Exploring categorical variables

Waffle chart

Import data

In [2]:

```
import pandas as pd

ROOT = "https://raw.githubusercontent.com/kirenz/modern-statistics/main/data/"
DATA = "loans.csv"

df = pd.read_csv(ROOT + DATA)

df.head()
```

Out[2]:

	emp_title	emp_length	state	homeownership	annual_income	verified_income	debt_to_income	annual_income_joint	verification_income_joint	debt_to_income_joint	...	sub_grade	issue_month	loan_stat
0	global config engineer	3.0	NJ	mortgage	90000.0	Verified	18.01	NaN	NaN	NaN	...	C3	Mar-2018	Curren
1	warehouse office clerk	10.0	HI	rent	40000.0	Not Verified	5.04	NaN	NaN	NaN	...	C1	Feb-2018	Curren
2	assembly	3.0	WI	rent	40000.0	Source Verified	21.15	NaN	NaN	NaN	...	D1	Feb-2018	Curren
3	customer service	1.0	PA	rent	30000.0	Not Verified	10.16	NaN	NaN	NaN	...	A3	Jan-2018	Curren
4	security supervisor	10.0	CA	rent	35000.0	Verified	57.96	57000.0	Verified	37.66	...	C3	Mar-2018	Curren

5 rows × 55 columns

Waffle chart

In [4]:

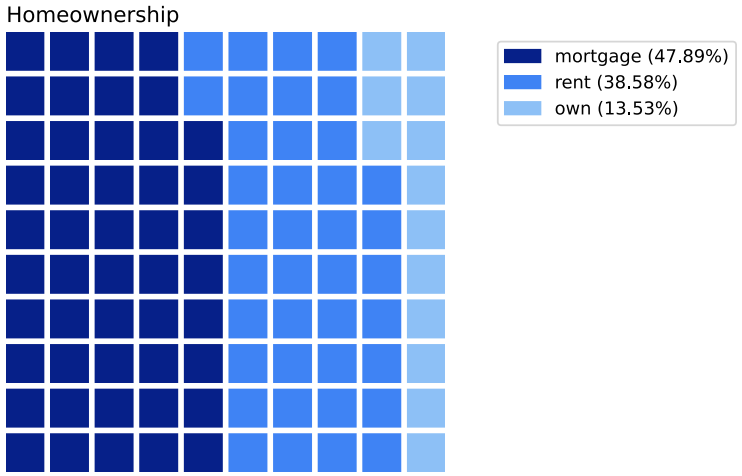
```
%matplotlib inline
import matplotlib.pyplot as plt
from pywaffle import Waffle

# Custom colors
blue = "#3F83F4"
blue_dark = "#062089"
blue_light = "#8DC0F6"
blue_lighter = "#BBE4FA"
grey = "#9C9C9C"
grey_dark = "#777777"
grey_light = "#B2B2B2"
orange = "#EF8733"
colors_blue = [blue_dark, blue, blue_light]
colors = [blue_dark, blue, orange, grey_dark, grey, grey_light]
```

In [5]:

```
data = (df.homeownership.value_counts() / df.homeownership.value_counts().sum())*100
values = data.round(2).to_dict()

fig = plt.figure(
    figsize=(6,6), FigureClass=Waffle,
    rows=10, values=values, colors=colors_blue,
    title={'label': 'Homeownership', 'loc': 'left'},
    labels=[f"{k} ({v}%)" for k, v in values.items()],
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
)
```



In [6]:

```
data = (df.loan_status.value_counts() / df.loan_status.value_counts().sum())*100
values = data.round(2).to_dict()

fig = plt.figure(
    figsize=(6,6), FigureClass=Waffle,
    rows=10, values=values, colors=colors,
    title={'label': 'Loan status', 'loc': 'left'},
    labels=[f"{k} ({v}%)" for k, v in values.items()],
    legend={'loc': 'upper left', 'bbox_to_anchor': (1.1, 1)}
)
```