```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('/content/netflix_customer_churn.csv')
```

```
df.drop('customer_id', axis=1, inplace=True)
```

```
df.head()
```

	age	gender	subscription_type	watch_hours	last_login_days	region	device	monthly_fee	churned	payment_method
0	51	Other	Basic	14.73	29	Africa	TV	8.99	1	Gift Card
1	47	Other	Standard	0.70	19	Europe	Mobile	13.99	1	Gift Card
2	27	Female	Standard	16.32	10	Asia	TV	13.99	0	Crypto
3	53	Other	Premium	4.51	12	Oceania	TV	17.99	1	Crypto
4	56	Other	Standard	1.89	13	Africa	Mobile	13.99	1	Crypto

```
df.isna().sum()
```

	0
customer_id	0
age	0
gender	0
subscription_type	0
watch_hours	0
last_login_days	0
region	0
device	0
monthly_fee	0
churned	0
payment_method	0
number_of_profiles	0
avg_watch_time_per_day	0
favorite_genre	0

dtype: int64

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   customer_id             5000 non-null   object
 1   age                     5000 non-null   int64
 2   gender                  5000 non-null   object
 3   subscription_type       5000 non-null   object
 4   watch_hours             5000 non-null   float64
 5   last_login_days         5000 non-null   int64
 6   region                  5000 non-null   object
 7   device                  5000 non-null   object
 8   monthly_fee             5000 non-null   float64
 9   churned                 5000 non-null   int64
 10  payment_method          5000 non-null   object
 11  number_of_profiles      5000 non-null   int64
 12  avg_watch_time_per_day  5000 non-null   float
 13  favorite_genre          5000 non-null   objec
dtypes: float64(3), int64(4), object(7)
memory usage: 547.0+ KB
```

```
df.describe()
```

	age	watch_hours	last_login_days	monthly_fee	churned	number_of_profiles	avg_watch_time_per_day
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	43.847400	11.649450	30.089800	13.683400	0.503000	3.024400	0.874800
std	15.501128	12.014654	17.536078	3.692062	0.500041	1.415841	2.619824
min	18.000000	0.010000	0.000000	8.990000	0.000000	1.000000	0.000000
25%	30.000000	3.337500	15.000000	8.990000	0.000000	2.000000	0.110000
50%	44.000000	8.000000	30.000000	13.990000	1.000000	3.000000	0.290000
75%	58.000000	16.030000	45.000000	17.990000	1.000000	4.000000	0.720000
max	70.000000	110.400000	60.000000	17.990000	1.000000	5.000000	98.420000

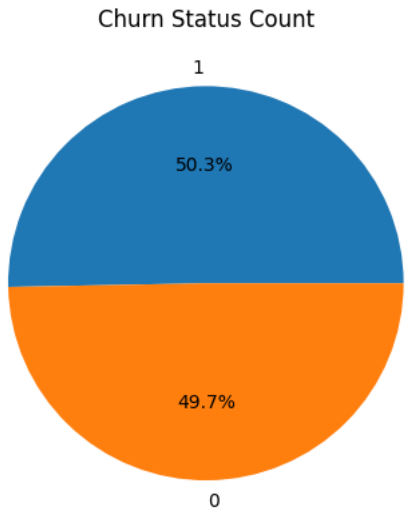
```
df.loc[df['churned'] == 'No', 'churned'] = 0
df.loc[df['churned'] == 'Yes', 'churned'] = 1
```

```
df['churned'].value_counts()
```

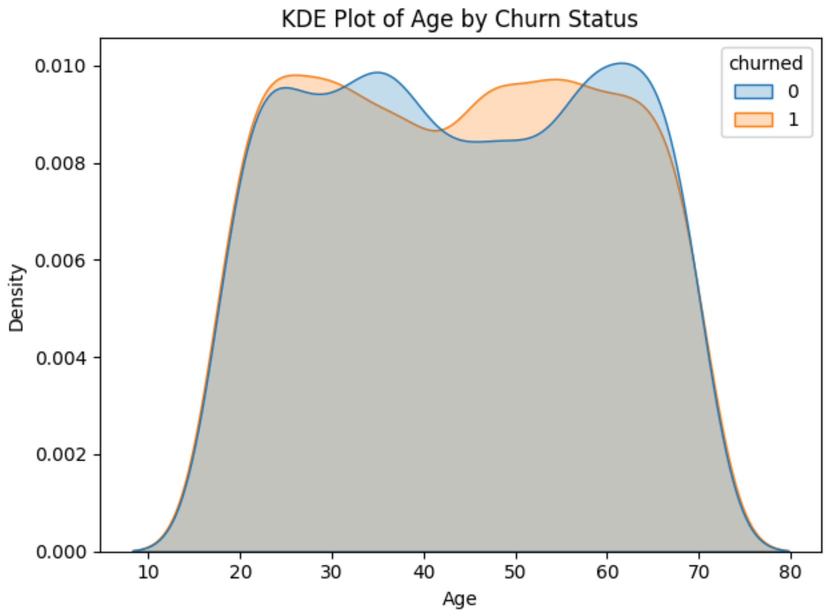
churned	count
1	2515
0	2485

dtype: int64

```
plt.pie(df['churned'].value_counts(), labels=df['churned'].value_counts().index, autopct='%1.1f%%')
plt.title('Churn Status Count')
plt.show()
```




```
sns.kdeplot(data=df, x='age', hue='churned', fill=True)
plt.title(f'KDE Plot of Age by Churn Status')
plt.xlabel('Age')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```

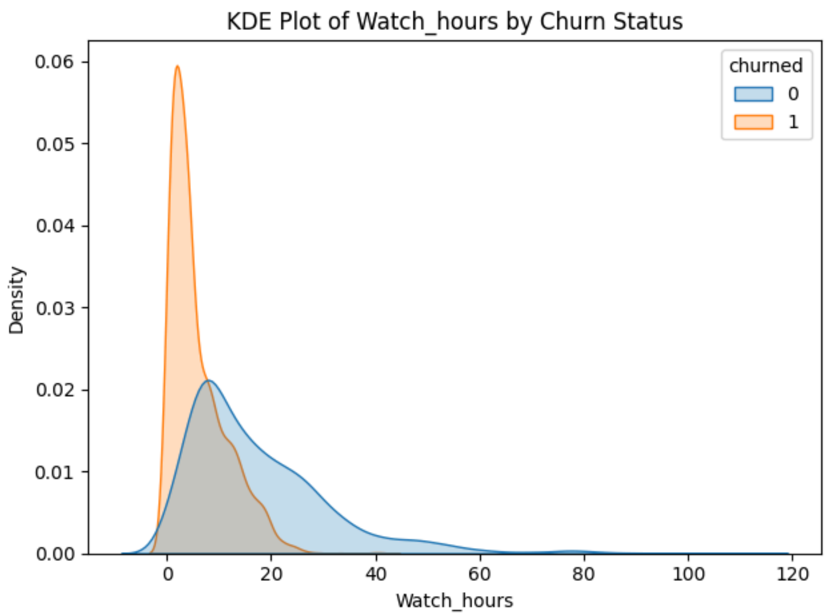


KDE Plot of Age by Churn Status

Key Insights:

- The KDE curves for both churned and non-churned customers follow a nearly identical distribution across ages, indicating that age alone does not strongly differentiate churn behaviour.
- Both groups show a peak in density around the early 30s and again in the late 50s to early 60s, suggesting higher customer concentrations in those age ranges.
- The churned population shows slightly higher density in the 50–60 age range, while the non-churned population has a minor peak around the mid-30s

```
sns.kdeplot(data=df, x='watch_hours', hue='churned', fill=True)
plt.title(f'KDE Plot of Watch_hours by Churn Status')
plt.xlabel('Watch_hours')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```



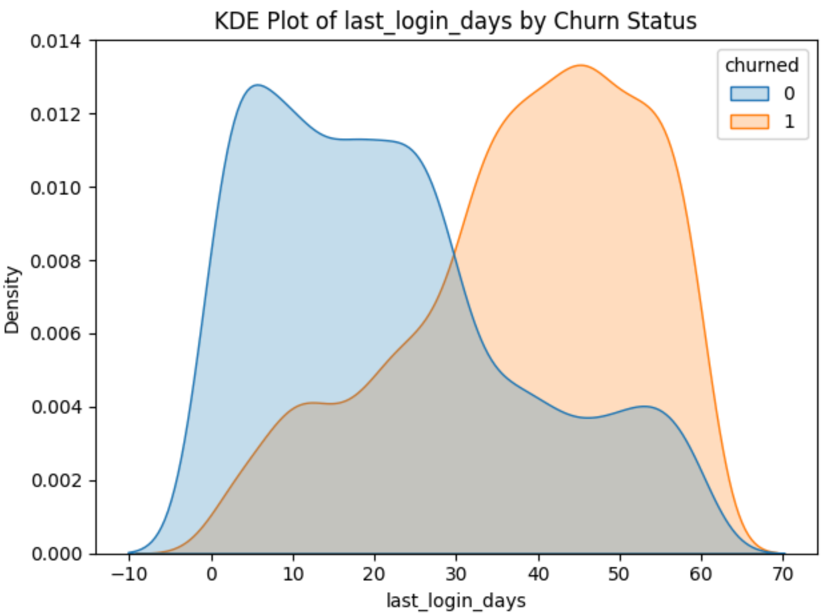
KDE Plot of Watch Hours by Churn Status

Key Insights:

- Churned customers have a sharp density peak at low watch hours (0–5 hours), indicating that the majority of customers who churn engage very little with the service.
- Non-churned customers show a broader distribution, with the peak between 5 and 15 hours, and a notable tail extending beyond 40+ hours, reflecting higher and more varied engagement.

- There is a strong correlation between lower watch hours and churn. A significantly higher number of users who churned watched fewer hours.
- The overall trend is active users with more watch time are less likely to churn.

```
sns.kdeplot(data=df, x='last_login_days', hue='churned', fill=True)
plt.title(f'KDE Plot of last_login_days by Churn Status')
plt.xlabel('last_login_days')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```



KDE Plot of Last login days by Churn Status

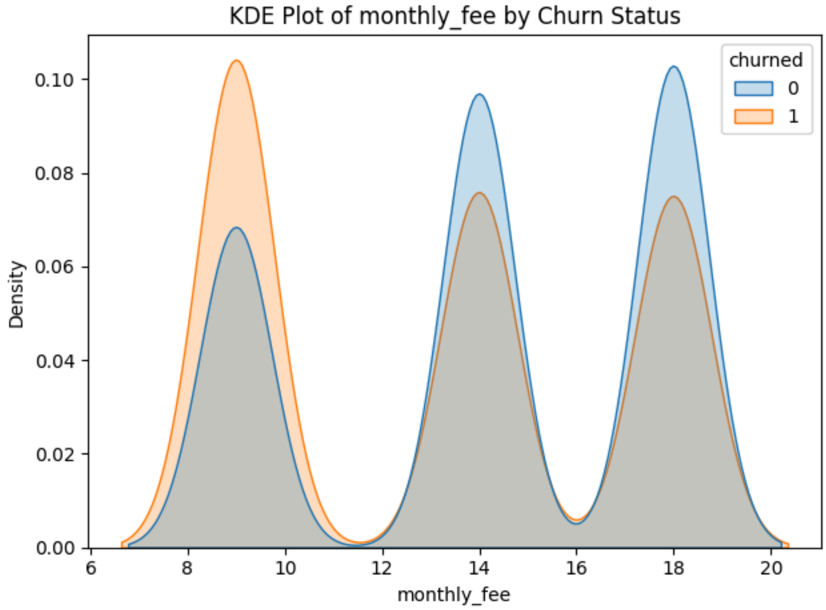
Key Insights:

- Recent activity strongly correlates with retention. Customers who logged in within the past 20 days are much less likely to churn.
- Inactivity beyond ~30 days is a critical churn indicator. After this threshold, churn probability increases significantly.

Recommendation:

- Customers in the 20–35 day range represent a key target group for re-engagement strategies (e.g., personalised offers, reminders, or support outreach).

```
sns.kdeplot(data=df, x='monthly_fee', hue='churned', fill=True)
plt.title(f'KDE Plot of monthly_fee by Churn Status')
plt.xlabel('monthly_fee')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```



KDE Plot of Monthly Fee Distribution by Churn Status

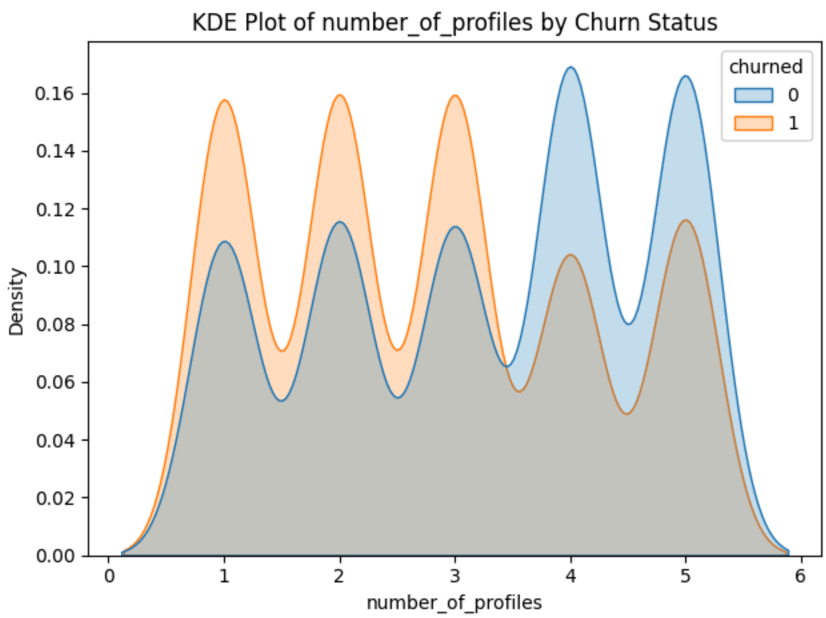
Key Insights:

- At the lowest fee tier, customers who churned show a higher density than retained customers, suggesting that low-paying customers are more likely to leave.
- At mid to high fee tier, retained customers dominate, indicating stronger customer loyalty in higher pricing tiers.

Recommendation:

- Strategies like bundled benefits, targeted loyalty programs, or tier migration incentives could help retain price-sensitive customers.
- Higher-paying customers appear less prone to churn, suggesting opportunities to upsell lower-tier customers into higher-value plans.

```
sns.kdeplot(data=df, x='number_of_profiles', hue='churned', fill=True)
plt.title(f'KDE Plot of number_of_profiles by Churn Status')
plt.xlabel('number_of_profiles')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```



KDE Plot of Number of profiles by Churn Status

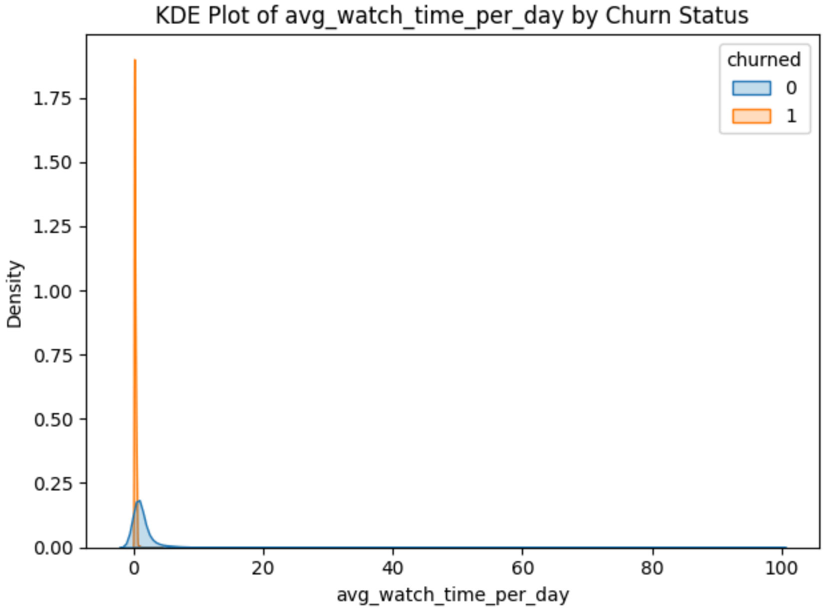
Key Insights:

- Fewer profiles (1–3) = Higher churn risk.
- More profiles (4–5) = Lower churn risk, likely due to stronger household/family engagement or greater perceived value.

Recommendation:

- Encouraging customers to add more profiles (e.g., for family sharing or personalised recommendations) could increase retention.

```
sns.kdeplot(data=df, x='avg_watch_time_per_day', hue='churned', fill=True)
plt.title(f'KDE Plot of avg_watch_time_per_day by Churn Status')
plt.xlabel('avg_watch_time_per_day')
plt.ylabel('Density')
plt.tight_layout()
plt.show()
```



KDE Plot of Average Watch Time per Day by Churn Status

Key Insights:

- Low watch time (0–1 hour/day) = Strongly associated with churn.
- Moderate to high watch time (>2 hours/day) = Strongly associated with retention.

Recommendation:

- Encourage low-engagement users to consume more content through personalised recommendations, or reminders.
- Highlight popular or trending content to increase watch time and reduce churn risk.

Summary Of Important findings derived from KDE Plots

Skewed Features:

- KDE plots effectively highlight skewness, which is the asymmetry of a distribution and helps in understanding whether a dataset violates the assumption of normality, which is crucial for many statistical models.

Heavy tails and kurtosis:

- A distribution with heavy tails means that there is a greater probability of extreme values occurring compared to a normal distribution. This is important for tasks like risk assessment, as it suggests that extreme events are more common than one might assume.

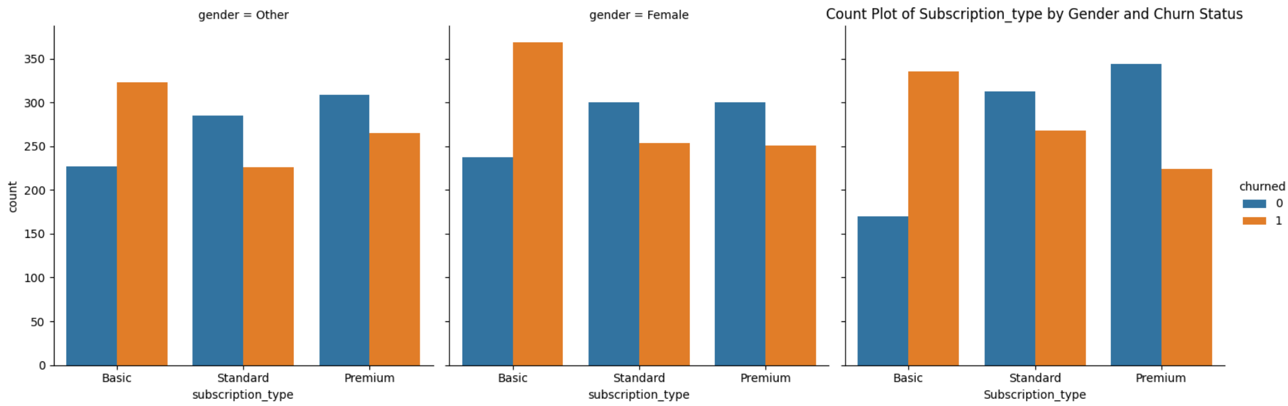
Non-linear patterns:

- Bivariate KDE plots shows non-linear relationships between two variables by depicting clustered "hills" or "contours" of higher density that don't follow a simple straight line. This can indicate that a linear model would be insufficient and that a more complex, non-linear approach is needed.

Important but extreme outliers:

- KDE plots expose important but extreme outliers, while these can sometimes be data entry errors, they can also represent significant and meaningful data points prompting a closer look rather than a simple removal

```
sns.catplot(data=df, x='subscription_type', hue='churned', col='gender', kind='count')
plt.title(f'Count Plot of Subscription_type by Gender and Churn Status')
plt.xlabel('Subscription_type')
plt.ylabel('Count')
plt.show()
```

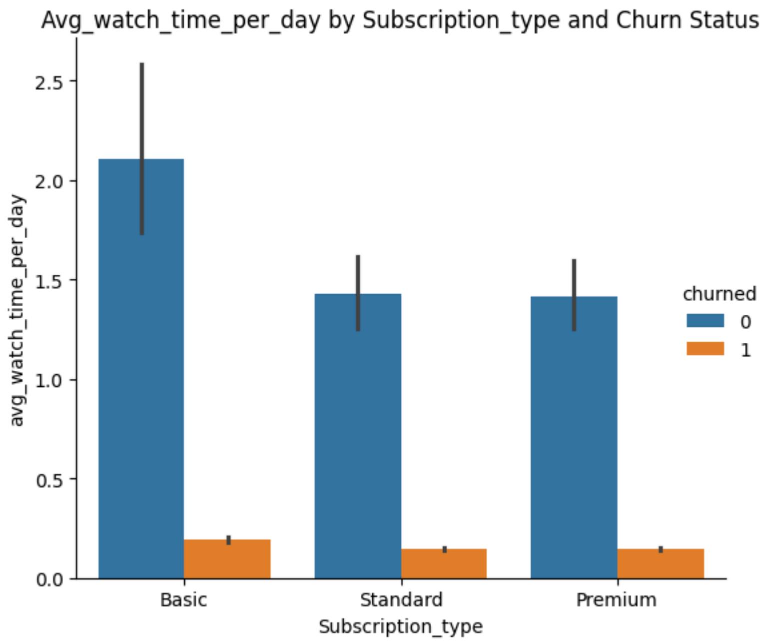


Count Plot of Subscription_type by Gender and Churn Status

Key Insights:

- Premium subscription retains users best, regardless of gender.
- Basic subscription has the highest churn suggesting a need for improvement in that tier

```
sns.catplot(data=df, x='subscription_type', y='avg_watch_time_per_day', hue='churned', kind='bar')
plt.title(f'Avg_watch_time_per_day by Subscription_type and Churn Status')
plt.xlabel('Subscription_type')
plt.ylabel('avg_watch_time_per_day')
plt.tight_layout()
plt.show()
```

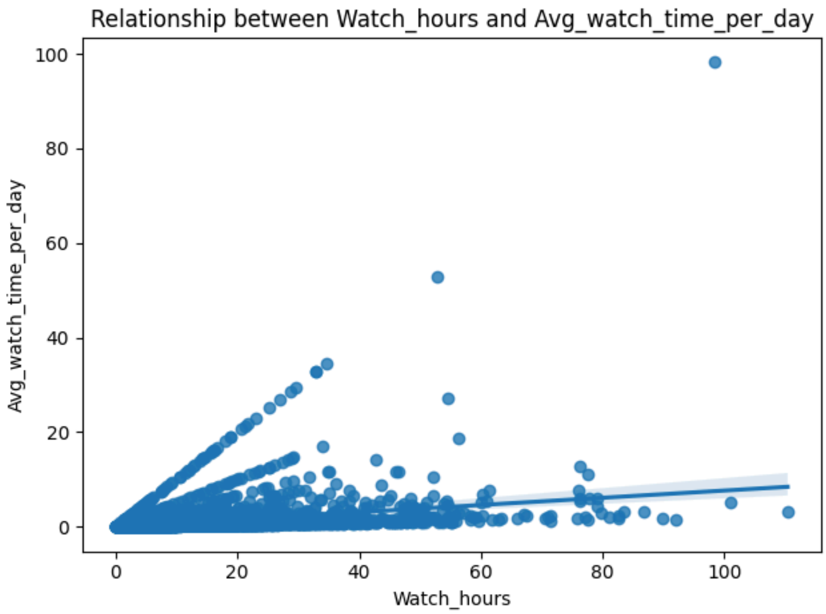


Avg_watch_time_per_day by Subscription_type and Churn Status

Key Insights:

- The chart suggests a strong inverse relationship between watch time and churn: users who spend more time watching are less likely to churn, approximately 1.6 hours per day.
- The average daily watch time for users who have churned is very low, around 0.2 hours per day.
- This highlights that engagement is a crucial factor in customer retention regardless of their subscription plan.

```
sns.regplot(data=df, x='watch_hours', y='avg_watch_time_per_day')
plt.title(f'Relationship between Watch_hours and Avg_watch_time_per_day')
plt.xlabel('Watch_hours')
plt.ylabel('Avg_watch_time_per_day')
plt.tight_layout()
plt.show()
```

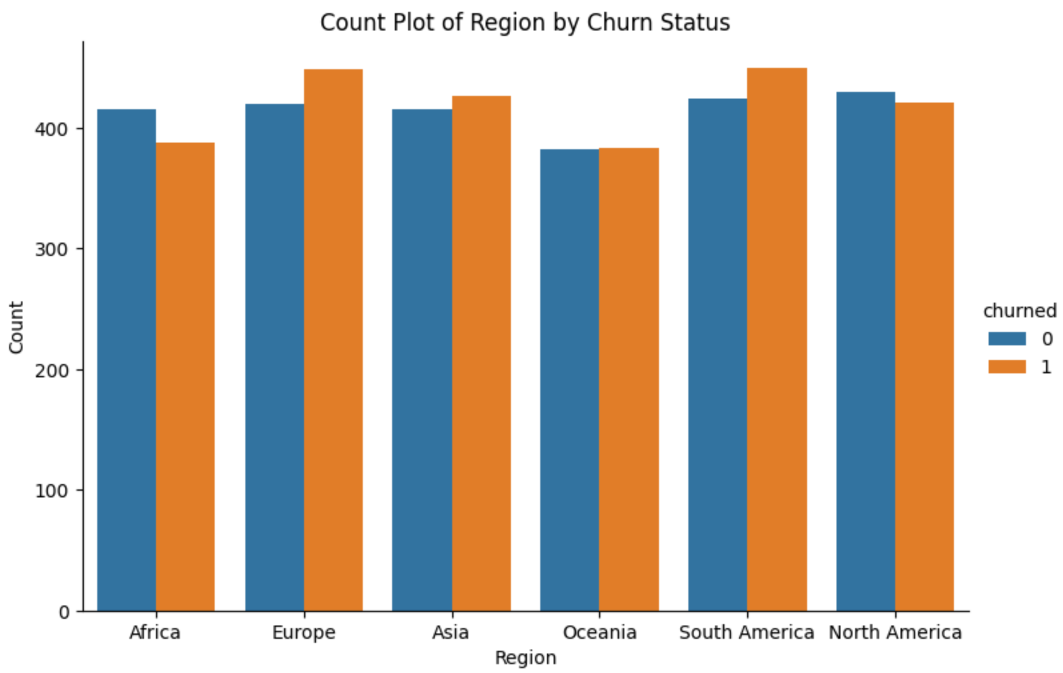


Relplot of Watch_hours by Avg_watch_time_per_day

Key Insights:

- most users have low monthly watch hours and a low average daily watch time.
- There's a clear positive linear relationship between the two variables this means that as a user's total monthly watch hours increase, their average daily watch time also tends to increase
- The spread of the data points shows that while there's a general trend, individual user behavior can vary significantly. For example, some users have a high number of monthly watch hours but a lower average daily watch time, likely due to watching less consistently over a longer period.

```
sns.catplot(data=df, x='region', hue='churned', kind='count', height=5, aspect=1.5)
plt.title(f'Count Plot of Region by Churn Status')
plt.xlabel('Region')
plt.ylabel('Count')
plt.show()
```

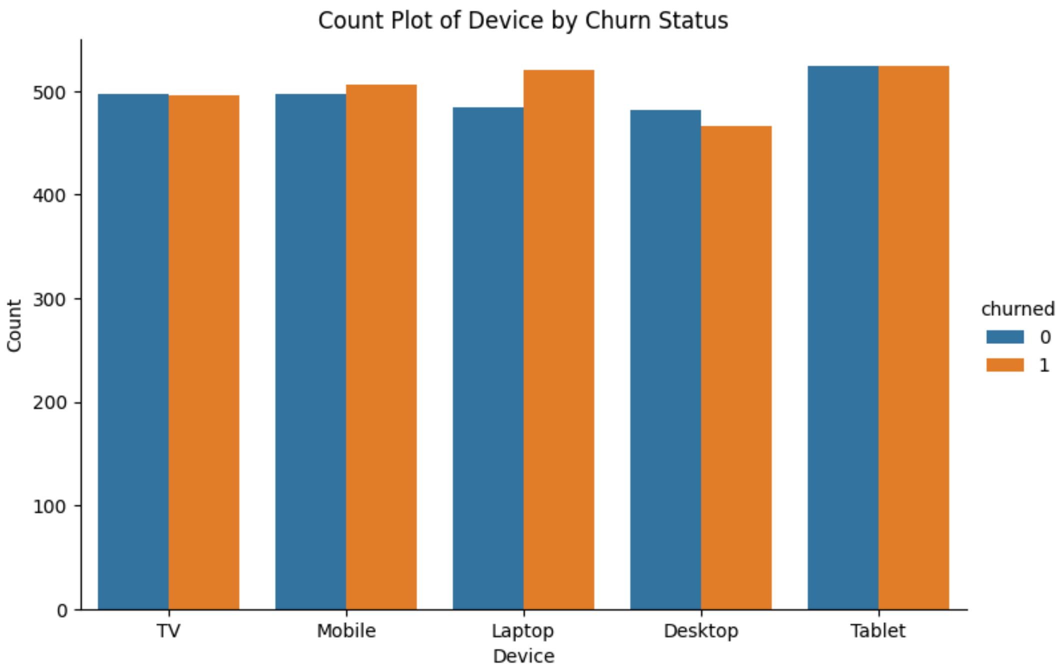


Count Plot of Region by Churn Status

Key Insights:

- The total number of users (churned and not churned) is roughly similar across most regions, with counts generally ranging between 375 and 450.
- For most regions—Europe, Asia, South America, and North America—the count of churned users is slightly higher than or comparable to the count of non-churned users. This suggests a relatively high churn rate in these areas.
- In Africa and Oceania, the counts of non-churned users are slightly higher than those who churned, indicating potentially better retention in these regions compared to others.

```
sns.catplot(data=df, x='device', hue='churned', kind='count', height=5, aspect=1.5)
plt.title(f'Count Plot of Device by Churn Status')
plt.xlabel('Device')
plt.ylabel('Count')
plt.show()
```

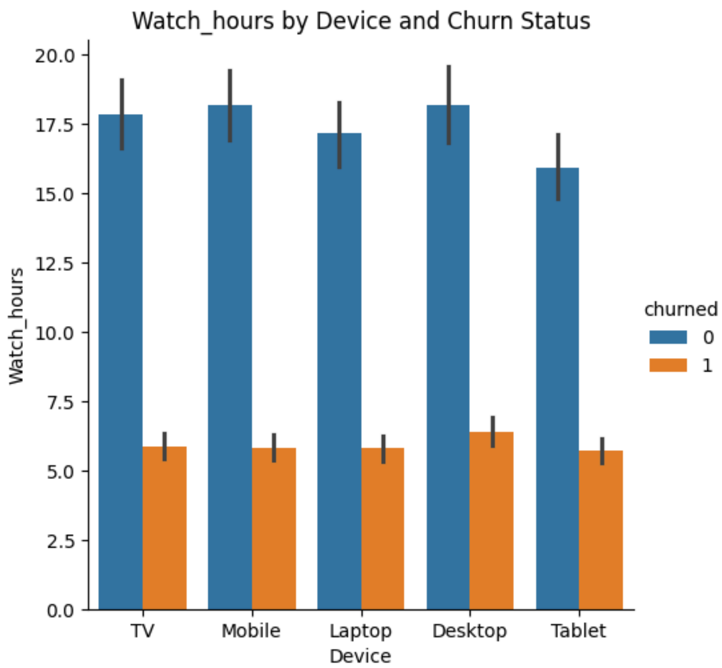


Count Plot of Device by Churn Status

Key Insights:

- the data suggests that device type does not appear to be a major factor in predicting customer churn, as the counts for churned and non-churned customers are very similar across most device categories. However, tablets show the most notable difference, with a slightly higher tendency for churn.

```
sns.catplot(data=df, x='device', y='watch_hours', hue='churned',  kind='bar')
plt.title(f'Watch_hours by Device and Churn Status')
plt.xlabel('Device')
plt.ylabel('Watch_hours')
plt.show()
```

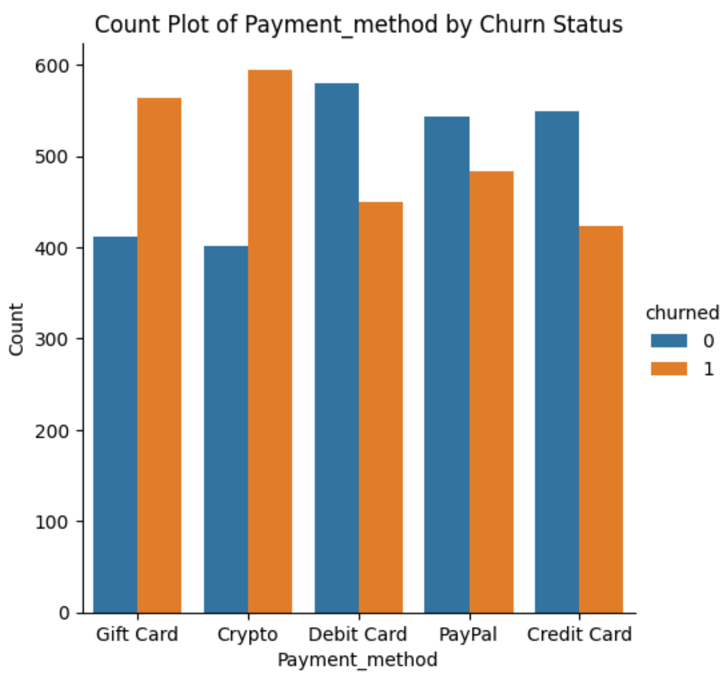


Watch_hours by Device and Churn Status

Key Insights:

- Across all device types, there is a consistent and significant trend: users who have not churned have a much higher average of monthly watch hours than those who have churned.

```
sns.catplot(data=df, x='payment_method', hue='churned', kind='count')
plt.title(f'Count Plot of Payment_method by Churn Status')
plt.xlabel('Payment_method')
plt.ylabel('Count')
plt.show()
```

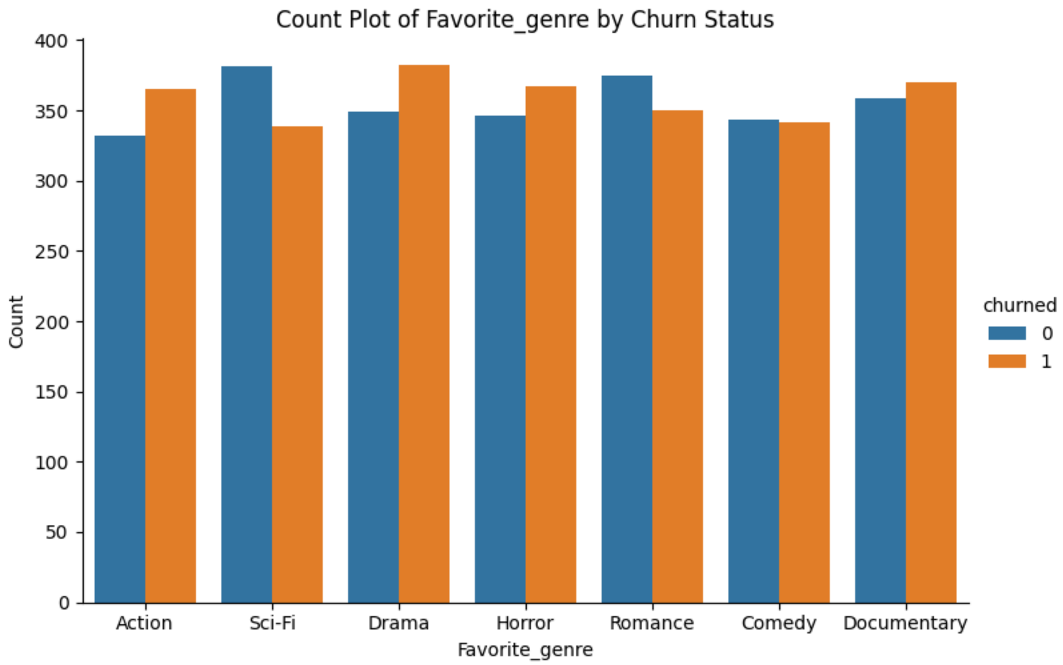


Count Plot of Payment_method by Churn Status

Key Insights:

- Crypto and Gift Card users have a significantly higher churn rate. This suggests that these payment methods are associated with a greater likelihood of a user not renewing their subscription.
- Credit Card, Debit Card, and PayPal users show the opposite trend, with the count of non-churned users being higher than churned users. This indicates that these traditional and widely-used payment methods are linked to better user retention.

```
sns.catplot(data=df, x='favorite_genre', hue='churned', kind='count', height=5, aspect=1.5)
plt.title(f'Count Plot of Favorite_genre by Churn Status')
plt.xlabel('Favorite_genre')
plt.ylabel('Count')
plt.show()
```

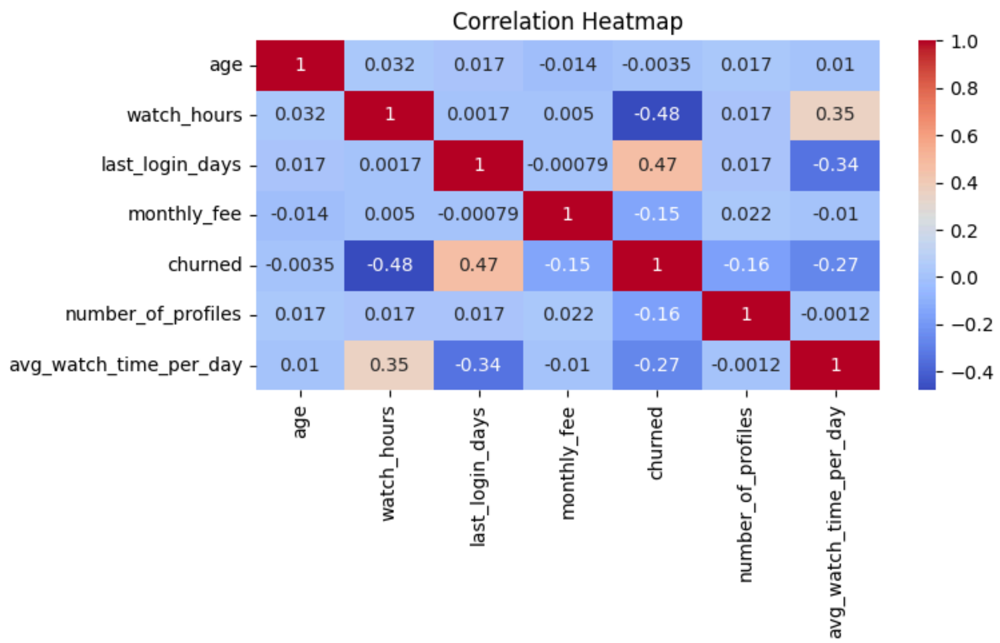


Count Plot of Favorite_genre by Churn Status

Key Insights:

- the chart suggests that while genre preference doesn't cause a massive difference in churn, users whose favorite genres are Sci-Fi or Romance may be slightly more likely to remain subscribed.

```
fig, ax = plt.subplots(figsize=(8, 5))
sns.heatmap(df[['age', 'watch_hours', 'last_login_days', 'monthly_fee', 'churned',
       'number_of_profiles', 'avg_watch_time_per_day']].corr(), annot=True, cmap='coolwarm', ax=ax)
plt.title('Correlation Heatmap')
plt.tight_layout()
plt.show()
```



Correlation Heatmap

Key Insights:

- There is a strong negative correlation between churned and watch_hours (-0.48). This is the strongest relationship shown on the map, indicating that as monthly watch hours decrease, the likelihood of a user churning significantly increases.
- There is a strong positive correlation between churned and last_login_days (0.47). This indicates that as the number of days since a user's last login increases, the probability of them churning also increases.
- There's a strong positive correlation between watch_hours and avg_watch_time_per_day (0.35), which is expected as these are two measures of user engagement.
- The correlation between churned and number_of_profiles is a weak negative (-0.16), suggesting that a higher number of profiles on an account might slightly decrease the chance of churn, but the relationship is not very strong.
- All other correlations are very weak (close to zero), indicating that variables like age and monthly_fee have little to no linear relationship with other variables in this dataset.