Waze Project

Milestone 3 / 3a - Data exploration and cleaning. Visualization building

Exploratory data analysis

The purpose of this project is to conduct exploratory data analysis (EDA) on a provided dataset.

The goal is to continue the examination of the data, adding relevant visualizations that help communicate the story that the data tells.

This notebook has 4 parts:

Part 1: Imports, links, and loading

Part 2: Data Cleaning and Exploration

Part 3: Building visualizations

Part 4: Evaluating and Conclusion

Imports and data loading

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
# Load the dataset into a dataframe
df = pd.read_csv('waze_dataset.csv')
```

Data cleaning and exploration

In [3]:
```
df.head(10)
```

Out[3]:

	ID	label	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav1	total_navigati
0	0	retained	283	226	296.748273	2276	208	
1	1	retained	133	107	326.896596	1225	19	
2	2	retained	114	95	135.522926	2651	0	
3	3	retained	49	40	67.589221	15	322	
4	4	retained	84	68	168.247020	1562	166	
5	5	retained	113	103	279.544437	2637	0	
6	6	retained	3	2	236.725314	360	185	
7	7	retained	39	35	176.072845	2999	0	
8	8	retained	57	46	183.532018	424	0	

9	9	churned	84	68	244.802115	2997	72

In [5]:
```
df.size
```

Out[5]:
```
194987
```

In [6]:
```
df.describe()
```

Out[6]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.00000
mean	7499.000000	80.633776	67.281152	189.964447	1749.837789	121.60597
std	4329.982679	80.699065	65.913872	136.405128	1008.513876	148.12154
min	0.000000	0.000000	0.000000	0.220211	4.000000	0.00000
25%	3749.500000	23.000000	20.000000	90.661156	878.000000	9.00000
50%	7499.000000	56.000000	48.000000	159.568115	1741.000000	71.00000
75%	11248.500000	112.000000	93.000000	254.192341	2623.500000	178.00000
max	14998.000000	743.000000	596.000000	1216.154633	3500.000000	1236.00000

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   ID                       14999 non-null  int64  
 1   label                    14299 non-null  object 
 2   sessions                 14999 non-null  int64  
 3   drives                   14999 non-null  int64  
 4   total_sessions           14999 non-null  float64
 5   n_days_after_onboarding  14999 non-null  int64  
 6   total_navigations_fav1   14999 non-null  int64  
 7   total_navigations_fav2   14999 non-null  int64  
 8   driven_km_drives         14999 non-null  float64
 9   duration_minutes_drives  14999 non-null  float64
 10  activity_days            14999 non-null  int64  
 11  driving_days             14999 non-null  int64  
 12  device                   14999 non-null  object 
dtypes: float64(3), int64(8), object(2)
memory usage: 1.5+ MB
```

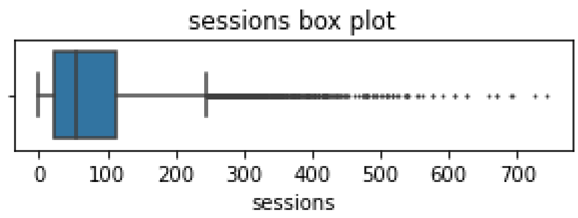
Visualizations

'Sessions' EDA

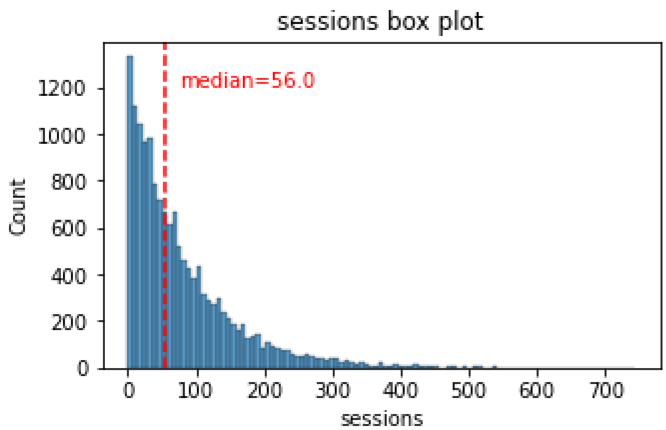
`sessions`

The number of occurrence of a user opening the app during the month

In [8]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['sessions'], fliersize=1)
plt.title('sessions box plot');
```



In [9]:
```
# Histogram
plt.figure(figsize=(5,3))
sns.histplot(x=df['sessions'])
median = df['sessions'].median()
plt.axvline(median, color='red', linestyle='--')
plt.text(75,1200, 'median=56.0', color='red')
plt.title('sessions box plot');
```



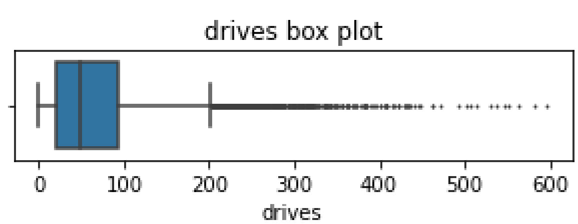
The `sessions` variable exhibits a skewed distribution to the right, where approximately 50% of the observations consist of 56 sessions or fewer. However, the boxplot reveals that a subset of users has more than 700 sessions.

'Drives' EDA

`drives`

An occurrence of driving at least 1 km during the month

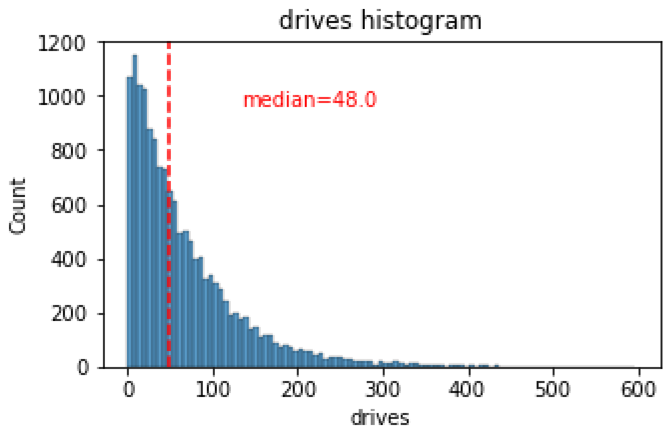
In [10]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['drives'], fliersize=1)
plt.title('drives box plot');
```



In [11]:
```
# Helper function to plot histograms based on the
# format of the `sessions` histogram
def histogrammer(column_str, median_text=True, **kwargs):    # **kwargs = any keyword ar
                                                             # from the sns.histplot() f
    median=round(df[column_str].median(), 1)
    plt.figure(figsize=(5,3))
    ax = sns.histplot(x=df[column_str], **kwargs)            # Plot the histogram
    plt.axvline(median, color='red', linestyle='--')         # Plot the median line
    if median_text==True:                                    # Add median text unless se
```

```
        ax.text(0.25, 0.85, f'median={median}', color='red',
            ha="left", va="top", transform=ax.transAxes)
    else:
        print('Median:', median)
    plt.title(f'{column_str} histogram');
```

In [12]:
```
# Histogram
histogrammer('drives')
```



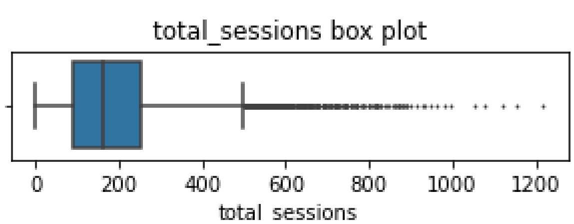
The **`drives`** data exhibits a distribution resembling that of the **`sessions`** variable. It is right-skewed, resembles a log-normal distribution, with a median of 48. However, a subset of drivers recorded over 400 drives in the last month.

'Total Sessions' EDA

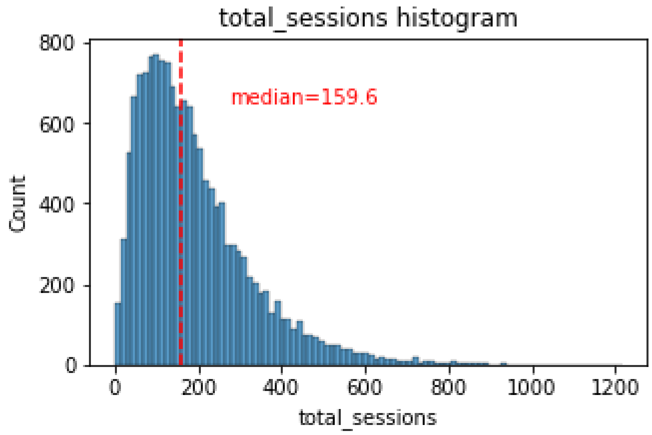
`total_sessions`

A model estimate of the total number of sessions since a user has onboarded

In [13]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['total_sessions'], fliersize=1)
plt.title('total_sessions box plot');
```



In [14]:
```
# Histogram
histogrammer('total_sessions')
```



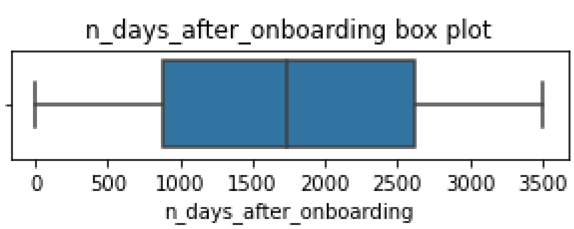
The distribution of `total_sessions` is right-skewed, appearing closer to a normal distribution compared to the previous variables. The median total number of sessions is approximately 159.6. This observation is noteworthy because if the median number of sessions in the last month was 48 and the median total sessions was around 160, it suggests that a significant proportion of a user's overall sessions possibly occurred within the last month.

'n Days After Onboarding' EDA

`n_days_after_onboarding`

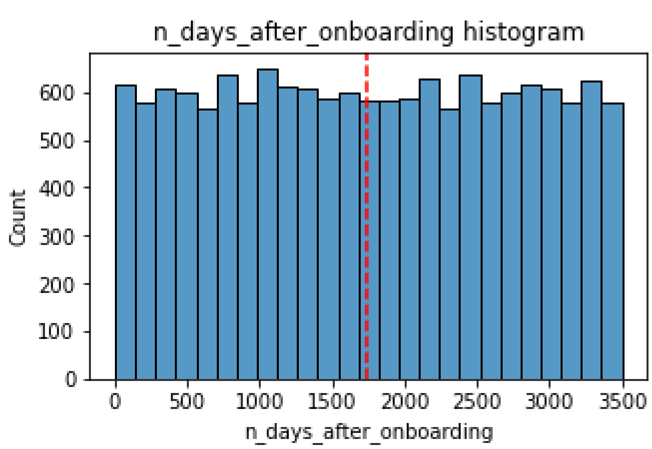
The number of days since a user signed up for the app

In [15]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['n_days_after_onboarding'], fliersize=1)
plt.title('n_days_after_onboarding box plot');
```



In [16]:
```
# Histogram
histogrammer('n_days_after_onboarding', median_text=False)
```

```
Median: 1741.0
```



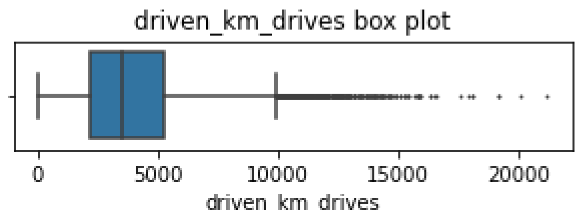
The total user tenure is a uniform distribution with values ranging from near-zero to ~3,500 days, or roughly 9.5 years.

'Driven KM Drives' EDA

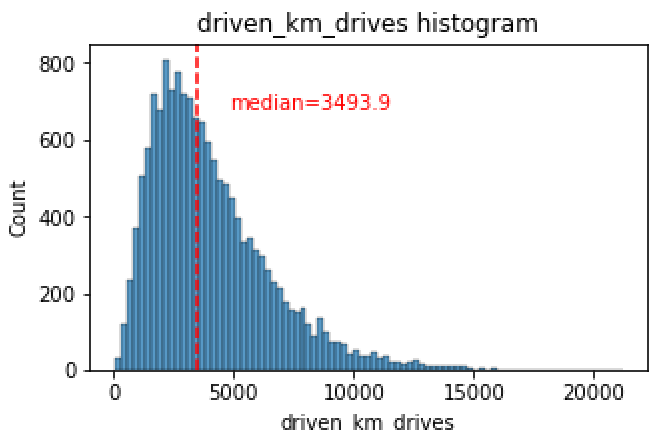
`driven_km_drives`

Total kilometers driven during the month

In [17]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['driven_km_drives'], fliersize=1)
plt.title('driven_km_drives box plot');
```



In [18]:
```
# Histogram
histogrammer('driven_km_drives')
```



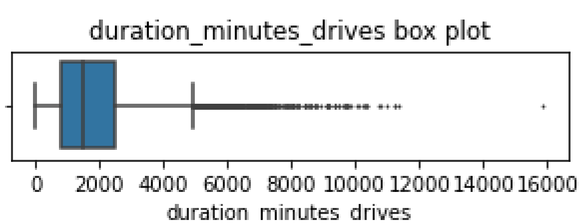
The distribution of drives completed by each user in the last month exhibits right-skewed normal distribution. Roughly 50% of users drove fewer than 3,495 kilometers during that period.

'Duration Minutes Drives' EDA

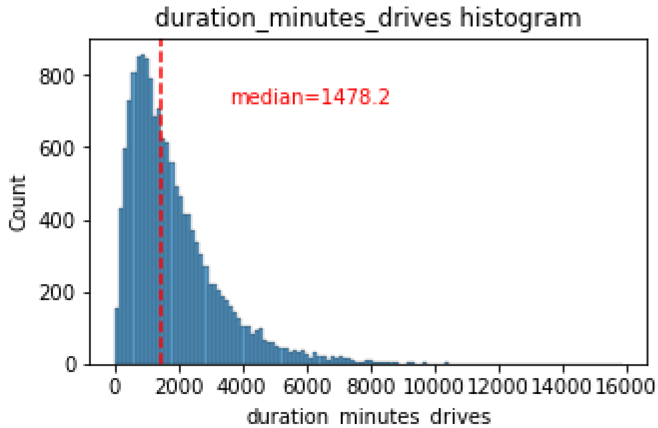
`duration_minutes_drives`

Total duration driven in minutes during the month

In [19]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['duration_minutes_drives'], fliersize=1)
plt.title('duration_minutes_drives box plot');
```



In [20]:
```
# Histogram
histogrammer('duration_minutes_drives')
```



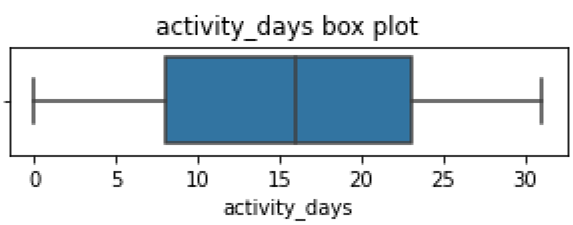
The `duration_minutes_drives` variable has a normalish distribution with a heavily skewed right tail. Around 50% of the users had a driving duration of less than 1,478 minutes (equivalent to about 25 hours), while certain users recorded over 250 hours of driving time throughout the month.

'Activity Days' EDA

`activity_days`

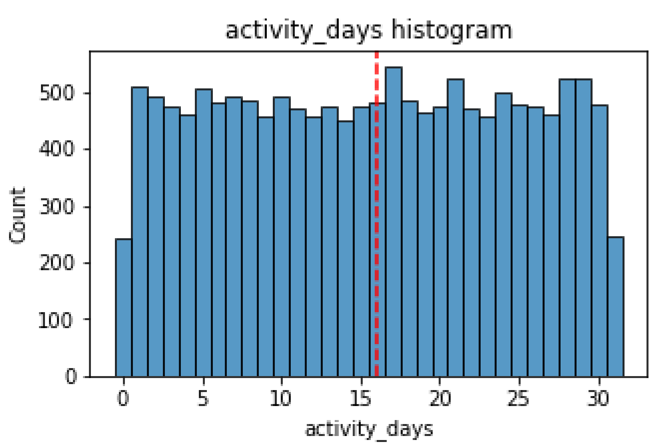
Number of days the user opens the app during the month

In [21]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['activity_days'], fliersize=1)
plt.title('activity_days box plot');
```



In [22]:
```
# Histogram
histogrammer('activity_days', median_text=False, discrete=True)
```

```
Median: 16.0
```



In the past month, users had a median of 16 app openings. The box plot displays a distribution that is centered. The histogram indicates a relatively uniform pattern with approximately 500 individuals opening the app on each day count. However, there are approximately 250 users who did not open the app at all, while another 250 users opened it every day throughout the month.

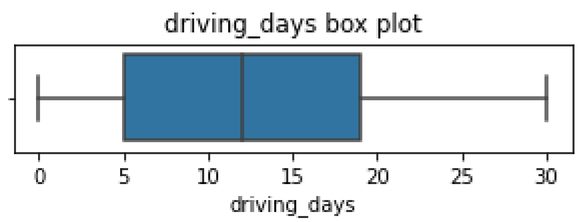
This distribution is of interest because it does not align with the distribution of `sessions`, which one might assume would be closely related to `activity_days`.

'Driving Days' EDA

`driving_days`

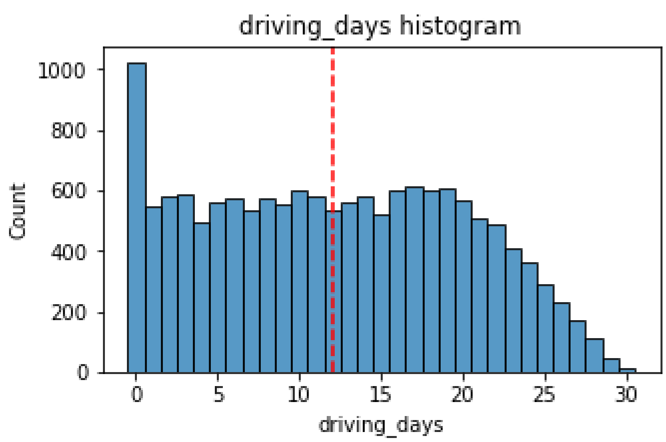
Number of days the user drives (at least 1 km) during the month

In [23]:
```
# Box plot
plt.figure(figsize=(5,1))
sns.boxplot(x=df['driving_days'], fliersize=1)
plt.title('driving_days box plot');
```



In [24]:
```
# Histogram
histogrammer('driving_days', median_text=False, discrete=True)
```

```
Median: 12.0
```



The frequency of users driving each month shows a relatively uniform pattern, closely aligned with the number of days they accessed the app within the same period. However, it's worth noting that the distribution of `driving_days` skews towards lower values.

Interestingly, there were nearly twice as many users (~1,000 versus ~550) who didn't engage in any driving activity throughout the month. This is interesting when considering the information provided about `activity_days`.

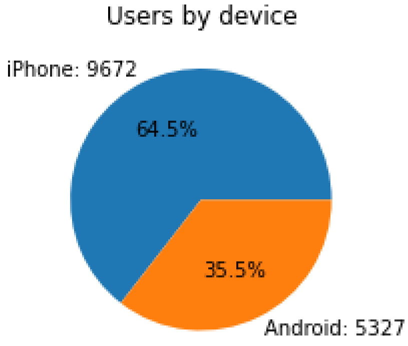
'Device' EDA

`device`

The type of device a user starts a session with

In [25]:
```
# Pie chart
fig = plt.figure(figsize=(3,3))
data=df['device'].value_counts()
plt.pie(data,
```

```
        labels=[f'{data.index[0]}: {data.values[0]}',
                f'{data.index[1]}: {data.values[1]}'],
        autopct='%1.1f%%'
        )
plt.title('Users by device');
```



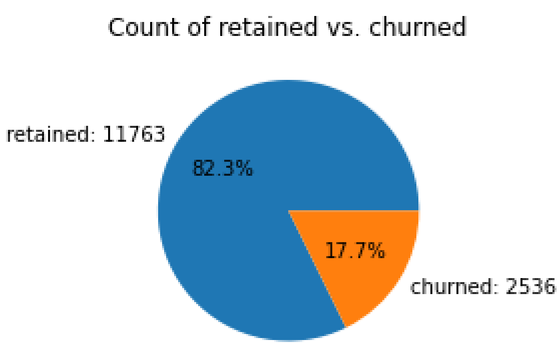
There are almost twice as many iPhone users as Android users.

'Label' EDA

`label`

Binary target variable ("retained" vs "churned") for if a user has churned anytime during the course of the month

In [26]:
```
# Pie chart
fig = plt.figure(figsize=(3,3))
data=df['label'].value_counts()
plt.pie(data,
        labels=[f'{data.index[0]}: {data.values[0]}',
                f'{data.index[1]}: {data.values[1]}'],
        autopct='%1.1f%%'
        )
plt.title('Count of retained vs. churned');
```



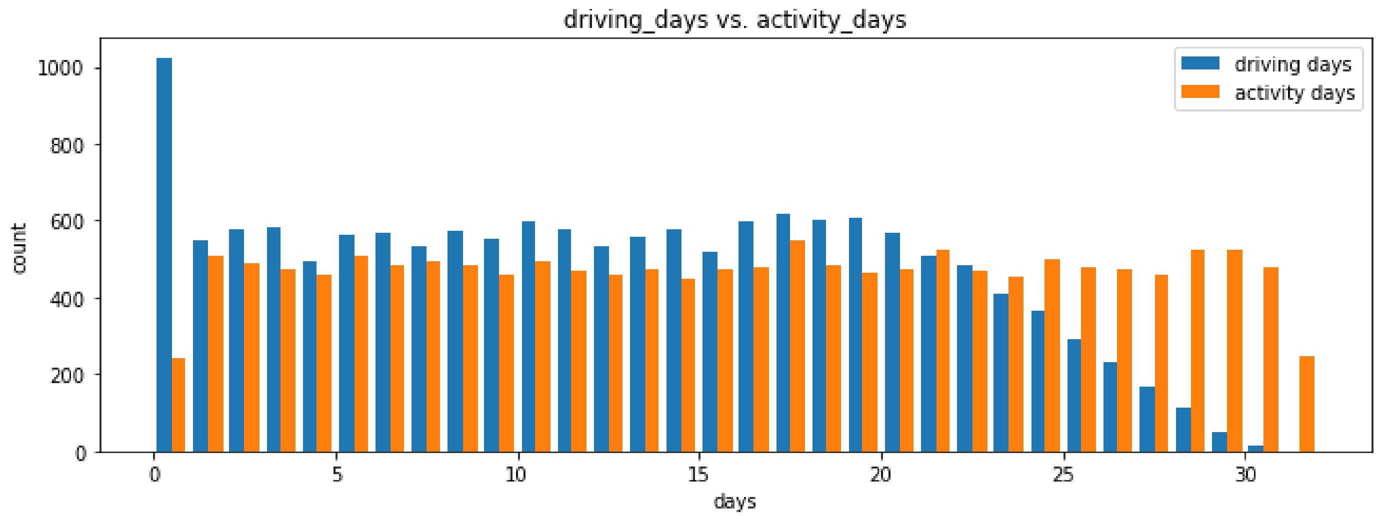
Most of the users were retained. Less than 18% of the users churned.

Driving Days vs Activity Days EDA

`driving days` vs. `activity days`

In [27]:
```
# Histogram
plt.figure(figsize=(12,4))
label=['driving days', 'activity days']
plt.hist([df['driving_days'], df['activity_days']],
         bins=range(0,33),
```

```
             label=label)
plt.xlabel('days')
plt.ylabel('count')
plt.legend()
plt.title('driving_days vs. activity_days');
```

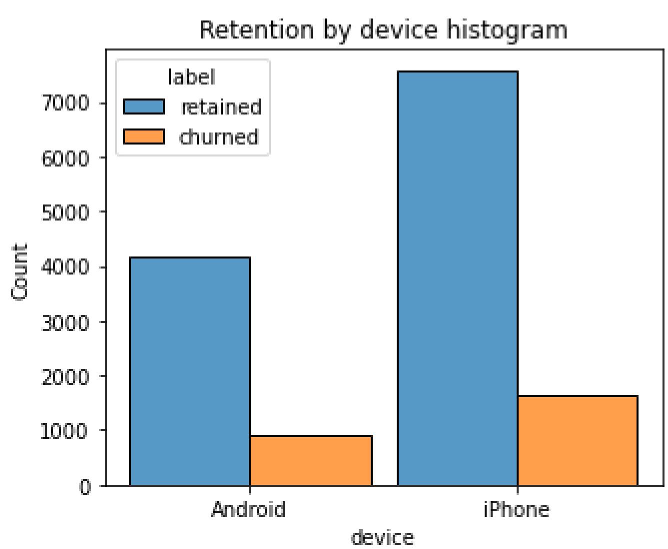


This is interesting. Initially, more users had an increase in `driving_days` compared to `activity_days`. They two stayed fairly consistent through until around day 21. Then, `driving_days` steadily declined, while `activity_days` remained near its previous levels. This would suggest that though users weren't driving as much, they were still opening and using the app.

Retention by device EDA

`Device` : iPhone vs Android

In [30]:
```
# Histogram
plt.figure(figsize=(5,4))
sns.histplot(data=df,
             x='device',
             hue='label',
             multiple='dodge',
             shrink=0.9
             )
plt.title('Retention by device histogram');
```



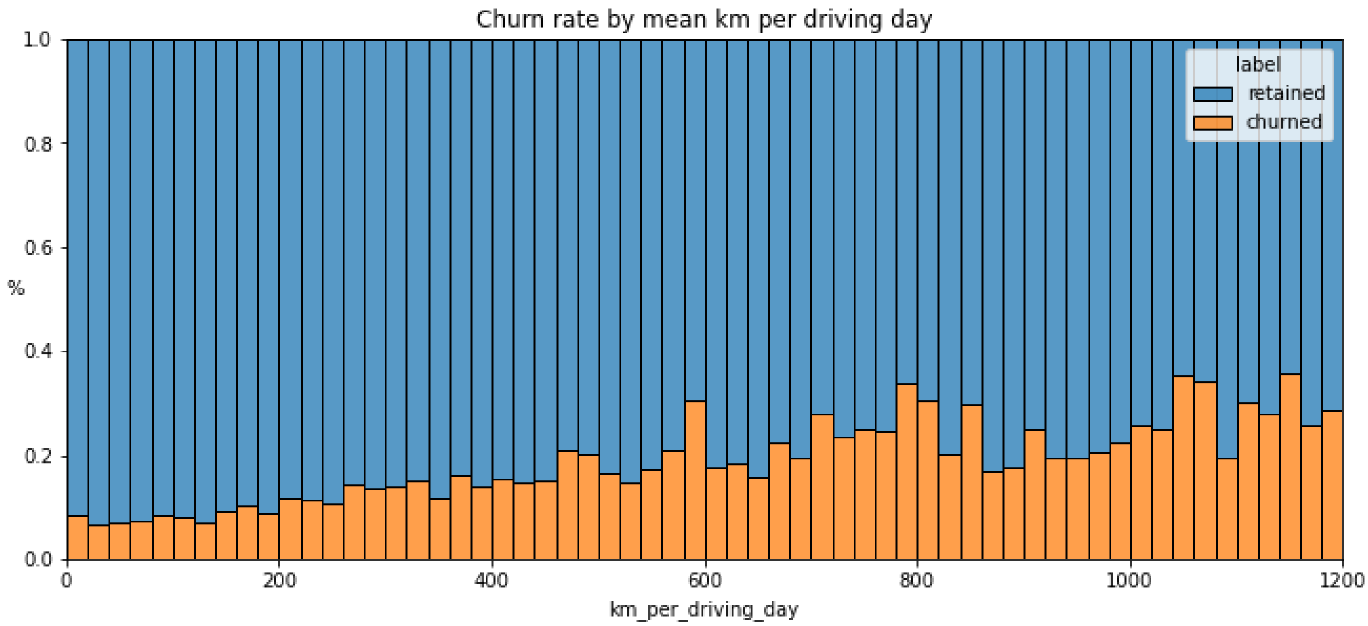
The ratio of users who churned to those who were retained remains consistent across both Android and iPhone devices. It is worth noting that iPhone users had higher numbers of churn and retention, thought that is likely due to the popularity of the iPhone.

Retention by kilometers driven per driving day EDA

`km_per_driving_day`

In [ ]:
```
# 1. Create `km_per_driving_day` column
df['km_per_driving_day'] = df['driven_km_drives'] / df['driving_days']
```

In [32]:
```
# Histogram
plt.figure(figsize=(12,5))
sns.histplot(data=df,
             x='km_per_driving_day',
             bins=range(0,1201,20),
             hue='label',
             multiple='fill')
plt.ylabel('%', rotation=0)
plt.title('Churn rate by mean km per driving day');
```

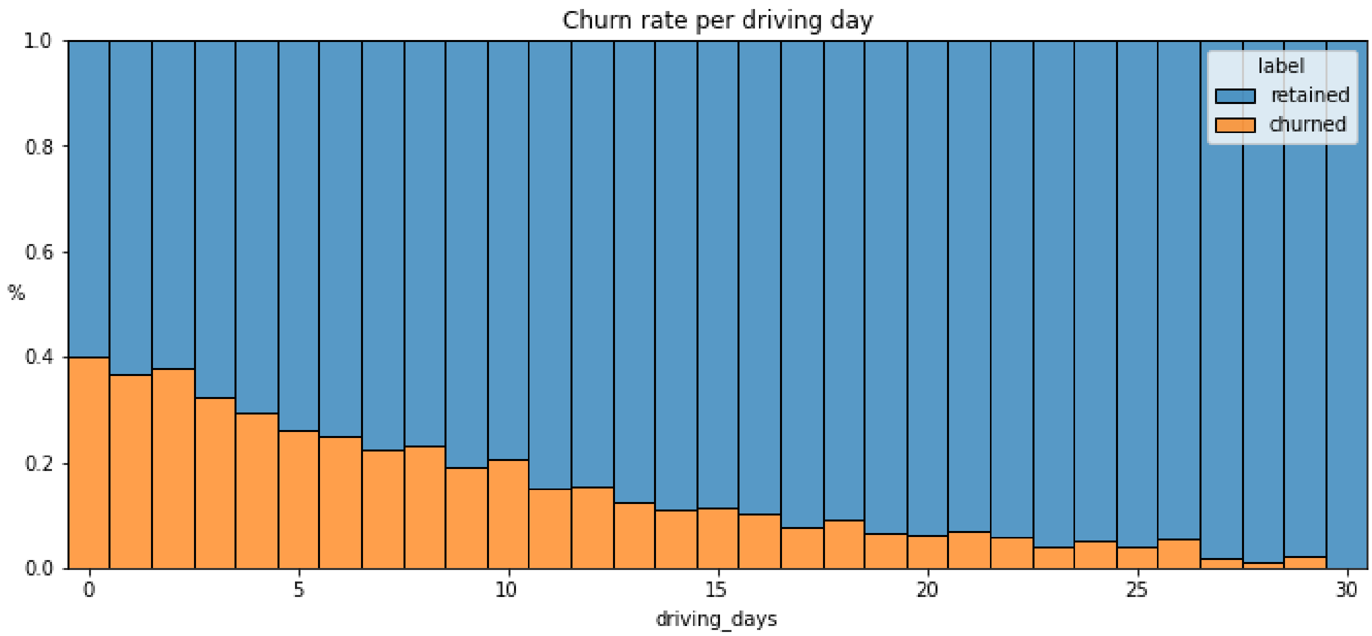


As the average daily distance driven increases, the churn rate also tends to rise. It would be valuable to delve deeper into the reasons why users who cover longer distances choose to discontinue using the app.

Churn rate per number of driving days EDA

`driving days`

In [33]:
```
# Histogram
plt.figure(figsize=(12,5))
sns.histplot(data=df,
             x='driving_days',
             bins=range(1,32),
             hue='label',
             multiple='fill',
             discrete=True)
plt.ylabel('%', rotation=0)
plt.title('Churn rate per driving day');
```



The likelihood of churn decreased as the frequency of app usage increased. Among users who did not use the app at all in the last month, 40% churned, whereas none of the users who used the app for 30 days experienced churn.

Proportion of sessions that occurred in the last month EDA

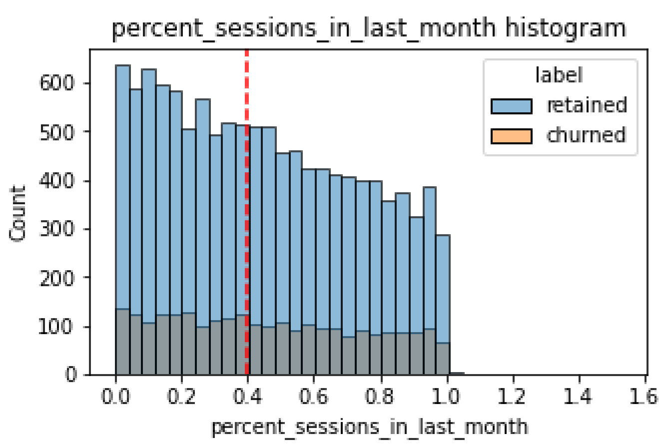
In [34]:
```
df['percent_sessions_in_last_month'] = df['sessions'] / df['total_sessions']
```

In [35]:
```
df['percent_sessions_in_last_month'].median()
```

Out[35]:
```
0.42309702992763176
```

In [36]:
```
# Histogram
histogrammer('percent_sessions_in_last_month',
              hue=df['label'],
              multiple='layer',
              median_text=False)
```

```
Median: 0.4
```

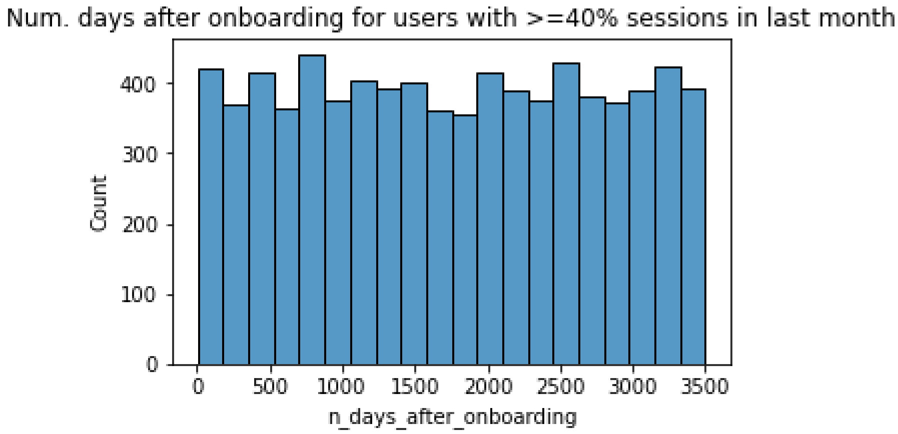


In [37]:
```
df['n_days_after_onboarding'].median()
```

Out[37]:
```
1741.0
```

Around half of the users included in the dataset had 40% or more of their sessions concentrated solely in the last month. Despite this, the median time elapsed since their initial onboarding is 4.77 years.

In [38]:
```
# Histogram
data = df.loc[df['percent_sessions_in_last_month']>=0.4]
plt.figure(figsize=(5,3))
sns.histplot(x=data['n_days_after_onboarding'])
plt.title('Num. days after onboarding for users with >=40% sessions in last month');
```



The number of days since users onboarded, who have experienced 40% or more of their total sessions within the last month, conforms to a uniform distribution. This is an interesting observation. Why the sudden surge in app usage by these longstanding users during the recent month?

Outliers due to skew

In [39]:
```
def outlier_imputer(column_name, percentile):
    # Calculate threshold
    threshold = df[column_name].quantile(percentile)
    # Impute threshold for values > than threshold
    df.loc[df[column_name] > threshold, column_name] = threshold

    print('{:>25} | percentile: {} | threshold: {}'.format(column_name, percentile, thre
```

In [40]:
```
for column in ['sessions', 'drives', 'total_sessions',
               'driven_km_drives', 'duration_minutes_drives']:
               outlier_imputer(column, 0.95)
```

```
                 sessions | percentile: 0.95 | threshold: 243.0
                   drives | percentile: 0.95 | threshold: 201.0
           total_sessions | percentile: 0.95 | threshold: 454.3632037399997
         driven_km_drives | percentile: 0.95 | threshold: 8889.7942356
  duration_minutes_drives | percentile: 0.95 | threshold: 4668.899348999999
```

In [41]:
```
df.describe()
```

Out[41]:

	ID	sessions	drives	total_sessions	n_days_after_onboarding	total_navigations_fav
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.00000
mean	7499.000000	76.568705	64.058204	184.031320	1749.837789	121.60597
std	4329.982679	67.297958	55.306924	118.600463	1008.513876	148.12154
min	0.000000	0.000000	0.000000	0.220211	4.000000	0.00000
25%	3749.500000	23.000000	20.000000	90.661156	878.000000	9.00000
50%	7499.000000	56.000000	48.000000	159.568115	1741.000000	71.00000
75%	11248.500000	112.000000	93.000000	254.192341	2623.500000	178.00000
max	14998.000000	243.000000	201.000000	454.363204	3500.000000	1236.00000

Conclusion

Types of distributions noticed in the variables:

- The majority of variables displayed either a strong right-skewness or a uniform distribution. In the case of right-skewed distributions, this indicates that a significant portion of users had values concentrated towards the lower end of the variable's range. Conversely, for variables exhibiting a uniform distribution, users had an approximately equal likelihood of possessing values across the entire range of that variable.

Indications the data may be erroneous or problematic:

- The majority of the data exhibited no issues, and there was no clear indication that any particular variable was entirely erroneous. However, a few variables contained highly unlikely or potentially impossible outlier values, such as driven_km_drives. Additionally, certain monthly variables, such as activity_days and driving_days, raise concerns as they possess conflicting maximum values of 31 and 30, respectively. This discrepancy suggests that data collection might not have been conducted within the same month for both of these variables, warranting further investigation.

Further questions that need to be explored or asked to the Waze team:

- I would like to inquire with the Waze data team to validate whether the monthly variables were collected within the same month, considering the discrepancy in maximum values—some variables indicating 30 days while others reflecting 31 days. Furthermore, I am interested in understanding the underlying reasons behind the sudden surge in app usage by a significant number of long-time users specifically within the last month. It would be valuable to investigate whether any changes occurred during that period that could have triggered such behavioral shifts.

Percentage of users churned and what percentage were retained:

- The churn rate among users was below 18%, while the majority, approximately 82%, were retained.

Factors that correlated with user churn?

- There was a positive correlation between the distance driven per driving day and user churn. In other words, the farther a user drove on each driving day, the higher the likelihood of churn. Conversely, the number of driving days exhibited a negative correlation with churn. Users who had a higher frequency of driving days within the last month were less likely to churn.

Representation of varying tenure lengths in the dataset:

- The data includes users spanning a range of tenures, from brand new to approximately 10 years, and they are fairly evenly represented. This observation is supported by the histogram depicting the distribution of n_days_after_onboarding, which demonstrates a uniform pattern for this variable.