = house_ads.group_by(['date_served','is_correct_lang']).len().sort(['date_served','is_correct_lang'])
language_check = language_check.pivot(columns='is_correct_lang',index='date_served',values='len')
row_sum = language_check.select(pl.sum_horizontal(pl.all().exclude('date_served').alias('row_sum')))
language_check_df = language_check.with_columns(row_sum)
language_check_df = language_check_df.with_columns((pl.col('Yes')/pl.col('row_sum')*100).alias('pct'))
print(language_check_df)

fig = px.line(
       language_check_df,
       x='date_served',
       y='pct',
       title='Percentage of users being served ads in the right language',
       template='plotly_white',
       labels={'pct': 'Percentage', 'date_served':'Date served'},
       markers=True
)
fig.update_traces(line_color='green')
fig.update_layout(
       xaxis=dict(tickangle=45),
       yaxis=dict(range=[0, 100]),
)
fig.show()
```

```
shape: (32, 5)
┌─────────────┬─────┬──────┬─────────┬───────────┐
│ date_served ┆ Yes ┆ No   ┆ row_sum ┆ pct       │
│ ---         ┆ --- ┆ ---  ┆ ---     ┆ ---       │
│ date        ┆ u32 ┆ u32  ┆ u32     ┆ f64       │
╞═════════════╪═════╪══════╪═════════╪═══════════╡
│ null        ┆ 1   ┆ null ┆ 1       ┆ 100.0     │
│ 0018-01-01  ┆ 189 ┆ 2    ┆ 191     ┆ 98.95288  │
│ 0018-01-02  ┆ 247 ┆ 3    ┆ 250     ┆ 98.8      │
│ 0018-01-03  ┆ 220 ┆ null ┆ 220     ┆ 100.0     │
│ 0018-01-04  ┆ 168 ┆ null ┆ 168     ┆ 100.0     │
│ …           ┆ …   ┆ …    ┆ …       ┆ …         │
│ 0018-01-27  ┆ 149 ┆ 18   ┆ 167     ┆ 89.221557 │
│ 0018-01-28  ┆ 136 ┆ 28   ┆ 164     ┆ 82.926829 │
│ 0018-01-29  ┆ 142 ┆ 24   ┆ 166     ┆ 85.542169 │
│ 0018-01-30  ┆ 145 ┆ 23   ┆ 168     ┆ 86.309524 │
│ 0018-01-31  ┆ 135 ┆ 23   ┆ 158     ┆ 85.443038 │
└─────────────┴─────┴──────┴─────────┴───────────┘
```



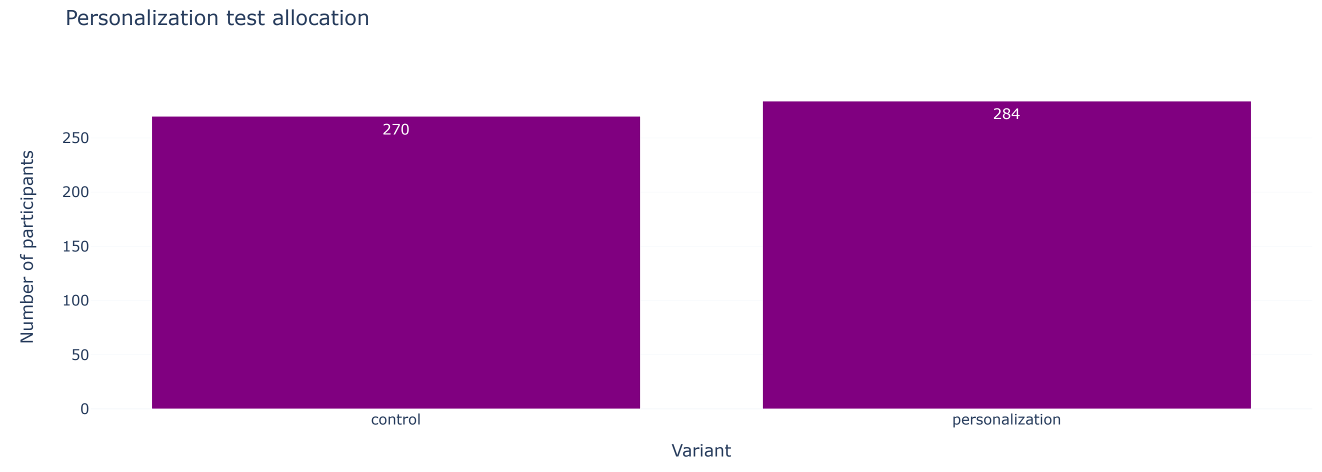
```python
house_ads_bug = house_ads.filter(house_ads['date_served'] < pl.datetime(2018, 1, 11).cast(pl.Date))
lang_conv_house_ads = conversion_rate(house_ads_bug, ['language_displayed'])
english_conv_rate = lang_conv_house_ads[['conversion_rate','language_displayed']].filter(pl.col('language_displayed')=='English')
lang_conv_house_ads = lang_conv_house_ads.with_columns((pl.col('conversion_rate')/english_conv_rate[0,0]).alias('conv_index_wrt_english'))
print(lang_conv_house_ads)
```

```
shape: (4, 5)
┌────────────────────┬─────────────────┬─────────────┬─────────────────┬────────────────────────┐
│ language_displayed ┆ users_converted ┆ users_total ┆ conversion_rate ┆ conv_index_wrt_english │
│ ---                ┆ ---             ┆ ---         ┆ ---             ┆ ---                    │
│ str                ┆ u32             ┆ u32         ┆ f64             ┆ f64                    │
╞════════════════════╪═════════════════╪═════════════╪═════════════════╪════════════════════════╡
│ Arabic             ┆ 7               ┆ 17          ┆ 0.411765        ┆ 6.0696                 │
│ German             ┆ 12              ┆ 27          ┆ 0.444444        ┆ 6.551315               │
│ Spanish            ┆ 17              ┆ 114         ┆ 0.149123        ┆ 2.198138               │
│ English            ┆ 262             ┆ 3862        ┆ 0.06784         ┆ 1.0                    │
└────────────────────┴─────────────────┴─────────────┴─────────────────┴────────────────────────┘
```

```python
converted = house_ads.group_by(['date_served', 'language_preferred']).agg([
    (pl.col('user_id').n_unique()).alias('user_num'),
    (pl.col('converted').sum()).alias('converted_num')
])
converted = converted.pivot(columns='language_preferred',index='date_served',values=['user_num','converted_num'])
print(converted)
```

```
shape: (32, 9)
┌──────────┬──────────┬──────────┬──────────┬───┬──────────┬──────────┬──────────┬──────────┐
│ date_serv ┆ user_num_ ┆ user_num_ ┆ user_num_ ┆ … ┆ converted ┆ converted ┆ converted ┆ converte │
│ ed       ┆ language_ ┆ language_ ┆ language_ ┆   ┆ _num_lang ┆ _num_lang ┆ _num_lang ┆ d_num_la │
│ ---      ┆ preferred ┆ preferred ┆ preferred ┆   ┆ uage_pref ┆ uage_pref ┆ uage_pref ┆ nguage_p │
│ date     ┆ _Ar…      ┆ _En…      ┆ _Ge…      ┆   ┆ err…      ┆ err…      ┆ err…      ┆ referr…  │
│          ┆ ---       ┆ ---       ┆ ---       ┆   ┆ ---       ┆ ---       ┆ ---       ┆ ---      │
│          ┆ u32       ┆ u32       ┆ u32       ┆   ┆ u32       ┆ u32       ┆ u32       ┆ u32      │
╞══════════╪══════════╪══════════╪══════════╪═══╪══════════╪══════════╪══════════╪══════════╡
│ 0018-01-1 ┆ 7         ┆ 127       ┆ 4         ┆ … ┆ 0         ┆ 6         ┆ 0         ┆ 0        │
│ 6        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-0 ┆ null      ┆ 143       ┆ 1         ┆ … ┆ null      ┆ 17        ┆ 0         ┆ 3        │
│ 5        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-3 ┆ 4         ┆ 139       ┆ 3         ┆ … ┆ 0         ┆ 4         ┆ 0         ┆ 2        │
│ 0        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-2 ┆ 3         ┆ 69        ┆ 4         ┆ … ┆ 0         ┆ 5         ┆ 0         ┆ 0        │
│ 3        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-1 ┆ 2         ┆ 189       ┆ 4         ┆ … ┆ 0         ┆ 6         ┆ 0         ┆ 0        │
│ 5        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ …        ┆ …         ┆ …         ┆ …         ┆ … ┆ …         ┆ …         ┆ …         ┆ …        │
│ 0018-01-1 ┆ 6         ┆ 121       ┆ 5         ┆ … ┆ 0         ┆ 5         ┆ 1         ┆ 0        │
│ 3        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-1 ┆ 7         ┆ 121       ┆ 6         ┆ … ┆ 0         ┆ 5         ┆ 1         ┆ 0        │
│ 8        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-2 ┆ 3         ┆ 75        ┆ 4         ┆ … ┆ 0         ┆ 4         ┆ 2         ┆ 0        │
│ 5        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ 0018-01-2 ┆ 5         ┆ 134       ┆ 3         ┆ … ┆ 0         ┆ 4         ┆ 0         ┆ 2        │
│ 8        ┆           ┆           ┆           ┆   ┆           ┆           ┆           ┆          │
│ null     ┆ null      ┆ 1         ┆ null      ┆ … ┆ null      ┆ 0         ┆ null      ┆ null     │
```

```
email = marketing.filter(pl.col('marketing_channel')=='Email')
alloc = email.group_by('variant').agg(pl.col('user_id').n_unique().alias('user_num'))
fig = px.bar(
    alloc,
    x='variant',
    y='user_num',
    color_discrete_sequence=['purple'],
    title='Personalization test allocation',
    template='plotly_white',
    text="user_num",
    labels={'variant': 'Variant', 'user_num': 'Number of participants'}
)
fig.show()
```


Personalization test allocation
Number of participants
control
personalization
Variant