In [1]:
```
import pandas as pd
import numpy as np
```

1. Load and Preview each CSV

In [3]:
```
# Declare root path to the files
path = "../data/"
```

In [4]:
```
# Load accounts
accounts = pd.read_csv(f"{path}accounts.csv")
```

In [5]:
```
# Display 5 first rows
accounts.head()
```

Out[5]:

	account_id	account_name	industry	country	signup_date	referral_source	plan_ti
0	A-2e4581	Company_0	EdTech	US	2024-10-16	partner	Ba
1	A-43a9e3	Company_1	FinTech	IN	2023-08-17	other	Ba
2	A-0a282f	Company_2	DevTools	US	2024-08-27	organic	Ba
3	A-1f0ac7	Company_3	HealthTech	UK	2023-08-27	other	Ba
4	A-ce550d	Company_4	HealthTech	US	2024-10-27	event	Enterpri

In [6]:
```
# Display column names, count of non-null values and data types
accounts.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   account_id       500 non-null    object
 1   account_name     500 non-null    object
 2   industry         500 non-null    object
 3   country          500 non-null    object
 4   signup_date      500 non-null    object
 5   referral_source  500 non-null    object
 6   plan_tier        500 non-null    object
 7   seats            500 non-null    int64 
 8   is_trial         500 non-null    bool  
 9   churn_flag       500 non-null    bool  
dtypes: bool(2), int64(1), object(7)
memory usage: 32.4+ KB
```

In [7]:
```
# Load churn_events
churn = pd.read_csv(f"{path}churn_events.csv")
```

In [8]:
```
# Display 5 first rows
churn.head()
```

Out[8]:

	churn_event_id	account_id	churn_date	reason_code	refund_amount_usd	preceding_
0	C-816288	A-c37cab	2024-10-27	pricing	4.03	
1	C-5a81e7	A-37f969	2024-06-25	support	96.45	
2	C-a174be	A-b07346	2024-11-12	budget	0.00	
3	C-accb39	A-1e50e0	2023-11-01	budget	54.94	
4	C-92f889	A-956988	2024-12-30	unknown	0.00	

In [9]:
```
# Display column names, count of non-null values and data types
churn.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 600 entries, 0 to 599
Data columns (total 9 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   churn_event_id           600 non-null    object
 1   account_id               600 non-null    object
 2   churn_date               600 non-null    object
 3   reason_code              600 non-null    object
 4   refund_amount_usd        600 non-null    float64
 5   preceding_upgrade_flag   600 non-null    bool
 6   preceding_downgrade_flag 600 non-null    bool
 7   is_reactivation          600 non-null    bool
 8   feedback_text            452 non-null    object
dtypes: bool(3), float64(1), object(5)
memory usage: 30.0+ KB
```

In [10]:
```
# Load feature_usage
feature_usage = pd.read_csv(f"{path}feature_usage.csv")
```

In [11]:
```
# Display 5 first rows
feature_usage.head()
```

Out[11]:

	usage_id	subscription_id	usage_date	feature_name	usage_count	usage_duration_se
0	U-1c6c24	S-0fcf7d	27/7/2023	feature_20	9	50
1	U-f07cb8	S-c25263	7/8/2023	feature_5	9	3
2	U-096807	S-f29e7f	7/12/2023	feature_3	9	14
3	U-6b1580	S-be655e	28/7/2024	feature_40	5	20
4	U-720a29	S-f9b1d0	2/12/2024	feature_12	12	9

In [12]:
```
# Display column names, count of non-null values and data types
feature_usage.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24979 entries, 0 to 24978
Data columns (total 8 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   usage_id             24979 non-null  object
 1   subscription_id      24979 non-null  object
 2   usage_date           24979 non-null  object
 3   feature_name         24979 non-null  object
 4   usage_count          24979 non-null  int64 
 5   usage_duration_secs  24979 non-null  int64 
 6   error_count          24979 non-null  int64 
 7   is_beta_feature      24979 non-null  bool  
dtypes: bool(1), int64(3), object(4)
memory usage: 1.4+ MB
```

In [13]:
```
# Load subscriptions
subscriptions = pd.read_csv(f"{path}subscriptions.csv")
```

In [14]:
```
# Display 5 first rows
subscriptions.head()
```

Out[14]:

	subscription_id	account_id	start_date	end_date	plan_tier	seats	mrr_amount	arr_
0	S-8cec59	A-3c1a3f	2023-12-23	2024-04-12	Enterprise	14	2786	
1	S-0f6f44	A-9b9fe9	2024-06-11	NaN	Pro	17	833	
2	S-51c0d1	A-659280	2024-11-25	NaN	Enterprise	62	0	
3	S-f81687	A-e7a1e2	2024-11-23	2024-12-13	Enterprise	5	995	
4	S-cff5a2	A-ba6516	2024-01-10	NaN	Enterprise	27	5373	

In [15]:
```
# Display column names, count of non-null values and data types
subscriptions.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   subscription_id    5000 non-null   object
 1   account_id         5000 non-null   object
 2   start_date         5000 non-null   object
 3   end_date           486 non-null    object
 4   plan_tier          5000 non-null   object
 5   seats              5000 non-null   int64 
 6   mrr_amount         5000 non-null   int64 
 7   arr_amount         5000 non-null   int64 
 8   is_trial           5000 non-null   bool  
 9   upgrade_flag       5000 non-null   bool  
 10  downgrade_flag     5000 non-null   bool  
 11  churn_flag         5000 non-null   bool  
 12  billing_frequency  5000 non-null   object
 13  auto_renew_flag    5000 non-null   bool  
dtypes: bool(5), int64(3), object(6)
memory usage: 376.1+ KB
```

In [16]:
```python
# Load support_tickets
support_tickets = pd.read_csv(f"{path}support_tickets.csv")
```

In [17]:
```python
# Display 5 first rows
support_tickets.head()
```

Out[17]:

	ticket_id	account_id	submitted_at	closed_at	resolution_time_hours	priority	first_r
0	T-0024de	A-712f1c	2023-07-27	2023-07-28 03:00:00	27.0	high	
1	T-4d04b9	A-e43bf7	2024-07-08	2024-07-09 03:00:00	27.0	urgent	
2	T-d5e12f	A-0f3e88	2024-10-17	2024-10-17 19:00:00	19.0	urgent	
3	T-dfce9a	A-4c56c9	2024-09-08	2024-09-09 23:00:00	47.0	medium	
4	T-c59f77	A-6f8ad2	2024-11-30	2024-12-01 02:00:00	26.0	medium	

In [18]:
```python
# Display column names, count of non-null values and data types
support_tickets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 9 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   ticket_id                    2000 non-null   object
 1   account_id                   2000 non-null   object
 2   submitted_at                 2000 non-null   object
 3   closed_at                    2000 non-null   object
 4   resolution_time_hours        2000 non-null   float64
 5   priority                     2000 non-null   object
 6   first_response_time_minutes  2000 non-null   int64
 7   satisfaction_score           1175 non-null   float64
 8   escalation_flag              2000 non-null   bool
dtypes: bool(1), float64(2), int64(1), object(5)
memory usage: 127.1+ KB
```

2. Check for nulls

In [20]:
```
# Check for nulls in accounts
print(accounts.isnull().sum())
```

```
account_id         0
account_name       0
industry           0
country            0
signup_date        0
referral_source    0
plan_tier          0
seats              0
is_trial           0
churn_flag         0
dtype: int64
```

In [21]:
```
# Check for nulls in churn
print(churn.isnull().sum())
```

```
churn_event_id          0
account_id              0
churn_date              0
reason_code             0
refund_amount_usd       0
preceding_upgrade_flag  0
preceding_downgrade_flag  0
is_reactivation         0
feedback_text           148
dtype: int64
```

In [22]:
```
# Check for nulls in feature_usage
print(feature_usage.isnull().sum())
```

```
usage_id               0
subscription_id        0
usage_date             0
feature_name           0
usage_count            0
usage_duration_secs    0
error_count            0
is_beta_feature        0
dtype: int64
```

In [23]:
```
# Check for nulls in subscriptions
print(subscriptions.isnull().sum())
```

```
subscription_id         0
account_id              0
start_date              0
end_date             4514
plan_tier               0
seats                   0
mrr_amount              0
arr_amount              0
is_trial                0
upgrade_flag            0
downgrade_flag          0
churn_flag              0
billing_frequency       0
auto_renew_flag         0
dtype: int64
```

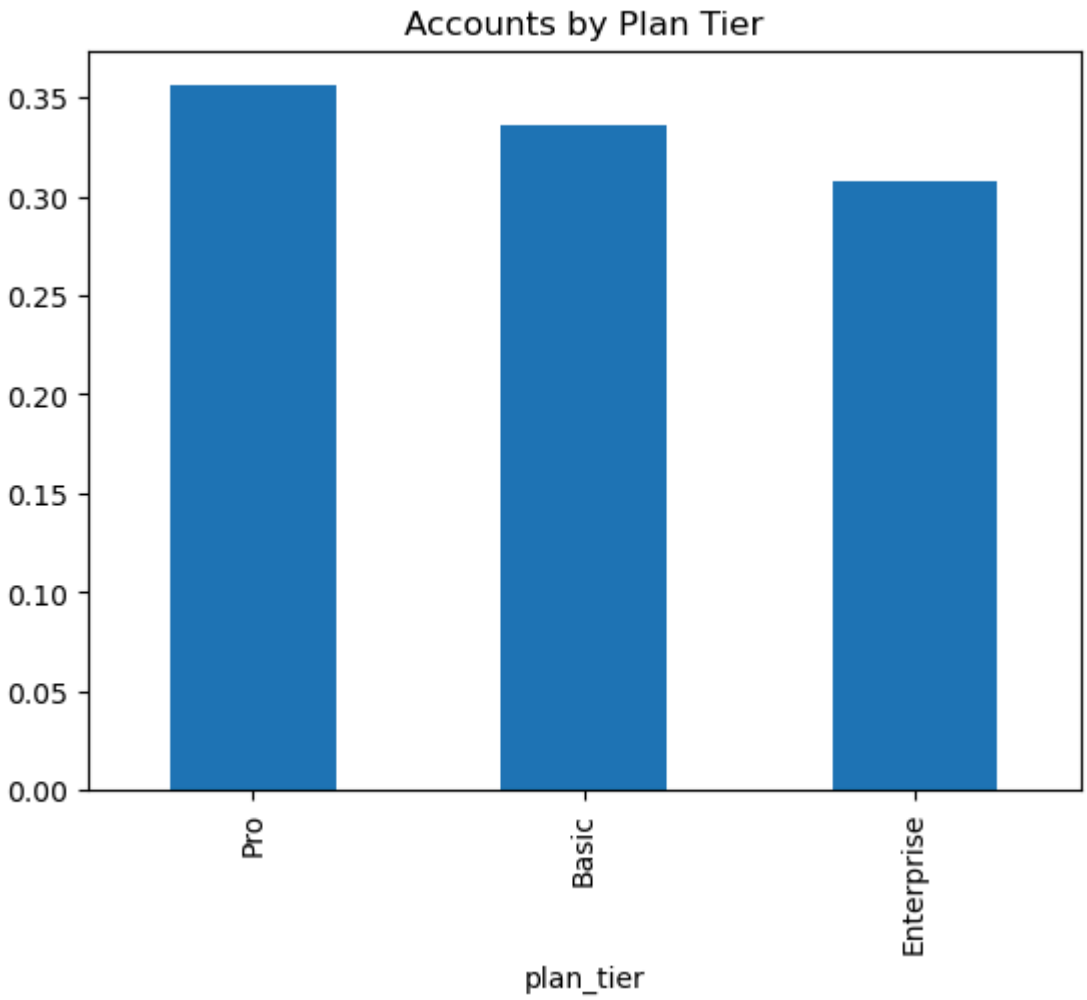
In [24]:
```
# Check for nulls in support_tickets
print(support_tickets.isnull().sum())
```

```
ticket_id                       0
account_id                      0
submitted_at                    0
closed_at                       0
resolution_time_hours           0
priority                        0
first_response_time_minutes     0
satisfaction_score            825
escalation_flag                 0
dtype: int64
```

3. Basic Statistics and Distributions

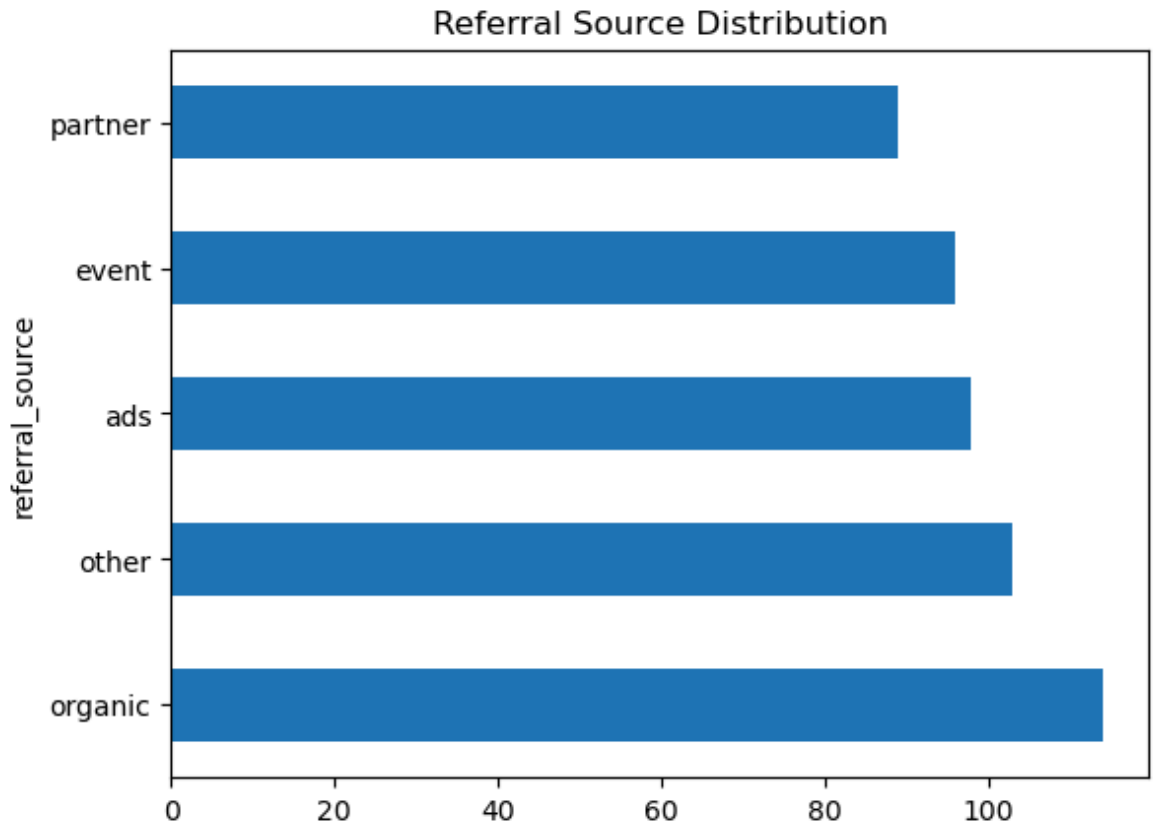
In [26]:
```
# Accounts by plan
accounts['plan_tier'].value_counts(normalize=True).plot(kind='bar', title='Accou
```

Out[26]:
```
<Axes: title={'center': 'Accounts by Plan Tier'}, xlabel='plan_tier'>
```



In [27]:
```
# Referral sources
accounts['referral_source'].value_counts().plot(kind='barh', title='Referral Sou
```

Out[27]:
```
<Axes: title={'center': 'Referral Source Distribution'}, ylabel='referral_sourc
e'>
```

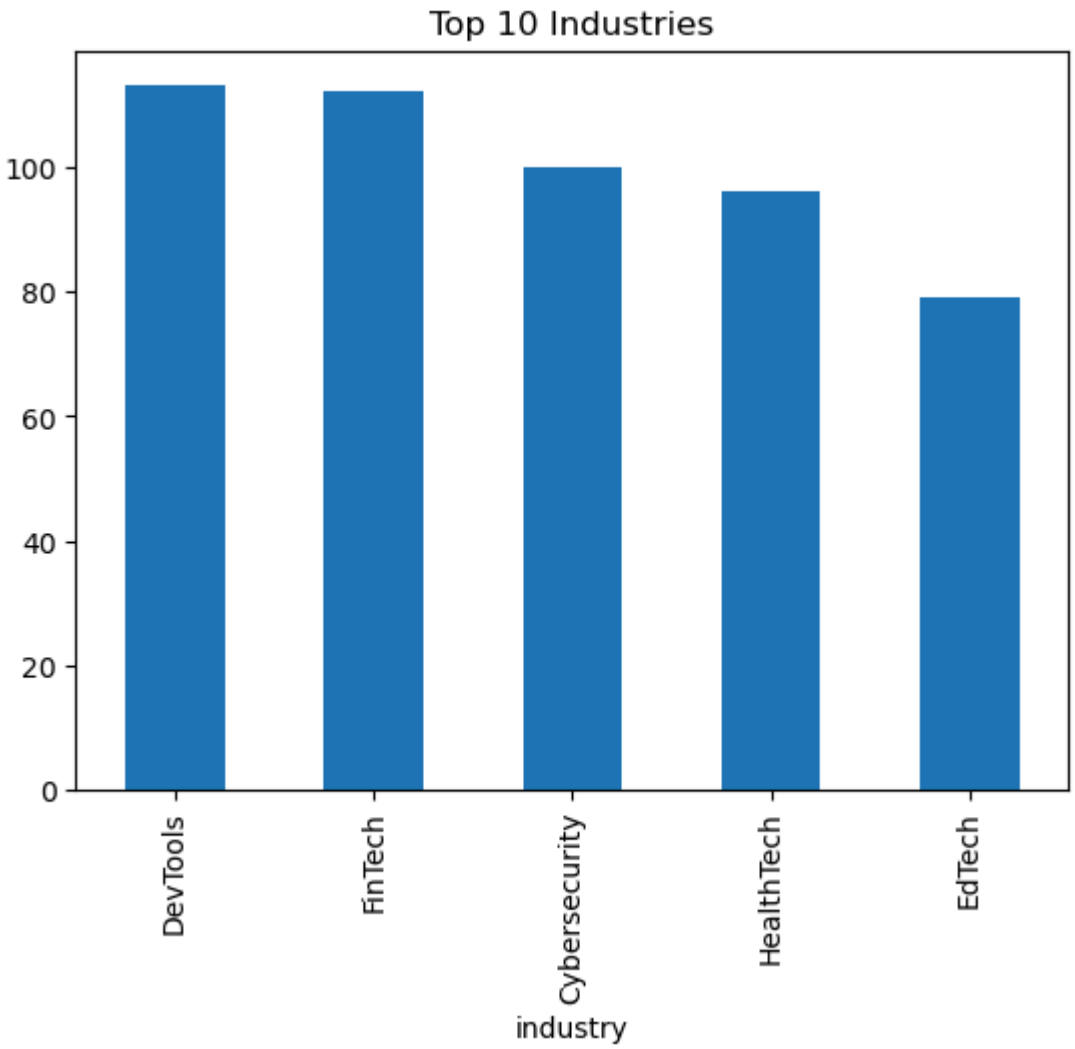


In [28]:
```
# Churn rate
churn_rate = accounts['churn_flag'].mean()
print(f"Overall churn rate: {churn_rate:.2%}")
```

```
Overall churn rate: 22.00%
```

In [29]:
```
# Industries
accounts['industry'].value_counts().head(10).plot(kind='bar', title='Top 10 Indu
```

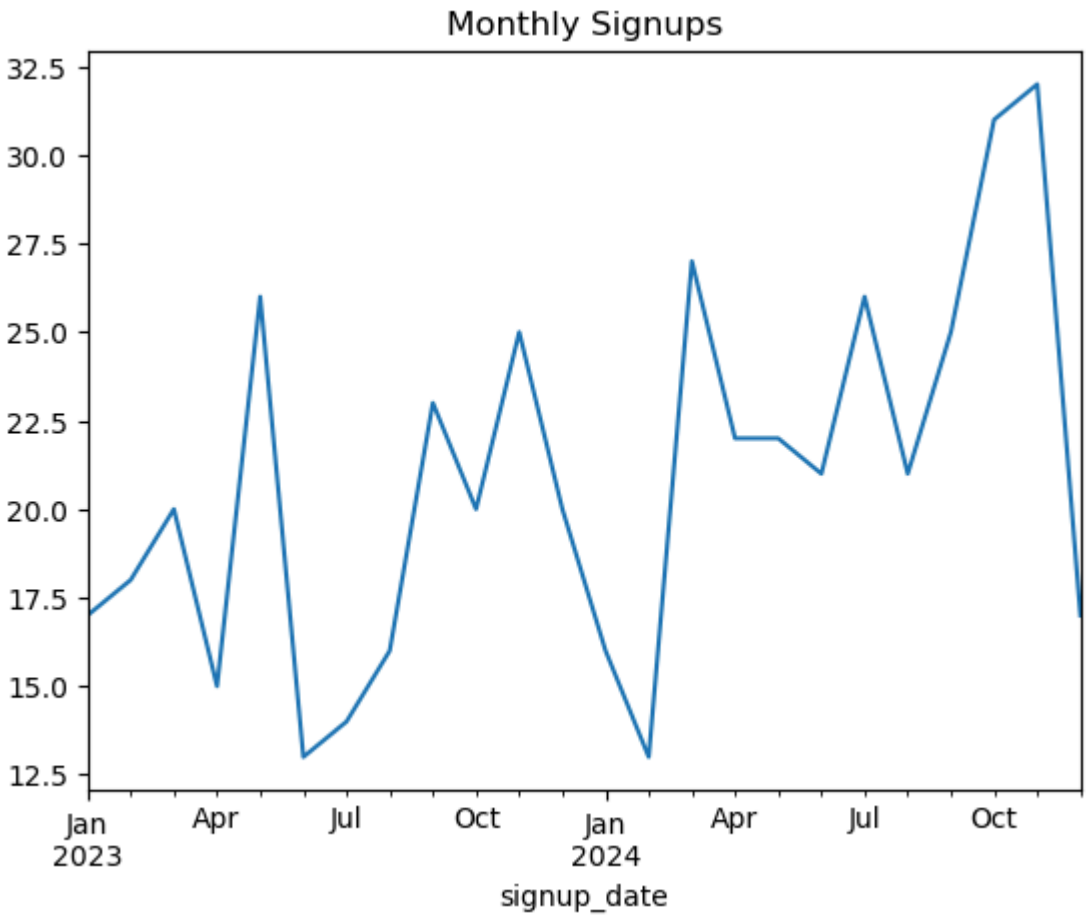
Out[29]:
```
<Axes: title={'center': 'Top 10 Industries'}, xlabel='industry'>
```



4. Time Trends

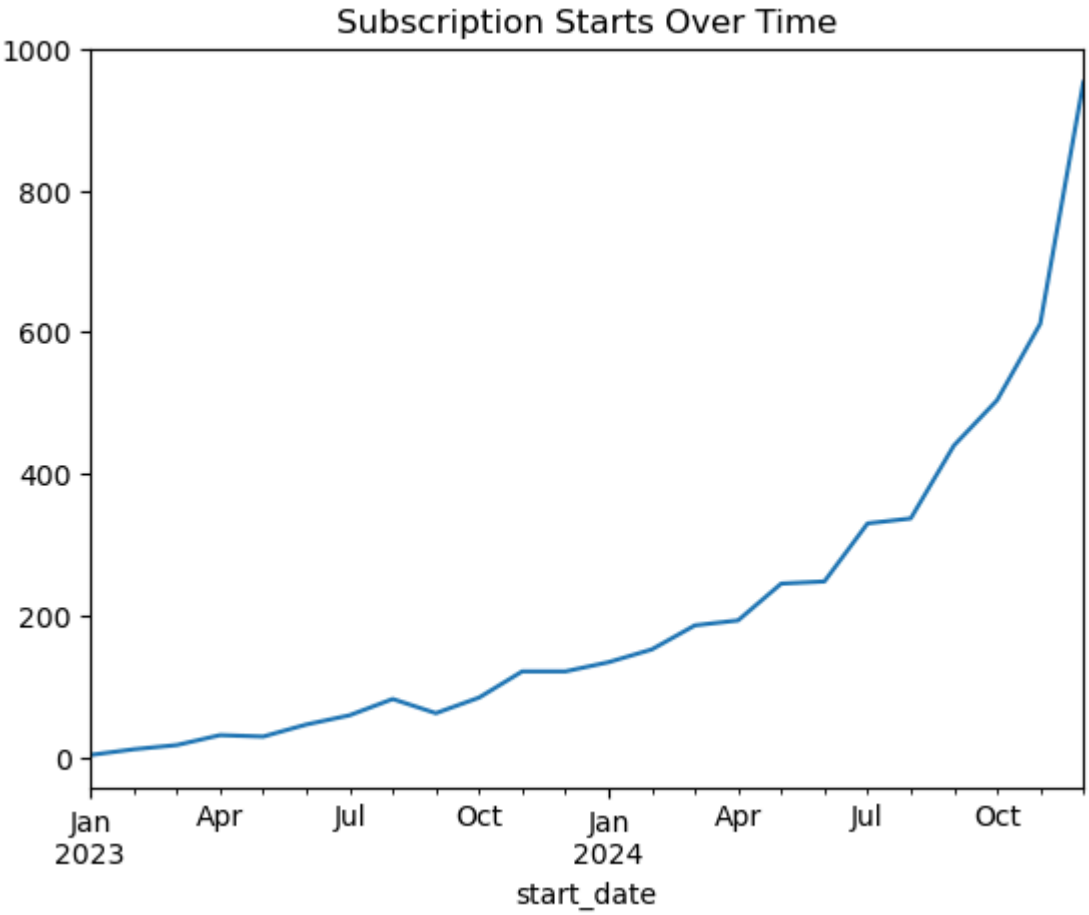
In [31]:
```
# Signups by Month
accounts['signup_date'] = pd.to_datetime(accounts['signup_date'])
accounts.set_index('signup_date').resample('M').size().plot(title="Monthly Signu
```

Out[31]:
```
<Axes: title={'center': 'Monthly Signups'}, xlabel='signup_date'>
```



In [32]:
```
# New subscriptions per month
subscriptions['start_date'] = pd.to_datetime(subscriptions['start_date'])
subscriptions.set_index('start_date').resample('M').size().plot(title="Subscript
```

Out[32]:
```
<Axes: title={'center': 'Subscription Starts Over Time'}, xlabel='start_date'>
```


Subscription Starts Over Time
1000
800
600
400
200
0
Jan
2023
Apr
Jul
Oct
Jan
2024
Apr
Jul
Oct
start_date

In [ ]: