Case study: Passwords

Import data

In [1]:

```
import pandas as pd

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

df = pd.read_csv(ROOT + DATA)
```

Data inspection

In [2]:

```
# First ten rows of the passwords dataset.
df.head(10)
```

Out[2]:

	rank	password	category	value	time_unit	offline_crack_sec	rank_alt	strength	font_size
0	1	password	password-related	6.91	years	2.170000e+00	1	8	11
1	2	123456	simple-alphanumeric	18.52	minutes	1.110000e-05	2	4	8
2	3	12345678	simple-alphanumeric	1.29	days	1.110000e-03	3	4	8
3	4	1234	simple-alphanumeric	11.11	seconds	1.110000e-07	4	4	8
4	5	qwerty	simple-alphanumeric	3.72	days	3.210000e-03	5	8	11
5	6	12345	simple-alphanumeric	1.85	minutes	1.110000e-06	6	4	8

	rank	password	category	value	time_unit	offline_crack_sec	rank_alt	strength	font_size
6	7	dragon	animal	3.72	days	3.210000e-03	7	8	11
7	8	baseball	sport	6.91	years	2.170000e+00	8	4	8
8	9	football	sport	6.91	years	2.170000e+00	9	7	11
9	10	letmein	password-related	3.19	months	8.350000e-02	10	8	11

In [3]:

```
# Bottom ten rows of the passwords dataset.
df.tail(10)
```

Out[3]:

	rank	password	category	value	time_unit	offline_crack_sec	rank_alt	strength	font_size
490	491	natasha	name	3.19	months	0.08350	493	7	11
491	492	sniper	cool-macho	3.72	days	0.00321	494	8	11
492	493	chance	name	3.72	days	0.00321	495	7	11
493	494	genesis	nerdy-pop	3.19	months	0.08350	496	7	11
494	495	hotrod	cool-macho	3.72	days	0.00321	497	7	11
495	496	reddog	cool-macho	3.72	days	0.00321	498	6	10
496	497	alexande	name	6.91	years	2.17000	499	9	12
497	498	college	nerdy-pop	3.19	months	0.08350	500	7	11
498	499	jester	name	3.72	days	0.00321	501	7	11
499	500	passw0rd	password-related	92.27	years	29.02000	502	28	21

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 9 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   rank                  500 non-null    int64
 1   password              500 non-null    objec
t
 2   category              500 non-null    objec
t
 3   value                 500 non-null    float
64
 4   time_unit             500 non-null    objec
t
 5   offline_crack_sec     500 non-null    float
```

```
64
 6   rank_alt              500 non-null    int64
 7   strength              500 non-null    int64
 8   font_size             500 non-null    int64
dtypes: float64(2), int64(4), object(3)
memory usage: 35.3+ KB
```

Data transformation

In [5]:

```
df["category"] = df["category"].astype("category")
df["time_unit"] = df["time_unit"].astype("category")
df["strenght"] = df["strength"].astype("category")
```

In [6]:

```
df.dtypes
```

Out[6]:

```
rank                 int64
password            object
category          category
value              float64
time_unit         category
offline_crack_sec  float64
rank_alt             int64
strength             int64
font_size            int64
strenght          category
dtype: object
```

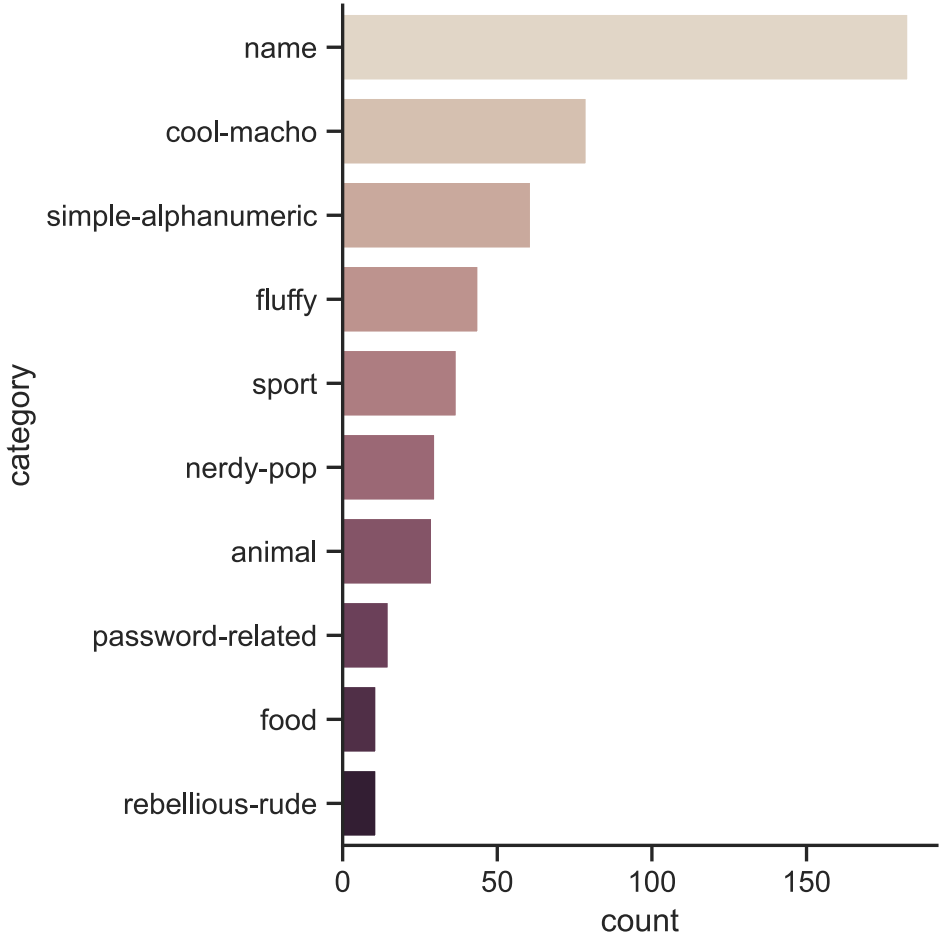
Data exploration

In [7]:

```
%matplotlib inline
%config InlineBackend.figure_formats = ['svg']

import seaborn as sns
import matplotlib.pyplot as plt

sns.set_theme(style="ticks", color_codes=True)
```

In [8]:

```
TOP_10 = df['category'].value_counts().iloc[:10].index

sns.catplot(y="category",
            kind="count",
            palette="ch:.25",
            data=df,
            order = TOP_10)

plt.show();
```


name
cool-macho
simple-alphanumeric
fluffy
sport
nerdy-pop
animal
password-related
food
rebellious-rude
category
0
50
100
150
count

- Next, we want to use `catplot` for all categorical variables.
- We use a for loop to obtain the desired result.
- First, let's extract all categorical variabel names as list:

In [9]:

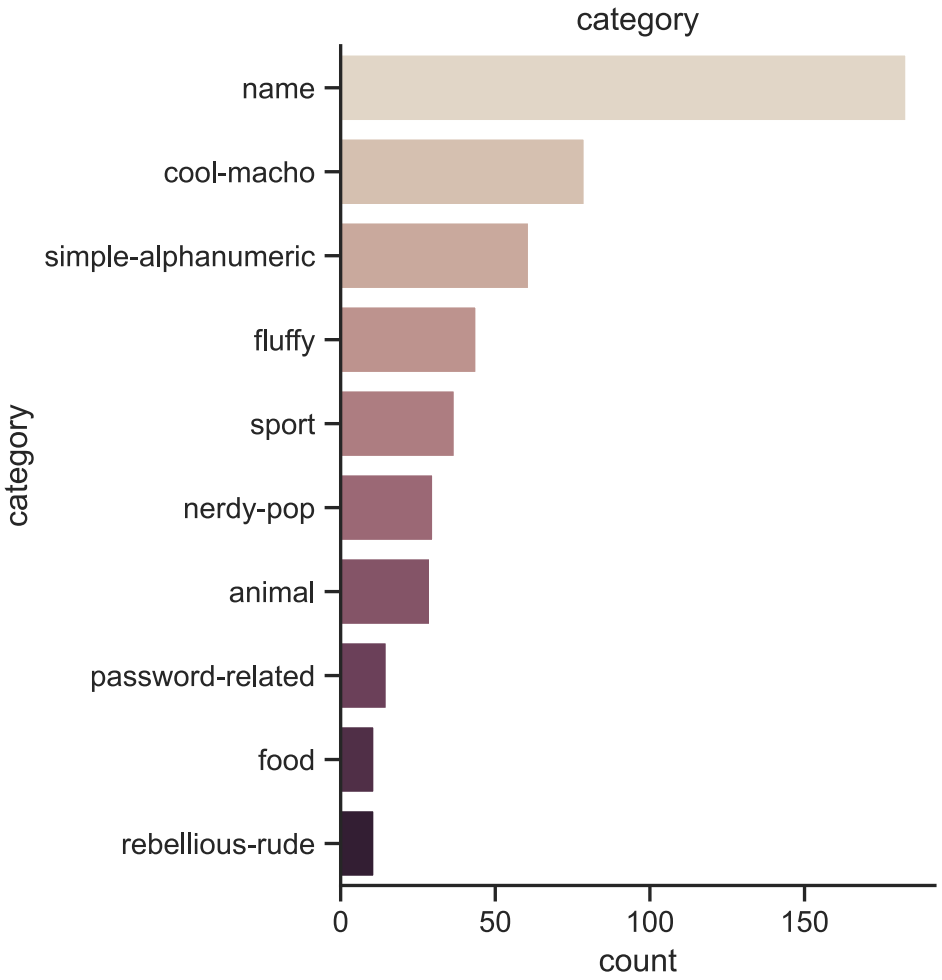
```
CATEGORICAL = df.select_dtypes(include=['category']).columns.tolist()
CATEGORICAL
```

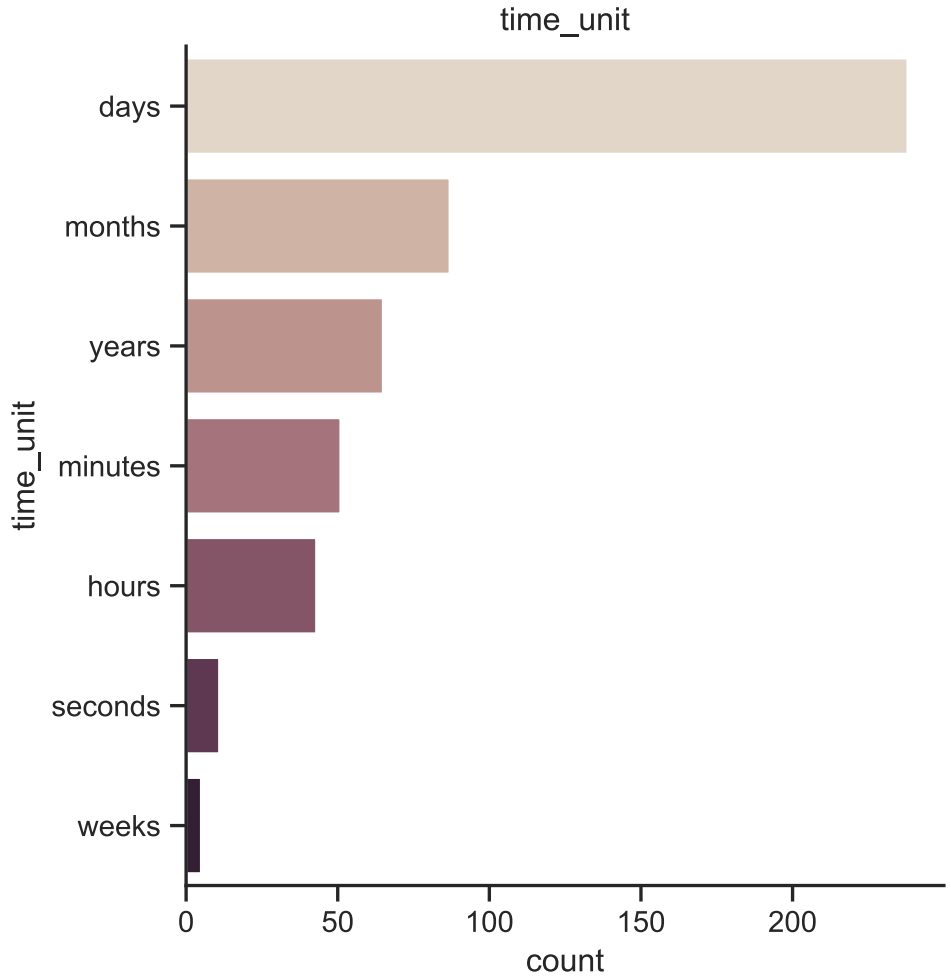
Out[9]:

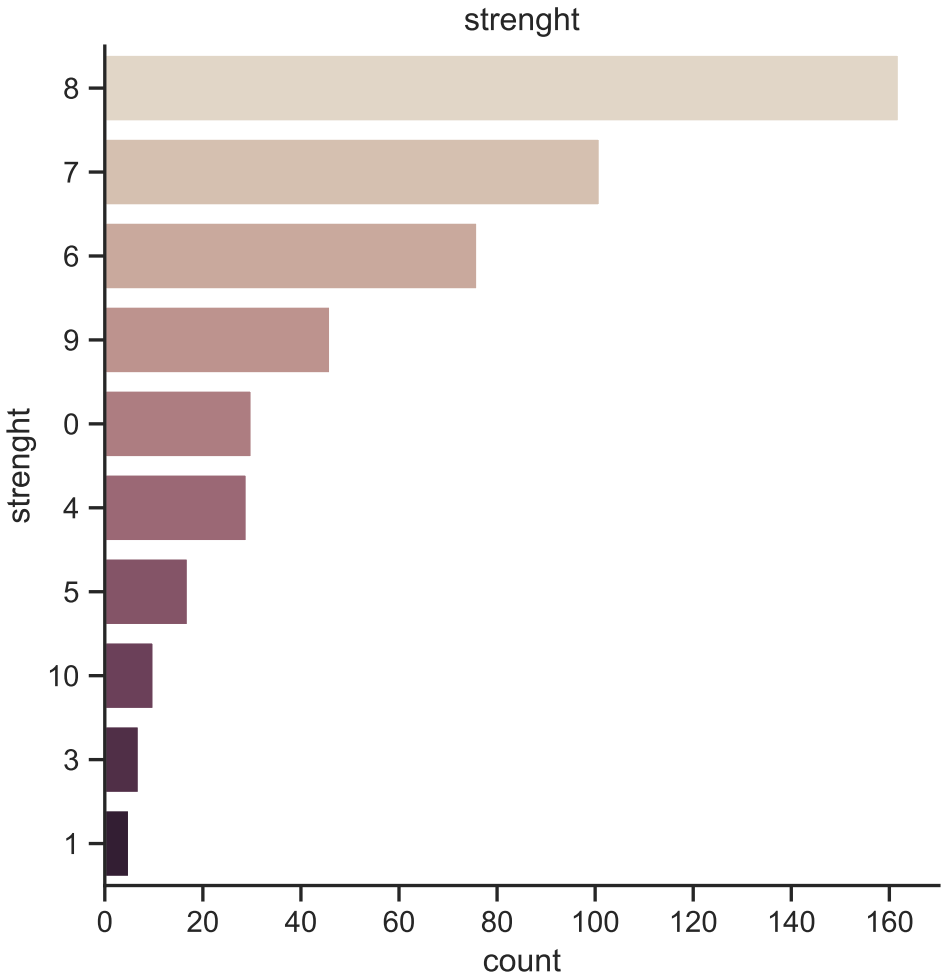
```
['category', 'time_unit', 'strenght']
```

In [17]:

```
for i in CATEGORICAL:

    TOP_10 = df[i].value_counts().iloc[:10].index

    g = sns.catplot(y=i,
            kind="count",
            palette="ch:.25",
            data=df,
            order = TOP_10)

    plt.title(i)
    plt.show();
```


category
name
cool-macho
simple-alphanumeric
fluffy
sport
nerdy-pop
animal
password-related
food
rebellious-rude
category
0
50
100
150
count


time_unit
days
months
years
minutes
hours
seconds
weeks
time_unit
0
50
100
150
200
count


strenght
8
7
6
9
0
4
5
10
3
1
strenght
0
20
40
60
80
100
120
140
160
count

Summary of numerical variables

In [ ]:

```
df.describe()
```