

# Аналитика данных с помощью pandas и matplotlib

В этом задании вам предлагается выполнить анализ данных криптовалют с помощью библиотек pandas и matplotlib. Задание выглядит как лабораторная работа, в которой вам предстоит заполнить недостающие клетки и ответить на ряд вопросов.

## 1. Данные (суммарно 2 балла)

Начнем с необходимых приготовлений.

```
In [5]: import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import ipywidgets # Библиотека для интерактивных контролов в jupyter notebook

%matplotlib inline
```

Загрузите заранее подготовленный датасет из файла "coins.csv". Создайте объект типа pandas.DataFrame с именем coins и в качестве индекса выберите колонку с датой.

```
In [174]: coins = pd.read_csv('coins.csv', index_col = 'date')
```

Посмотрим что получилось

```
In [175]: coins.head(4)
```

```
Out[175]:
```

	price	txCount	txVolume	activeAddresses	symbol	name	open	high	low	
date										
2013-04-28	135.30	41702.0	6.879868e+07	117984.0	BTC	Bitcoin	135.30	135.98	132.10	1
2013-04-28	4.30	9174.0	4.431952e+07	17216.0	LTC	Litecoin	4.30	4.40	4.18	
2013-04-29	134.44	51602.0	1.138128e+08	86925.0	BTC	Bitcoin	134.44	147.49	134.00	1
2013-04-29	4.37	9275.0	3.647810e+07	18395.0	LTC	Litecoin	4.37	4.57	4.23	

Поясним значения хранящиеся в колонках

- date - дата измерений
- name - полное название монеты
- symbol - сокращенное название монеты
- price - средняя цена монеты за торговый день в USD
- txCount - количество транзакций в сети данной монеты
- txVolume - объем монет переведенных между адресами в сети данной монеты
- activeAddresses - количество адресов совершавших а данный день транзакции в сети данной монеты
- open - цена монеты в начале торгов данного дня
- close - цена монеты в конце торгов данного дня
- high - самая высокая цена данной монеты в течение данного торгового дня
- low - самая низкая цена данной монеты в течение данного торгового дня
- volume - объем торгов данной монетой на биржах в данный день
- market - капитализация данной монеты в данный день

Изучим полученные данные. Ответьте на следующие вопросы (вставляйте клетки с кодом и текстом ниже):

1. Сколько всего различных монет представлено в датасете? (0.4 балла)

```
In [176]: len(set(coins['name']))
```

Out[176]: 70

2. За какой период данные мы имеем? (0.4 балла)

```
In [177]: print(coins.index.min())
print(coins.index.max())
```

2013-04-28  
2018-06-06

3. Есть ли пропуски в данных? Какой природы эти пропуски? (0.5 балла)

```
In [180]: if coins.isnull().values.any():
print("Coins includes missings", "\n")
else:
print("Coins doesn't include missings", "\n")
l = list(coins.columns)
for elem in l:
if coins[str(elem)].isnull().values.any() != 0:
count = coins[str(elem)].isnull().sum()
print("{} column {} has missings".format(elem, count))
```

Coins includes missings

price column 327 has missings  
txCount column 1520 has missings  
txVolume column 1830 has missings  
activeAddresses column 1520 has missings

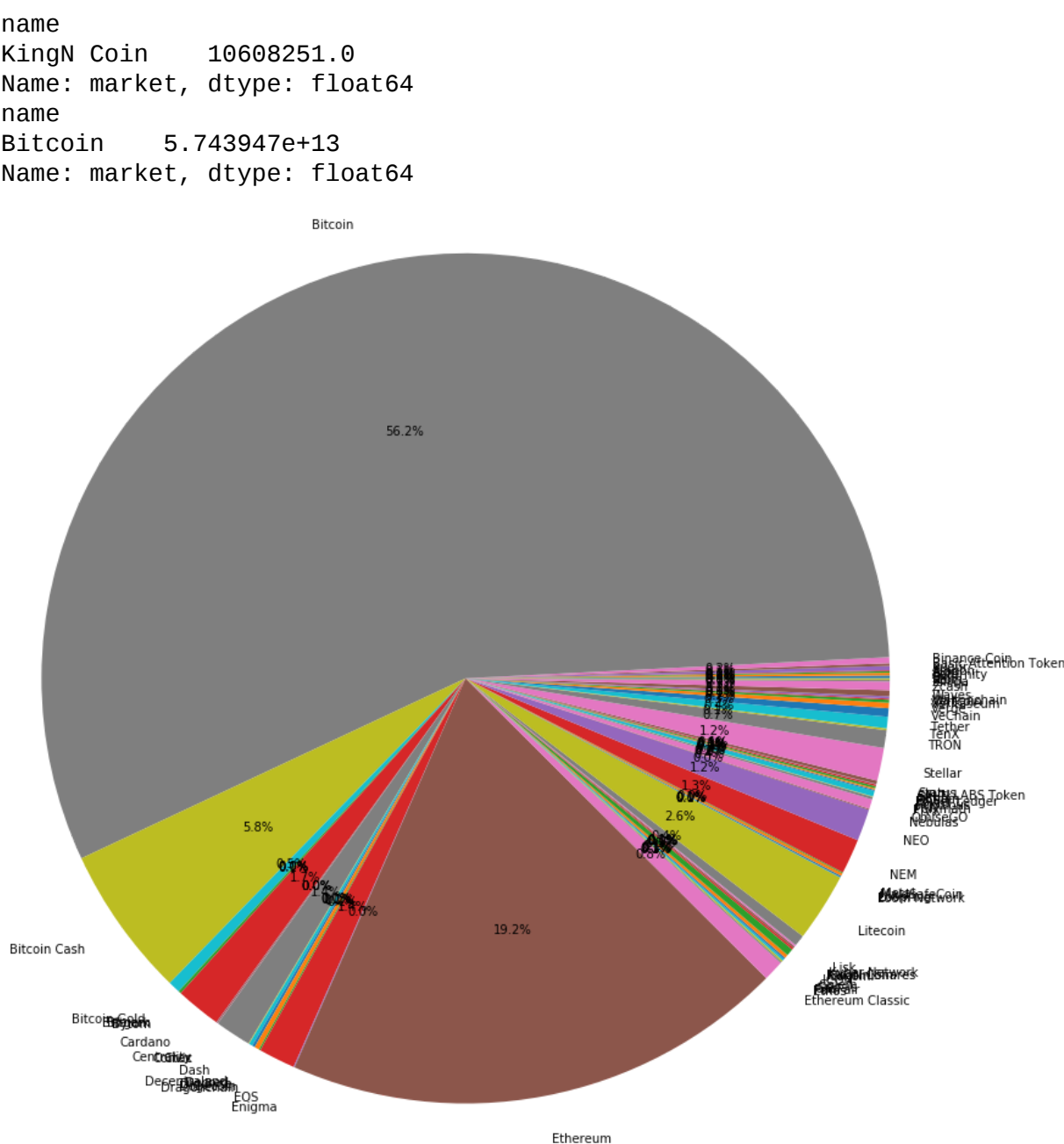
4. У какой монеты и когда была самая высокая цена? (0.2 балла)

```
In [10]: print(coins[coins['price'] == coins['price'].max()][ 'name' ])
```

date  
2017-12-17 Bitcoin  
Name: name, dtype: object

5. У какой монеты самая высокая и самая низкая суммарная капитализация? Постройте круговую диаграмму с долями. (0.5 балла)

```
In [196]: sums = coins.market.groupby(coins.name).sum()
print(sums[sums[sums.index] == sums.min()])
print(sums[sums[sums.index] == sums.max()])
plt.pie(sums, autopct='%1.1f%%', labels=sums.index, radius = 4);
plt.show()
```



## 2. Визуализация (1 балл)

Реализуйте функцию для визуализации цен выбранной валюты за выбранный диапазон дат.

На графике должны быть видны цены начала и конца продаж. А так же минимальная и максимальная цена за этот день. Подпишите график и оси координат. Добавьте сетку. Увеличьте размер изображения. Можете попробовать использовать matplotlib.finance.candlestick\_ohlc (mpl\_finance.candlestick\_ohlc), но можно и без него.

```
In [148]: def plot_fancy_price_action(coins, symbol, start_date, end_date):
coins.index = pd.to_datetime(coins.index)
coin = coins.loc[(coins.index >= start_date)&(coins.index <= end_date)&(coins['symbol'] == symbol), ['high', 'low', 'open', 'close']]
plot1 = plt.figure(figsize=(16, 9), dpi = 200).add_subplot(111)
coin.plot(ax=plot1, grid = True, figsize=(16, 9), fontsize = 12, label='line values', linewidth=1,)
coin.plot(ax=plot1, grid = True, legend = True, label='point values', style = 'o')
plt.title('Variation of prices during chosen time', fontsize=20)
plt.xlabel('DATE', fontdict=dict(family='serif', color='darkred', weight='normal', size=16))
plt.ylabel('PRICE', fontdict=dict(family='monospace', color='peru', weight='light', size=25))
```

Посмотрим, что получилось:

```
In [149]: plot_fancy_price_action(coins=coins, symbol='VERI', start_date='2013-06-01', end_date='2019-06-30')
```

