In [ ]:

```
%load_ext autoreload
%autoreload 2
```

In [54]:

```
import calendar
from collections import Counter
from functools import reduce
from operator import itemgetter
from functools import partial

import pandas as pd
import plotly.express as px
import plotly.figure_factory as ff
from mongoengine import connect

from src import settings
from src.data.vacancy import Vacancy
from src.features.clean import remove_html
from src.visualization.statistics import plot_value_counts
```

In [3]:

```
connect(
    host=settings.db_host,
    port=settings.db_port,
    db=settings.db_name
)
```

Out[3]:

```
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True,
read_preference=Primary())
```

In [22]:

```
df: pd.DataFrame = (
    Vacancy
        .objects
        .to_dataframe(include=[
            '_id',
            'name',
            'description',
            'salary',
            'schedule.name',
            'experience',
            'employment.name',
            'area.name',
            'address.lat',
            'address.lng',
            'address.city',
            'published_at',
            'specializations',
            'employer.name',
            'professional_roles',
            'key_skills',
        ])
)
```

In [23]:

```
df.set_index('_id', inplace=True)
```

In [24]:

```
df['description'] = df['description'].map(remove_html)
```

In [25]:

```
df.columns
```

Out[25]:
```
Index(['description', 'key_skills', 'schedule.name', 'experience.id',
       'experience.name', 'employment.name', 'salary.to', 'salary.from',
       'salary.currency', 'salary.gross', 'name', 'area.name', 'published_at',
       'employer.name', 'specializations', 'professional_roles',
       'address.city', 'address.lat', 'address.lng'],
      dtype='object')
```

In [26]:
```
df.shape
```

Out[26]:
```
(63273, 19)
```

In [29]:
```
df.published_at = pd.to_datetime(df.published_at)

count_by_month = {
    calendar.month_name[month]: sum(df.published_at.dt.month == month) for month in rang
}

px.bar(
    x=count_by_month.keys(),
    y=count_by_month.values(),
    labels={'x': 'Месяц', 'y': 'Количество вакансий'},
    title='Количество вакансий в зависимости от месяца'
)
```

In [32]:
```
plot_value_counts(
    df['experience.name'],
    x_label='Опыт',
    y_label='Количество вакансий',
    title='Количество вакансий в зависимости от опыта'
).update_xaxes(categoryorder='total descending')
```

In [33]:

```
plot_value_counts(
    df['schedule.name'],
    x_label='График',
    y_label='Количество вакансий',
    title='Количество вакансий в зависимости от графика работы'
).update_xaxes(categoryorder='total descending')
```

## Анализ навыков

In [34]:

```
key_skills = reduce(set.union, df.key_skills, set())
```

In [35]:

```
len(key_skills)
```

Out[35]:

```
10040
```

In [42]:

```
count_by_key_skill = reduce(Counter.__add__, map(Counter, df.key_skills))
```

In [49]:

```
ff.create_table([('Навык', 'Количество вакансий')] + count_by_key_skill.most_common(50))
```

## Анализ профобластей

In [85]:

```
df['profarea_names'] = df.specializations.map(lambda specs: list(set(map(itemgetter('pro
```

In [86]:

```
df.profarea_names.head(10)
```

Out[86]:

```
_id
49810439                               [Транспорт, логистика]
49810551       [Домашний персонал, Административный персонал]
49810468           [Спортивные клубы, фитнес, салоны красоты]
45788942                               [Транспорт, логистика]
49810601    [Бухгалтерия, управленческий учет, финансы пре...
49810507                          [Административный персонал]
49810469           [Спортивные клубы, фитнес, салоны красоты]
49810426           [Спортивные клубы, фитнес, салоны красоты]
47003369                                            [Продажи]
49810583    [Производство, сельское хозяйство, Медицина, ф...
Name: profarea_names, dtype: object
```

In [87]:

```
profareas = reduce(set.union, df.profarea_names, set())
```

In [88]:

```
len(profareas)
```

Out[88]:

```
28
```

In [91]:

```
count_by_profarea = reduce(Counter.__add__, map(Counter, df.profarea_names))
```

In [105…

```
profareas_df = pd.DataFrame(count_by_profarea, index=['Количество вакансий']).T.reset_in
```

In [106…

```
ff.create_table(profareas_df)
```

In [110…

```
px.bar(
    profareas_df,
    x='Профобласть',
    y='Количество вакансий',
    text_auto='.2s'
).update_xaxes(categoryorder='total descending')
```