Dataset

https://www.kaggle.com/datasets/faviovaz/marketing-ab-testing

Imports

In [4]:

```
# Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from scipy.stats import chi2_contingency, shapiro, levene, mannwhitneyu

import warnings
warnings.filterwarnings("ignore")

# Color schemas
colorcategories = ['#A6C8E0', '#3182BD','#1D3B5D']
colorback = 'rgba(0,0,0,0)'
colortext = '#36414e'
fsize = 12
```

Exploratory analysis

In [6]:

```
# Read data
df = pd.read_csv("marketing_AB.csv")
```

In [7]:

```
# Check first rows
df.head()
```

Out[7]:

	Unnamed: 0	user id	test group	converted	total ads	most ads day	most ads hour
0	0	1069124	ad	False	130	Monday	20
1	1	1119715	ad	False	93	Tuesday	22
2	2	1144181	ad	False	21	Tuesday	18
3	3	1435133	ad	False	355	Tuesday	10
4	4	1015700	ad	False	276	Friday	14

In [8]:

```
# Check null values and data types
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 588101 entries, 0 to 588100
Data columns (total 7 columns):
 #   Column         Non-Null Count   Dtype 
---  ------         --------------   ----- 
 0   Unnamed: 0     588101 non-null  int64 
 1   user id        588101 non-null  int64 
 2   test group     588101 non-null  object
 3   converted      588101 non-null  bool  
 4   total ads      588101 non-null  int64 
 5   most ads day   588101 non-null  object
 6   most ads hour  588101 non-null  int64 
dtypes: bool(1), int64(4), object(2)
memory usage: 27.5+ MB
```

In [9]:

```
# Check if user id is unique
df['user id'].is_unique
```

Out[9]:

```
True
```

In [10]:

```
# Drop unwanted columns
df = df.drop(columns={"Unnamed: 0", "user id"})
```

In [11]:

```
df.columns
```

Out[11]:

```
Index(['test group', 'converted', 'total ads', 'most ads day',
       'most ads hour'],
      dtype='object')
```

In [12]:

```
# Create a dataframe with only categorical variables
df_cat = df[['test group', 'converted', 'most ads day', 'most ads hour']]
df_cat.nunique()
```

Out[12]:

```
test group        2
converted         2
most ads day      7
most ads hour    24
dtype: int64
```

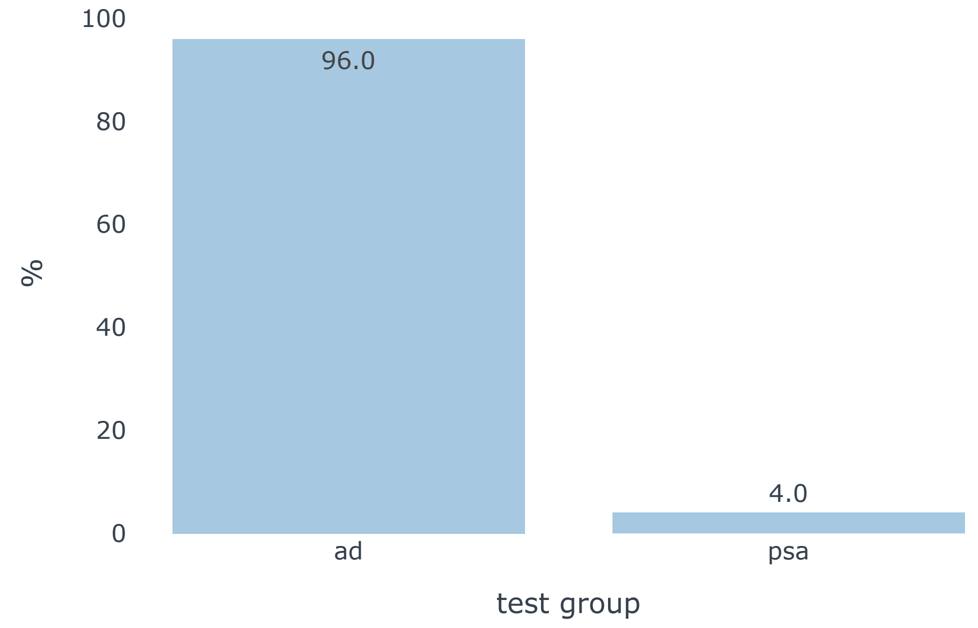
In [13]:

```
for i in df_cat.columns:
    print(i, ':',df_cat[i].unique())
```

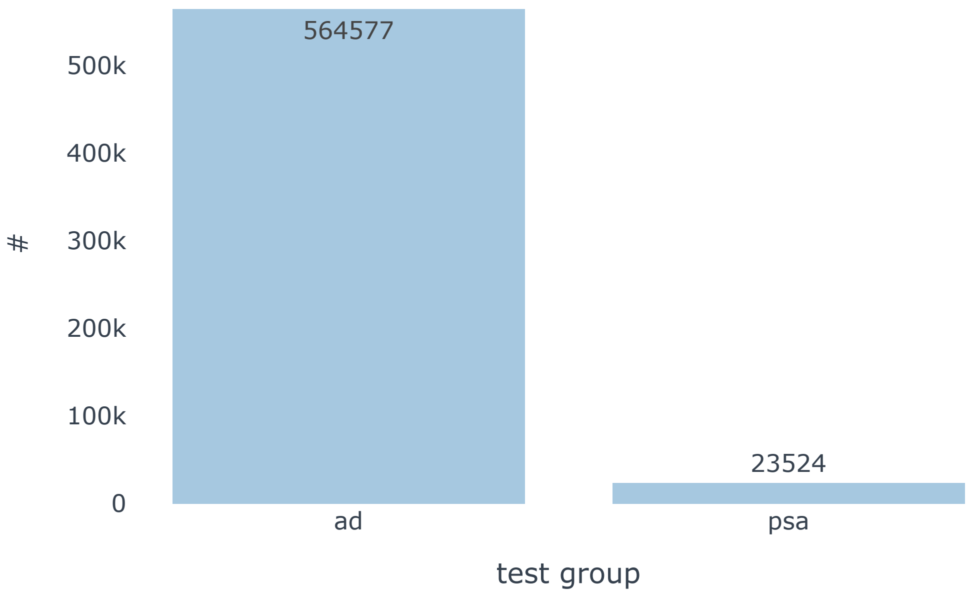
```
test group : ['ad' 'psa']
converted : [False  True]
most ads day : ['Monday' 'Tuesday' 'Friday' 'Saturday' 'Wednesday' 'Sunday' 'Thursday']
most ads hour : [20 22 18 10 14 13 19 11 12 16 21  3 23  4  8  0  2 15  1  6 17  7  9  5]
```

In [14]:

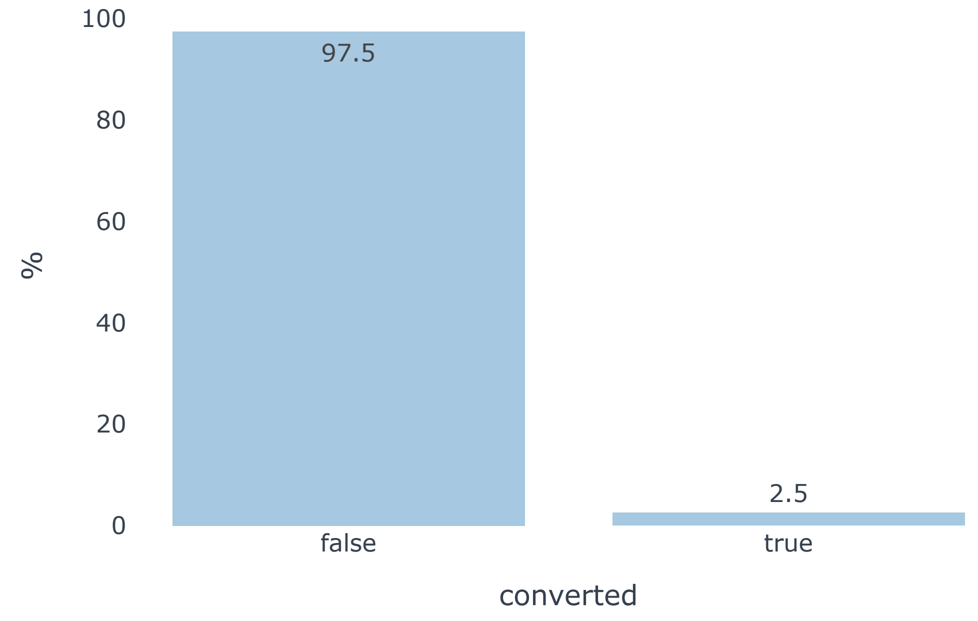
```
# Test group percentage
fig = px.histogram(df_cat, x ='test group', color_discrete_sequence=colorcategories, text_auto='.1f', histnorm='percent')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='test group',
                  yaxis_title='%',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```



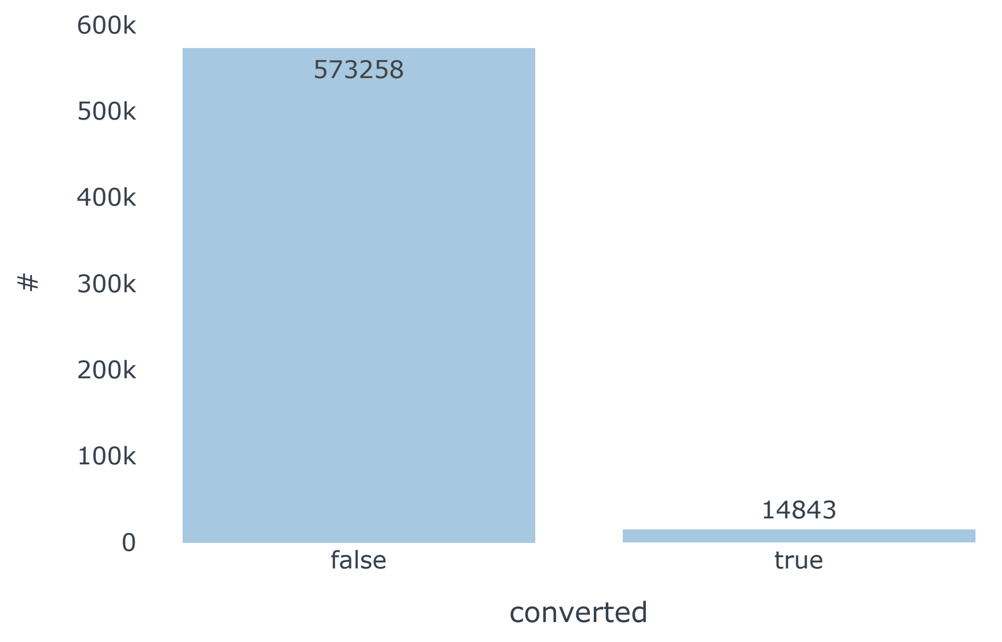
In [15]:
```
# Test group number
fig = px.histogram(df_cat, x ='test group', color_discrete_sequence=colorcategories, text_auto='.0f')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='test group',
                  yaxis_title='#',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```



In [16]:
```
# Converted percentage
fig = px.histogram(df_cat, x='converted', color_discrete_sequence=colorcategories, text_auto='.1f', histnorm='percent')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='converted',
                  yaxis_title='%',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```

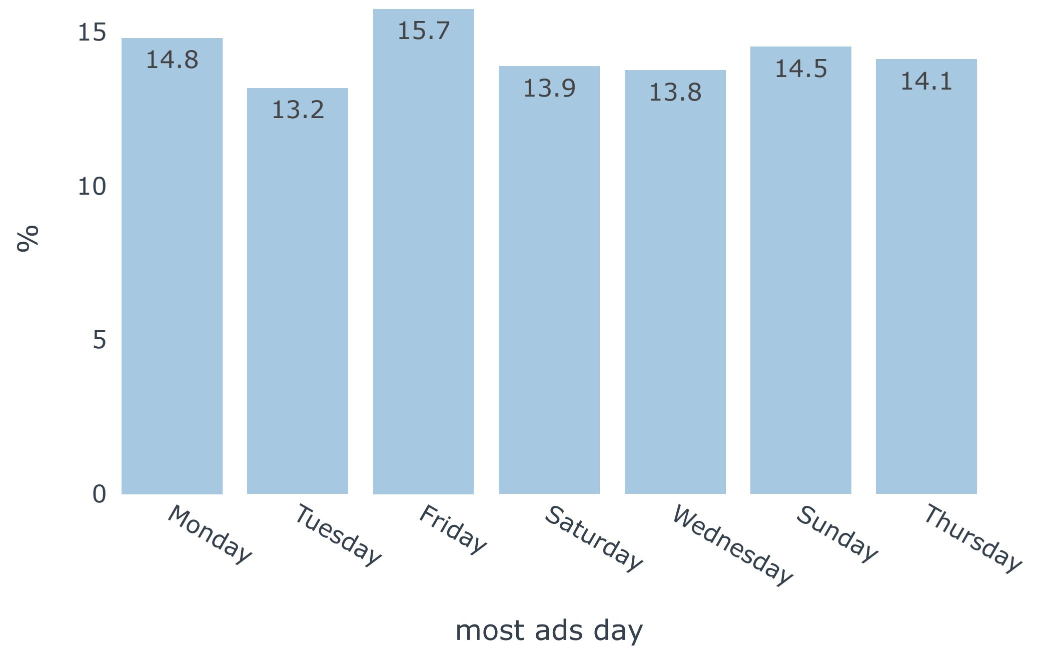


In [17]:
```
# Test group number
fig = px.histogram(df_cat, x='converted', color_discrete_sequence=colorcategories, text_auto='.0f')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='converted',
                  yaxis_title='#',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```



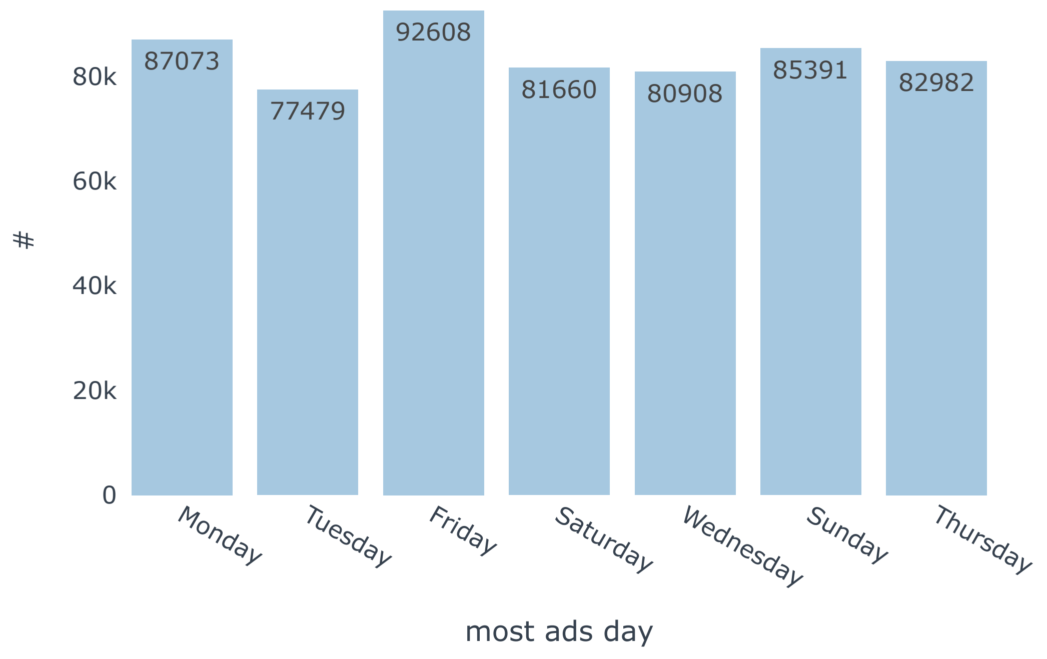
In [18]:

```python
# Most ads day percentage
fig = px.histogram(df_cat, x ='most ads day', color_discrete_sequence=colorcategories, text_auto='.1f', histnorm='percent')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='most ads day',
                  yaxis_title='%',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```



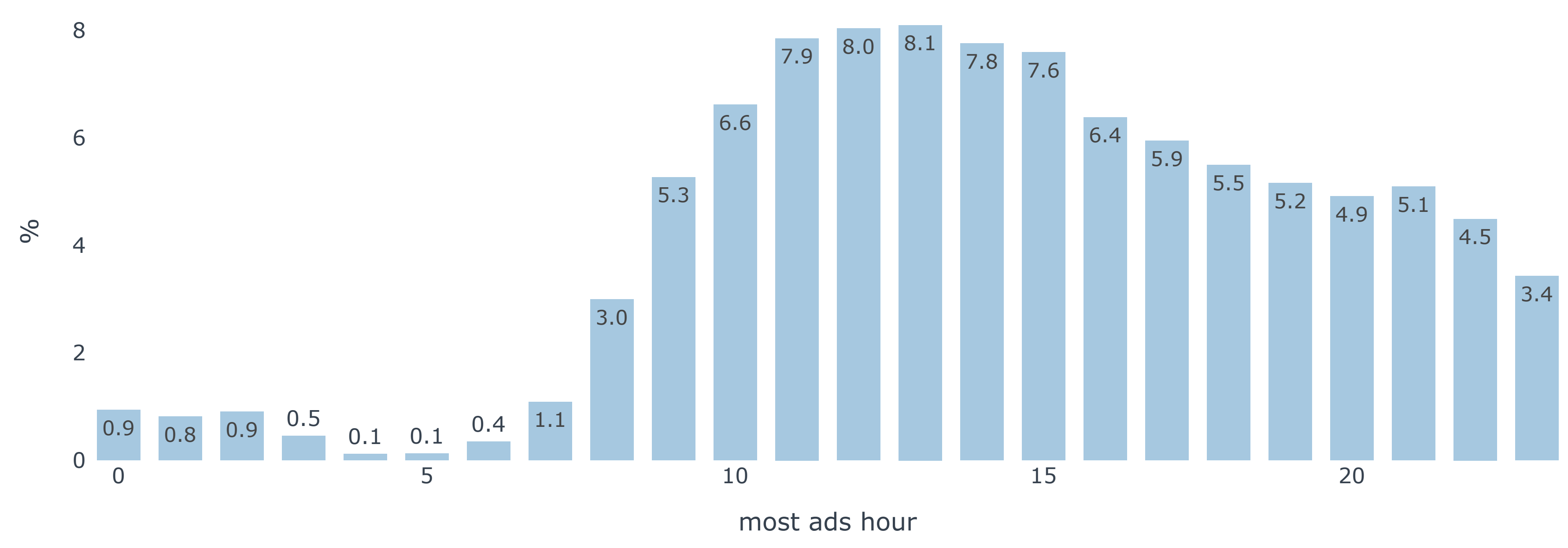
In [19]:

```python
# Most ads day number
fig = px.histogram(df_cat, x='most ads day', color_discrete_sequence=colorcategories, text_auto='.0f')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='most ads day',
                  yaxis_title='#',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=600,
                  height=400,
                  showlegend=False)

fig.show()
```



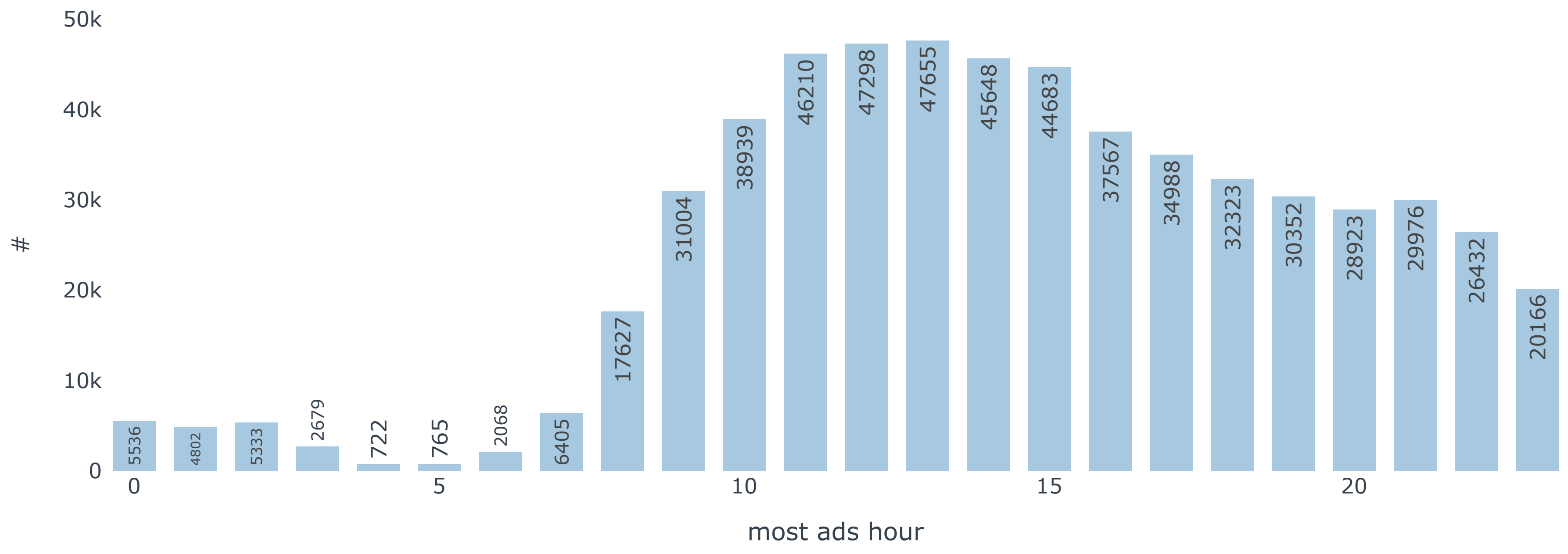
In [20]:

```python
# Most ads hour percentage
fig = px.histogram(df_cat, x='most ads hour', color_discrete_sequence= colorcategories,
                   text_auto='.1f', histnorm='percent', barmode='group')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='most ads hour',
                  yaxis_title='%',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=1000,
                  height=400,
                  showlegend=False,
                  bargap=0.3)

fig.update_traces(textangle=0)

fig.show()
```



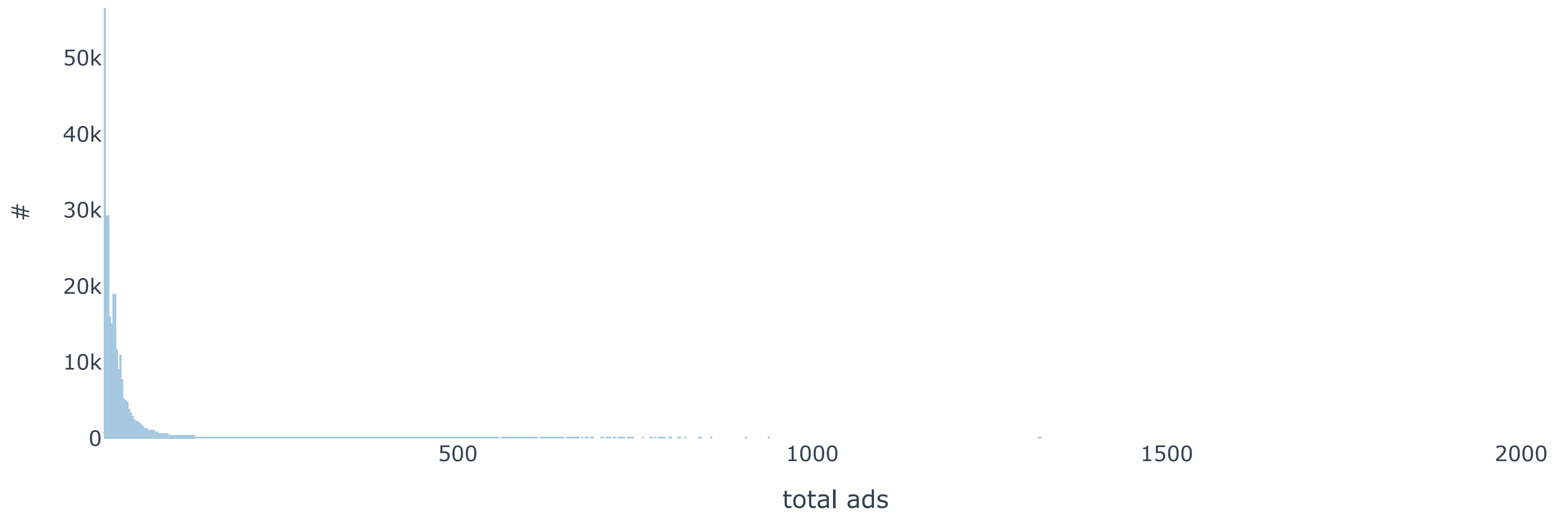
In [21]:

```python
# Most ads day number
fig = px.histogram(df_cat, x='most ads hour', color_discrete_sequence=colorcategories, text_auto='.0f', barmode='group')

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='most ads hour',
                  yaxis_title='#',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=1000,
                  height=400,
                  showlegend=False,
                  bargap=0.3)

fig.update_traces(textangle=-90)


fig.show()
```



In [22]:

```python
# Total ads number
fig = px.histogram(df, x='total ads', color_discrete_sequence=colorcategories)

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='total ads',
                  yaxis_title='#',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=1000,
                  height=400,
                  showlegend=False)



fig.show()
```



In [23]:

```python
# Total ads boxplot
fig = px.box(df, y='total ads', color_discrete_sequence=colorcategories)

fig.update_layout(font_size=fsize,
                  font_color=colortext,
                  title='',
                  xaxis_title='',
                  yaxis_title='total ads',
                  paper_bgcolor=colorback,
                  plot_bgcolor=colorback,
                  width=400,
                  height=1000,
                  showlegend=False)



fig.show()
```

```
In [24]: df['total ads'].describe()
```

```
Out[24]: count    588101.000000
         mean         24.820876
         std          43.715181
         min           1.000000
         25%           4.000000
         50%          13.000000
         75%          27.000000
         max        2065.000000
         Name: total ads, dtype: float64
```

```
In [25]: # Total ads number (reduced x axis range)
         fig = px.histogram(df, x='total ads', color_discrete_sequence=colorcategories)

         fig.update_layout(font_size=fsize,
                           font_color=colortext,
                           title='',
                           xaxis_title='total ads',
                           yaxis_title='#',
                           paper_bgcolor=colorback,
                           plot_bgcolor=colorback,
                           width=1000,
                           height=400,
                           xaxis_range=[0,50],
                           showlegend=False)



         fig.show()
```