In [149…

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

import numpy as np
import pandas as pd
import plotly.express as px
import plotly.offline as pyo
from mongoengine import connect

import src.settings as settings
from src.visualization.statistics import *
from src.features.preprocessing import convert_salary
from src.data.vacancy import Vacancy
```

In [7]:

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

Out[7]:

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

In [8]:

```
pyo.init_notebook_mode()
```

In [9]:

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

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

Получение данных

In [10]:

```
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',
        'specializations',
        'employer.name',
        'professional_roles',
        'key_skills',
    ])
)
```

In [11]:
```
df.set_index('_id', inplace=True)
```

Обработка

Удаление вакансий без зарплаты

In [14]:
```
df['salary.to'].fillna(df['salary.from'], inplace=True)

df = df[df['salary.from'].notna()]
df = df[df['salary.to'].notna()]
df = df[df['salary.currency'].notna()]
```

In [15]:
```
df['salary.currency'].isna().sum()
```

Out[15]:
```
0
```

In [16]:
```
df.shape
```

Out[16]:
```
(47440, 18)
```

Перевод всех зарплат в рубли

In [17]:
```
df[['salary.from', 'salary.to', 'salary.currency']] = df[['salary.from', 'salary.to', 's
    lambda row: [
        convert_salary(row['salary.from'], from_currency=row['salary.currency'], db=sett
        convert_salary(row['salary.to'], from_currency=row['salary.currency'], db=settin
        row['salary.currency']
    ], axis=1, result_type='expand')
```

In [89]:
```
df['mean_salary'] = np.round((df['salary.to'] + df['salary.from']) / 2)
```

In [19]:
```
df[df['salary.currency'] != 'RUR'].head(10)
```

Out[19]:

_id	description	key_skills	schedule.name	experience.id	experience.name	employment.name	
49810443	Требуемый опыт работы: 1–3 года Частичная заня...	[Internet, Голландский язык, Работа в команде,...	Удаленная работа	between1And3	От 1 года до 3 лет	Частичная занятость	8
49226918	Сеть магазинов "Соседи" приглашает на работу з...	[Управление персоналом, Пользователь ПК, Работ...	Сменный график	between1And3	От 1 года до 3 лет	Полная занятость	3
49225403	Обязанности: ручная бережная и качественная м...	[]	Полный день	noExperience	Нет опыта	Полная занятость	2
49225631	Приглашаем водителей для работы в "Такси Алмаз...	[]	Гибкий график	noExperience	Нет опыта	Частичная занятость	4
50155391	Обязанности: - Обработка входящего потока сооб...	[Грамотная речь, Пользователь ПК, Работа в ком...	Удаленная работа	noExperience	Нет опыта	Полная занятость	4
50157216	Please note that by applying to this vacancy y...	[Английский язык, Coaching, Leadership Skills,...	Полный день	noExperience	Нет опыта	Стажировка	6
50155707	Обязанности: Запуск рекламных компаний Google,...	[Английский язык, Маркетинговый анализ, Подгот...	Полный день	between1And3	От 1 года до 3 лет	Полная занятость	26
50156232	Makeomatic расширяет свою команду! Уже 7 лет м...	[Git, JavaScript, Node.js, Docker, GitHub, Dev...	Полный день	between3And6	От 3 до 6 лет	Полная занятость	33
50157118	Топ-менеджер/ Руководитель/ Управление (банковск...	[Работа в команде, CRM, Телефонные переговоры,...	Удаленная работа	between1And3	От 1 года до 3 лет	Полная занятость	33
50157432	Julia Valler Event Staffing is a high-end mode...	[Английский язык, Работа в команде, MS PowerPo...	Полный день	noExperience	Нет опыта	Полная занятость	7

In [90]:

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

Out[90]:

_id	description	key_skills	schedule.name	experience.id	experience.name	employment.nan
49810439	Обязанности: Своевременная подача автомобиля; ...	[]	Полный день	between3And6	От 3 до 6 лет	Полная занятос
49810551	Обязанности: Уборка дома 500 кв.м., стирка, г...	[Русский язык, Чистоплотность]	Полный день	between1And3	От 1 года до 3 лет	Полная занятос
49810468	Студия Красоты и здоровья Кристалл ищет парикм...	[Пользователь ПК, Работа в команде, Грамотная ...	Полный день	between1And3	От 1 года до 3 лет	Полная занятос
45788942	Условия: ЗП от 50 тысяч на руки (оклад 22 тыся...	[Складская логистика, Терминалы Сбора Данных, ...	Сменный график	noExperience	Нет опыта	Полная занятос
49810601	Уважаемые соискатели, рассматриваются кандидат...	[]	Полный день	moreThan6	Более 6 лет	Полная занятос
49810507	Логопедический Пункт 1 приглашает Администрато...	[Обучение персонала, Пользователь ПК, Организа...	Полный день	between1And3	От 1 года до 3 лет	Полная занятос
49810469	Студия Красоты и здоровья Кристалл ищет парикм...	[Пользователь ПК, Работа в команде, Грамотная ...	Полный день	between1And3	От 1 года до 3 лет	Полная занятос
49810426	Обязанности: выполнение услуг массажа на высок...	[антицеллюлитный, класический, спортивный, лим...	Полный день	between3And6	От 3 до 6 лет	Полная занятос
47003369	Медиахолдинг "Май Медиа" ищет менеджера по про...	[Прямые продажи, Телефонные переговоры, Навыки...	Полный день	between1And3	От 1 года до 3 лет	Полная занятос
43592367	Обязанности: Запрос цен и анализ по счетам от...	[MS PowerPoint, MS Access, Работа с базами дан...	Сменный график	noExperience	Нет опыта	Полная занятос

Добавление средней зарплаты

In [21]:
```
total_salary = df['mean_salary'].sum()
total_salary
```

Out[21]:
```
2782462963.154763
```

In [22]:
```
total_salary_by_area = df[['area.name', 'mean_salary']].groupby(['area.name'], as_index=
    columns={'mean_salary': 'total_salary'})
total_salary_by_area.head(10)
```

Out[22]:

	area.name	total_salary
0	Абаза	770000.0
1	Абай	86000.0
2	Абакан	8340594.0
3	Абан	119921.0
4	Абатское	195047.5
5	Абинск	65000.0
6	Авсюнино	131500.0
7	Агалатово	30000.0
8	Агаповка	86000.0
9	Агеево	66000.0

Анализ

Разделение зарплат по городам

In [23]:

```
other = total_salary_by_area.total_salary < (total_salary / 100)  # общая зарплата мены
other_value = total_salary_by_area.total_salary[other].agg('sum')
total_salary_by_area = total_salary_by_area[~other]
total_salary_by_area = total_salary_by_area.append({'area.name': 'Другие регионы', 'tota
                                                    ignore_index=True)
```

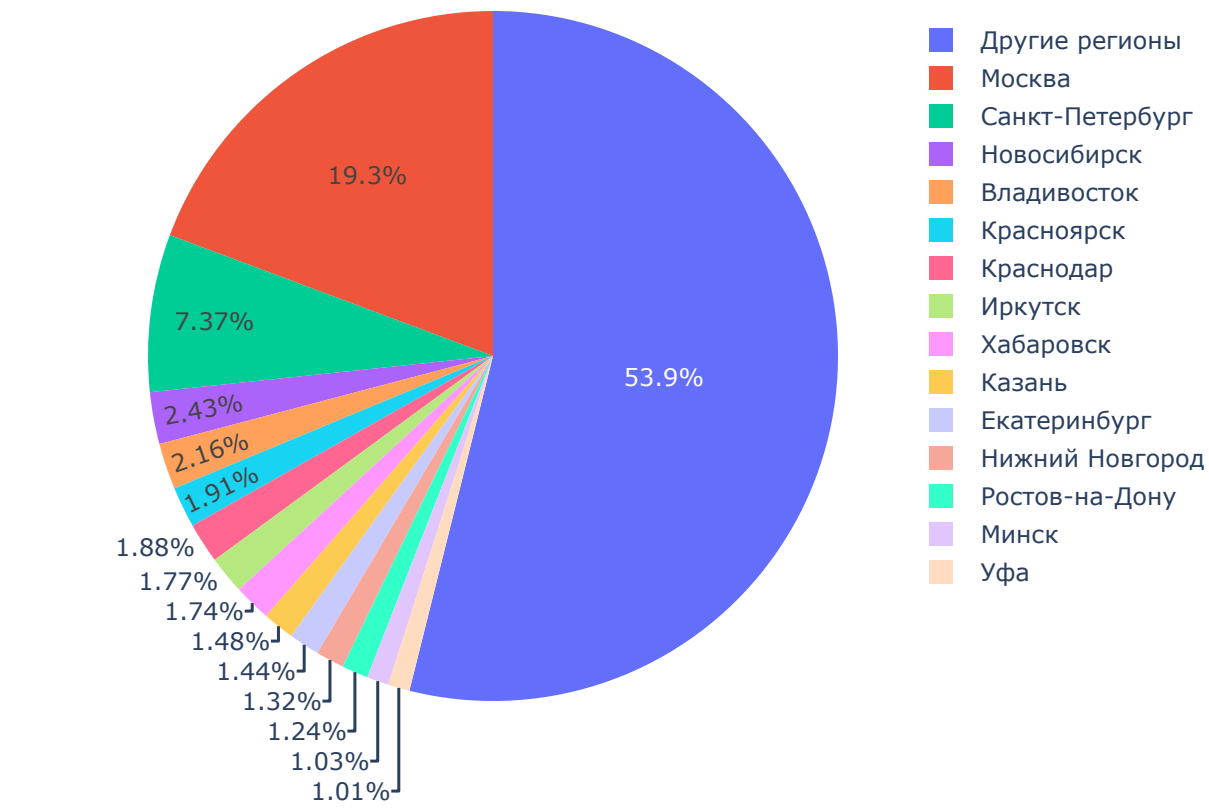
In [24]:

```
total_salary_by_area
```

Out[24]:

	area.name	total_salary
0	Владивосток	6.015153e+07
1	Екатеринбург	4.013486e+07
2	Иркутск	4.925897e+07
3	Казань	4.131740e+07
4	Краснодар	5.243045e+07
5	Красноярск	5.313064e+07
6	Минск	2.859788e+07
7	Москва	5.370685e+08
8	Нижний Новгород	3.684292e+07
9	Новосибирск	6.767413e+07
10	Ростов-на-Дону	3.439070e+07
11	Санкт-Петербург	2.051561e+08
12	Уфа	2.819119e+07
13	Хабаровск	4.851375e+07
14	Другие регионы	1.499604e+09

In [110…]
```
px.pie(total_salary_by_area, names='area.name', values='total_salary', title='Разделение
```

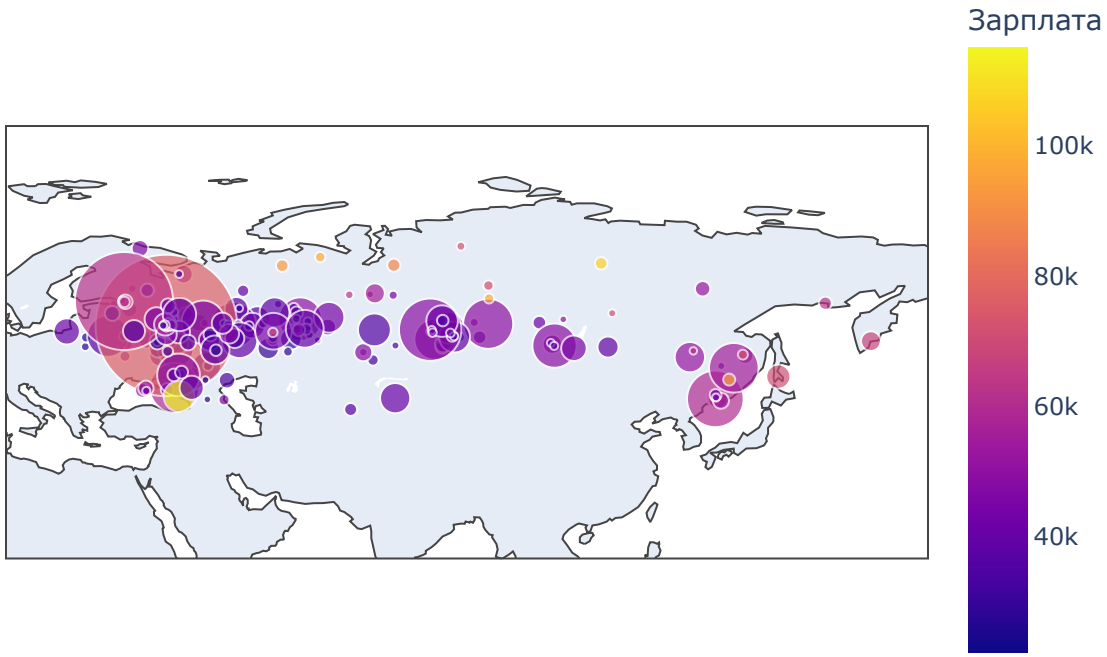


Количество вакансий и средняя зарплата относительно местоположения

In [74]:
```
geo_df = df.reset_index()[['_id', 'address.city', 'address.lat', 'address.lng', 'mean_sa
    .groupby('address.city', as_index=False)\
    .agg({'mean_salary': 'mean', '_id': 'count', 'address.lat': 'mean', 'address.lng': '
    .rename(columns={'_id': 'count'})\
```

In [106…]
```
px.scatter_geo(
    geo_df[(geo_df['count'] > 10) & (geo_df['mean_salary'] < 200_000)],
    lat='address.lat',
    lon='address.lng',
    size='count',
    fitbounds='locations',
    color='mean_salary',
    hover_data=['address.city'],
    center={'lat': 53, 'lon': 83},
    size_max=50,
    labels={'mean_salary': 'Зарплата'},
    title='Количество вакансий и средняя зарплата относительно города'
)
```

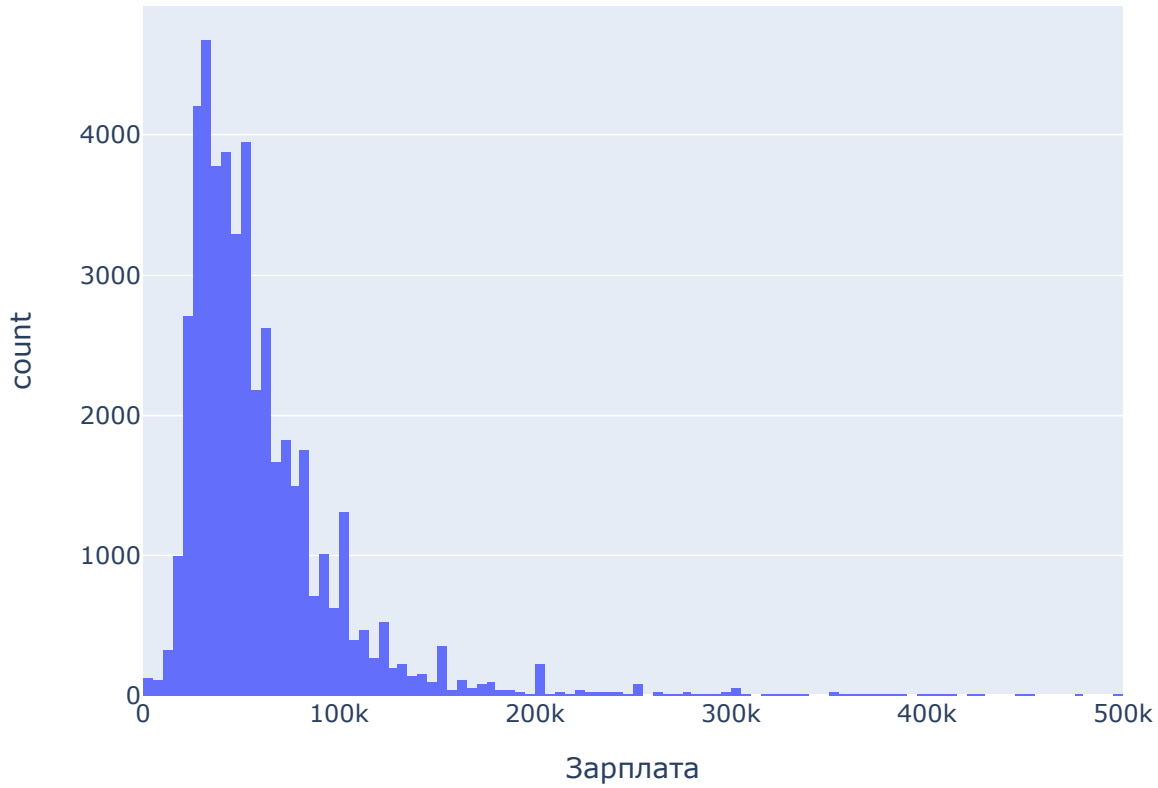
Количество вакансий и средняя зарплата относительно города



Распределения

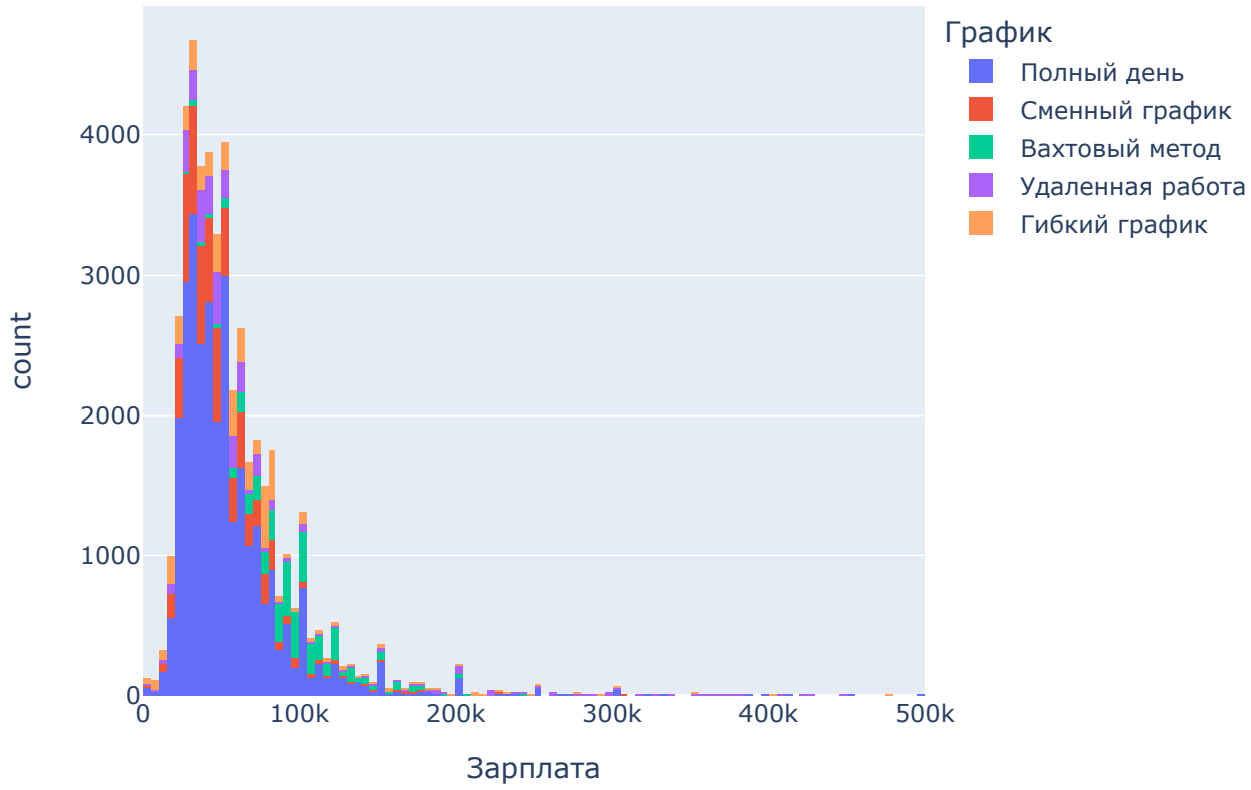
In [116…

```
px.histogram(
    df[df.mean_salary < 500_000],
    x='mean_salary',
    nbins=100,
    title='Распределение зарплат',
    labels={'mean_salary': 'Зарплата'}
)
```



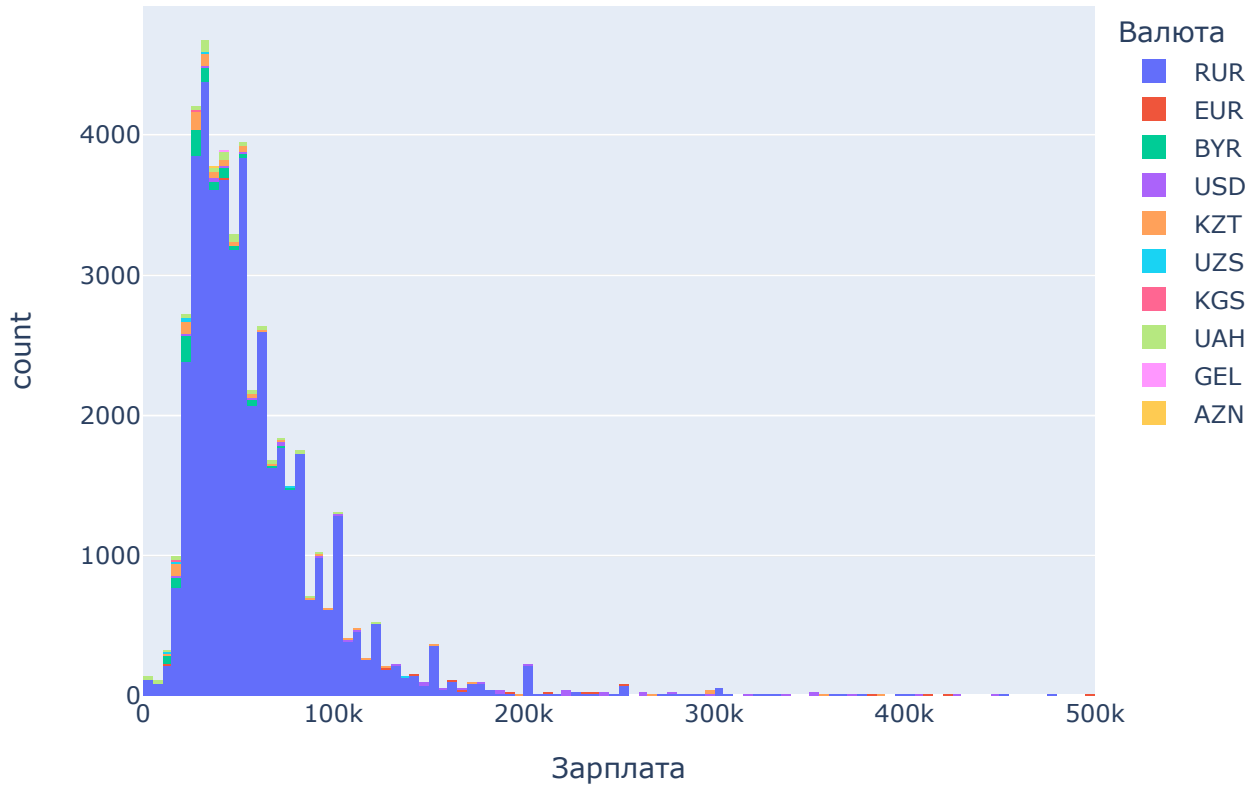
In [114…

```
px.histogram(
    df[df.mean_salary < 500_000],
    x='mean_salary',
    color='schedule.name',
    nbins=100,
    title='Распределение зарплат с учетом графика работы',
    labels={'mean_salary': 'Зарплата', 'schedule.name': 'График'}
)
```



In [115…

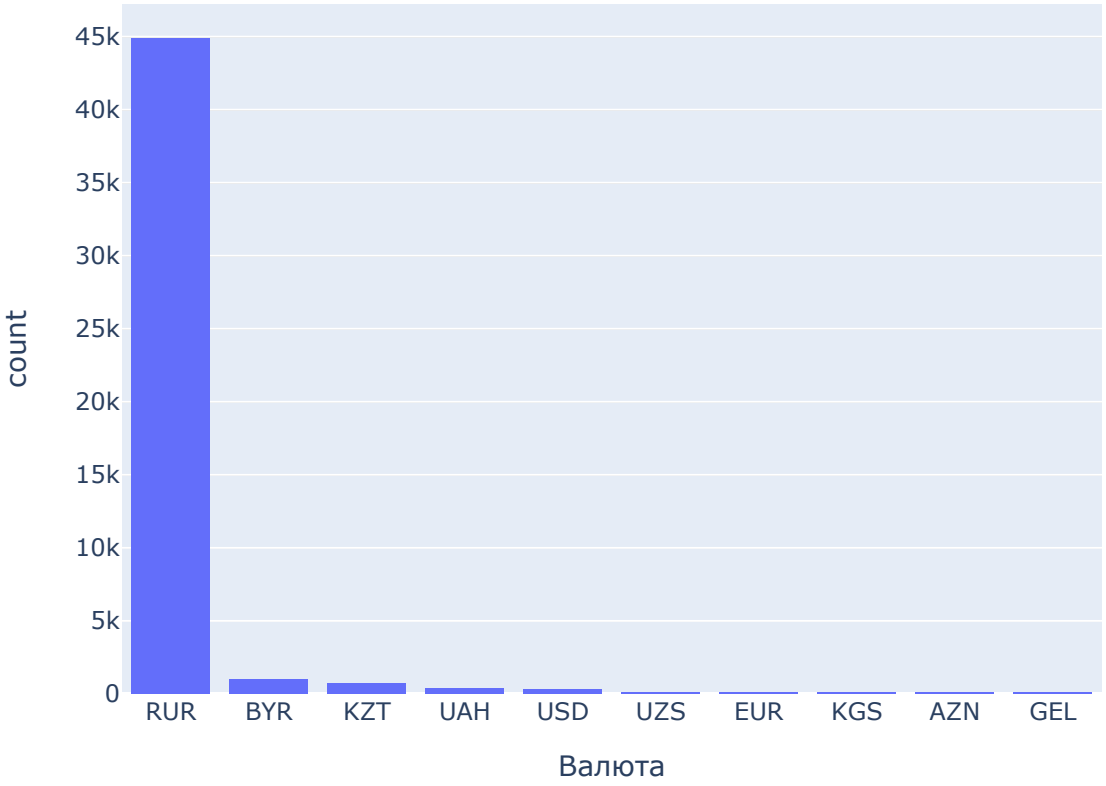
```
px.histogram(
    df[df.mean_salary < 500_000],
    x='mean_salary',
    color='salary.currency',
    nbins=100,
    title='Распределение зарплат с учетом валюты работы',
    labels={'mean_salary': 'Зарплата', 'salary.currency': 'Валюта'}
)
```



In [122…

```
px.histogram(
    df,
    x='salary.currency',
    title='Количество вакансий для каждой валюты',
    labels={'salary.currency': 'Валюта'}
).update_xaxes(categoryorder='total descending')
```

Количество вакансий для каждой валюты



Анализ зарплат относительно профобластей

In [245…

```
df_specs = df.copy()
df_specs.specializations = df_specs.specializations.map(lambda specs: list(map(itemgette
df_specs = df_specs[df_specs.specializations.notna()]
df_specs['specialization_profarea_names'] = df.specializations.map(lambda specs: list(se
df_specs = df_specs[df_specs.specialization_profarea_names.notna()]
```

In [246…

```
df_specs[['specialization_profarea_names', 'specializations']].head(10)
```

Out[246]:

_id	specialization_profarea_names	specializations
49810439	[Транспорт, логистика]	[Автоперевозки, Водитель, Логистика, Экспедитор]
49810551	[Административный персонал, Домашний персонал]	[Уборщица/уборщик, домработница/домработник, Г...
49810468	[Спортивные клубы, фитнес, салоны красоты]	[Парикмахер]
45788942	[Транспорт, логистика]	[Кладовщик, Рабочий склада, Логистика]
49810601	[Банки, инвестиции, лизинг, Бухгалтерия, управ...	[Учет заработной платы, Основные средства, Нал...
49810507	[Административный персонал]	[Управляющий офисом (Office manager), Персонал...
49810469	[Спортивные клубы, фитнес, салоны красоты]	[Парикмахер]
49810426	[Спортивные клубы, фитнес, салоны красоты]	[Массажист]
47003369	[Продажи]	[Прямые продажи, Менеджер по работе с клиентами]
43592367	[Бухгалтерия, управленческий учет, финансы пре...	[Аудит, Другое, Финансовый анализ]

In [247...

```
all_specializations = list(reduce(set.union, df_specs.specializations, set()))
all_specializations[:15]
```

Out[247]:

```
['Мебельное производство',
 'CRM системы',
 'Акции, Ценные бумаги',
 'Железнодорожные перевозки',
 'Оптимизация сайта (SEO)',
 'Кассир, Инкассатор',
 'Информационные технологии, Интернет, Мультимедиа',
 'Управляющий офисом (Office manager)',
 'Арт-директор',
 'Секретарь',
 'Машинист производства',
 'Закупки и снабжение',
 'Диспетчер',
 'Персональный ассистент',
 'Компьютерная безопасность']
```

In [248...

```
len(all_specializations)
```

Out[248]:

```
504
```

In [249...

```
count_by_specialization = {spec: df_specs.specializations.map({spec}.issubset).sum() for
count_by_specialization = Counter(count_by_specialization)
count_by_specialization.most_common(10)
```

Out[249]:

```
[('Розничная торговля', 10704),
 ('Начальный уровень, Мало опыта', 9135),
 ('Торговые сети', 8017),
 ('Продукты питания', 7397),
 ('Продажи', 6681),
 ('Продавец в магазине', 6055),
 ('Менеджер по работе с клиентами', 5314),
 ('Прямые продажи', 4366),
 ('Другое', 3127),
 ('Водитель', 2872)]
```

In [250…
```
all_profareas = reduce(set.union, df_specs.specialization_profarea_names, set())
```

In [251…
```
len(all_profareas)
```

Out[251]:
```
28
```

In [242…
```
{profarea: df_specs[df_specs.specialization_profarea_names.map({profarea}.issubset)]['me
```

Out[242]:
```
{'Административный персонал': 48667.45219370861,
 'Строительство, недвижимость': 84771.85741049125,
 'Туризм, гостиницы, рестораны': 47756.217606707316,
 'Информационные технологии, интернет, телеком': 92008.00538176925,
 'Инсталляция и сервис': 61575.194444444445,
 'Искусство, развлечения, масс-медиа': 56821.15873015873,
 'Высший менеджмент': 123710.24584717608,
 'Автомобильный бизнес': 72177.20721925133,
 'Производство, сельское хозяйство': 64457.76760048721,
 'Управление персоналом, тренинги': 65502.94584382872,
 'Транспорт, логистика': 67427.33646295663,
 'Бухгалтерия, управленческий учет, финансы предприятия': 48283.09805153991,
 'Юристы': 55641.35897435898,
 'Наука, образование': 45231.578881987574,
 'Государственная служба, некоммерческие организации': 52380.44400785855,
 'Добыча сырья': 94832.55590062111,
 'Безопасность': 53167.152825836216,
 'Спортивные клубы, фитнес, салоны красоты': 51376.46395250212,
 'Рабочий персонал': 64773.36851851852,
 'Медицина, фармацевтика': 55659.44958753437,
 'Маркетинг, реклама, PR': 60350.047784967646,
 'Консультирование': 92617.73353751915,
 'Страхование': 72418.31764705882,
 'Закупки': 62442.23834886817,
 'Банки, инвестиции, лизинг': 57242.99230111206,
 'Домашний персонал': 41872.405241935485,
 'Продажи': 49176.52389049481,
 'Начало карьеры, студенты': 43474.821203244705}
```

In [276…
```
df_profarea = pd.DataFrame({
    profarea: df_specs[
        df_specs.specialization_profarea_names
            .map({profarea}.issubset)
    ]['mean_salary'].agg(['mean', 'sum', 'count'])
    for profarea in all_profareas
}).T
df_profarea = df_profarea.astype(np.int64).reset_index().rename(columns={'index': 'profa
```

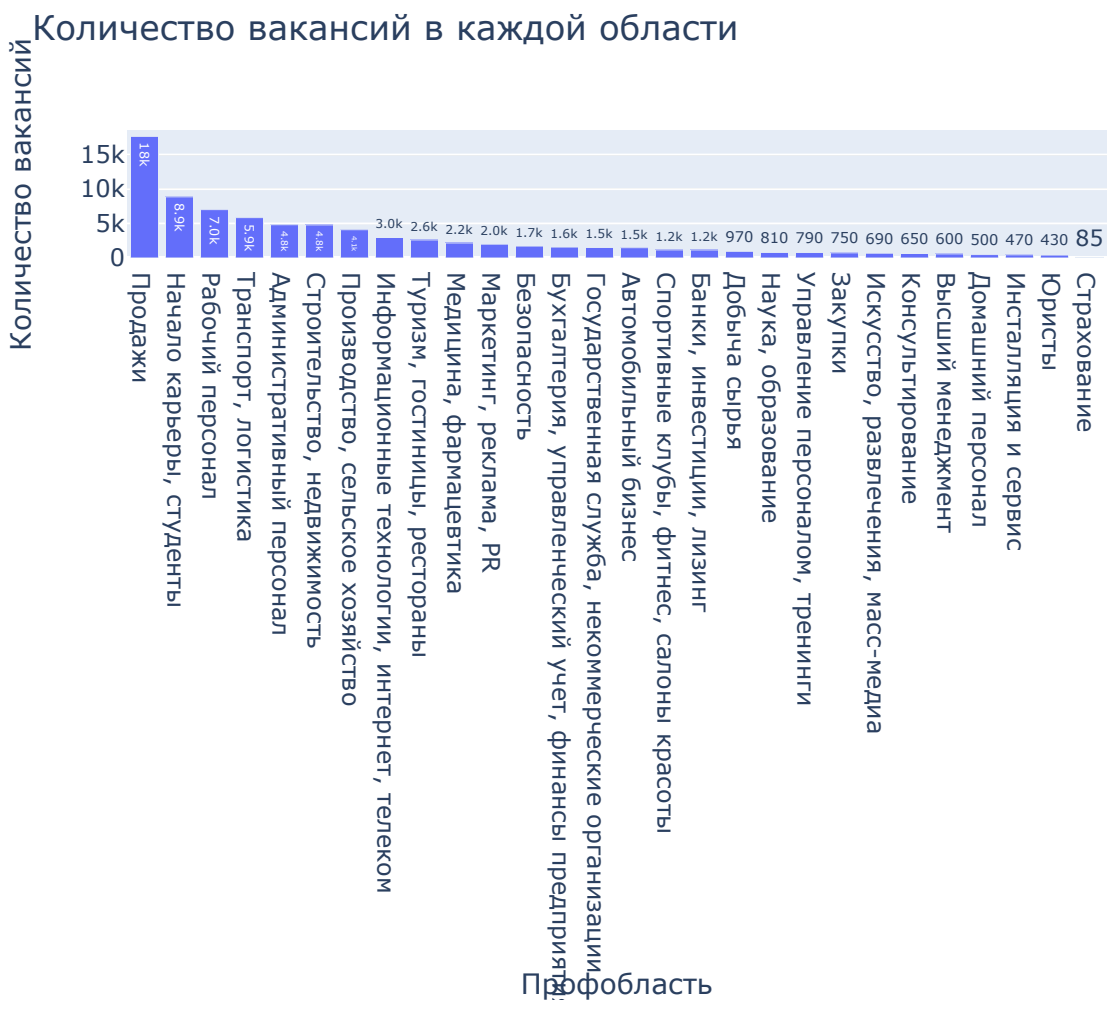
In [277…
```
df_profarea.head(10)
```

Out[277]:

	profarea	mean	sum	count
0	Административный персонал	48667	235161129	4832
1	Строительство, недвижимость	84771	407244003	4804
2	Туризм, гостиницы, рестораны	47756	125312315	2624
3	Информационные технологии, интернет, телеком	92008	273539800	2973
4	Инсталляция и сервис	61575	28817191	468
5	Искусство, развлечения, масс-медиа	56821	39377063	693
6	Высший менеджмент	123710	74473568	602
7	Автомобильный бизнес	72177	107977102	1496
8	Производство, сельское хозяйство	64457	264599136	4105
9	Управление персоналом, тренинги	65502	52009339	794

In [284…

```
px.bar(
    df_profarea,
    x='profarea',
    y='count',
    labels={'profarea': 'Профобласть', 'count': 'Количество вакансий'},
    text_auto='.2s',
    title='Количество вакансий в каждой области'
).update_xaxes(categoryorder='total descending')
```



In [282…

```
px.pie(
    df_profarea,
    names='profarea',
    values='count',
    labels={'index': 'Профобласть', 'count': 'Количество вакансий'},
    title='Доля вакансий для каждой области'
)
```

In [285…

```
px.bar(
    df_profarea,
    x='profarea',
    y='mean',
    labels={'profarea': 'Профобласть', 'mean': 'Средняя зарплата'},
    text_auto='.2s',
    title='Средняя зарплата в каждой области'
).update_xaxes(categoryorder='total descending')
```

In [286…

```
px.bar(
    df_profarea,
    x='profarea',
    y='sum',
    labels={'profarea': 'Профобласть', 'sum': 'Сумма всех зарплат'},
    text_auto='.2s',
    title='Сумма зарплат в каждой области'
).update_xaxes(categoryorder='total descending')
```

Сумма зарплат в каждой области
Сумма всех зарплат
800M
600M
400M
200M
0
870M
450M
410M
390M
390M
270M
260M
240M
130M
120M
120M
110M
92M
92M
80M
77M
74M
67M
61M
60M
52M
47M
39M
36M
29M
24M
21M
6.2M
Продажи
Рабочий персонал
Строительство, недвижимость
Транспорт, логистика
Начало карьеры, студенты
Информационные технологии, интернет, телеком
Производство, сельское хозяйство
Административный персонал
Туризм, гостиницы, рестораны
Медицина, фармацевтика
Маркетинг, реклама, PR
Автомобильный бизнес
Безопасность
Добыча сырья
Государственная служба, некоммерческие организации
Бухгалтерия, управленческий учет, финансы предприятия
Высший менеджмент
Банки, инвестиции, лизинг
Спортивные клубы, фитнес, салоны красоты
Консультирование
Управление персоналом, тренинги
Закупки
Искусство, развлечения, масс-медиа
Наука, образование
Инсталляция и сервис
Юристы
Домашний персонал
Страхование
Профобласть