```
# This Python program automatically collects cypto currencies
market data from CoinMarketCap
# website using its API. And it cleans the data and generates
charts.

 #This example uses Python 2.7 and the python-request library.

from requests import Request, Session
from requests.exceptions import ConnectionError, Timeout,
TooManyRedirects
import json

url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/
listings/latest'
parameters = {
  'start':'1',
  'limit':'10',
  # 'limit':'5000',
  'convert':'USD'
}
headers = {
  'Accepts': 'application/json',
  'X-CMC_PRO_API_KEY': 'xxxxxxxxxxxxxxxxxx',
}

session = Session()
session.headers.update(headers)

try:
  response = session.get(url, params=parameters)
  data = json.loads(response.text)
  #print(data)
except (ConnectionError, Timeout, TooManyRedirects) as e:
  print(e)

# had to use terminal to put "jupyter notebook --
NotebookApp.iopub_data_rate_limit=1e10"
# into Anaconda Prompt to change hits to allow to pull data

# check data type
type(data)

# use pandas to handle dictionary data
import pandas as pd

pd.json_normalize(data['status'])
```

```python
pd.json_normalize(data['data'])

# display all columns
pd.set_option('display.max_columns', None)
pd.json_normalize(data['data'])


# add data pulling timestamp
df = pd.json_normalize(data['data'])
df['timestamp'] = pd.to_datetime('now')
df
```

Out[3]:

	id	name	symbol	slug	num_market_pairs	date_added	tags	max_supply	circulating_supply	total_supply	platform	cmc_rank
0	1	Bitcoin	BTC	bitcoin	9914	2013-04-28T00:00:00.000Z	[mineable, pow, sha-256, store-of-value, state...	2.100000e+07	1.924984e+07	1.924984e+07	NaN	1
1	1027	Ethereum	ETH	ethereum	6328	2015-08-07T00:00:00.000Z	[pos, smart-contracts, ethereum-ecosystem, coi...	NaN	1.223739e+08	1.223739e+08	NaN	2
2	825	Tether	USDT	tether	45697	2015-02-25T00:00:00.000Z	[payments, stablecoin, asset-backed-stablecoin...	NaN	6.626371e+10	7.314177e+10	NaN	3
3	3408	USD Coin	USDC	usd-coin	9365	2018-10-08T00:00:00.000Z	[medium-of-exchange, stablecoin, asset-backed-...	NaN	4.461419e+10	4.461419e+10	NaN	4
4	1839	BNB	BNB	bnb	1164	2017-07-25T00:00:00.000Z	[marketplace, centralized-exchange, payments, ...	2.000000e+08	1.599645e+08	1.599800e+08	NaN	5
5	52	XRP	XRP	xrp	871	2013-08-04T00:00:00.000Z	[medium-of-exchange, enterprise-solutions, arr...	1.000000e+11	5.034350e+10	9.998918e+10	NaN	6
6	4687	Binance USD	BUSD	binance-usd	5942	2019-09-20T00:00:00.000Z	[stablecoin, asset-backed-stablecoin, binance-...	NaN	1.651448e+10	1.651448e+10	NaN	7
7	74	Dogecoin	DOGE	dogecoin	629	2013-12-15T00:00:00.000Z	[mineable, pow, scrypt, medium-of-exchange, me...	NaN	1.326708e+11	1.326708e+11	NaN	8
8	2010	Cardano	ADA	cardano	624	2017-10-01T00:00:00.000Z	[dpos, pos, platform, research, smart-	4.500000e+10	3.450985e+10	3.528563e+10	NaN	9

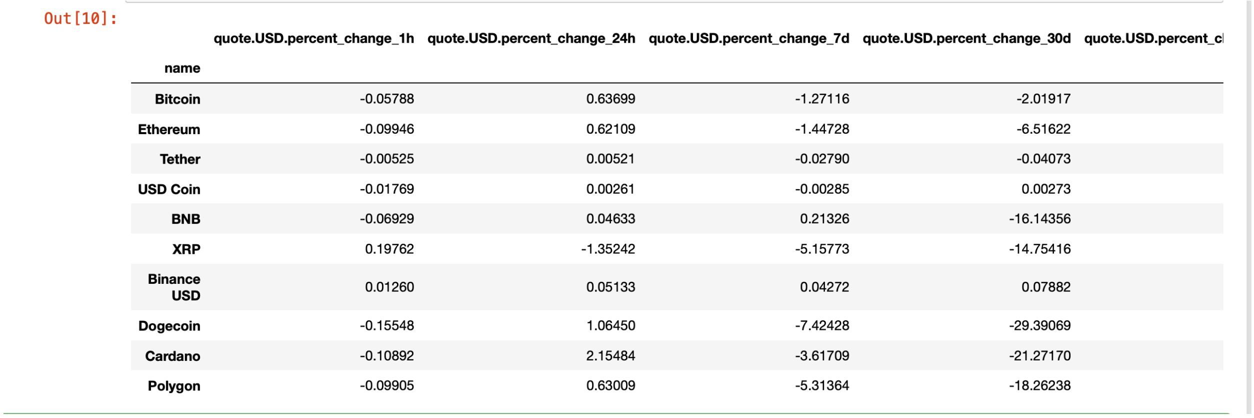
```
def api_runner():
  global df

  url = 'https://pro-api.coinmarketcap.com/v1/cryptocurrency/listings/latest'
  parameters = {
    'start':'1',
    'limit':'10',
    # 'limit':'5000',
    'convert':'USD'
  }
  headers = {
    'Accepts': 'application/json',
    'X-CMC_PRO_API_KEY': 'xxxxxxxxx',
  }

  session = Session()
  session.headers.update(headers)

  try:
    response = session.get(url, params=parameters)
    data = json.loads(response.text)
    #print(data)
  except (ConnectionError, Timeout, TooManyRedirects) as e:
    print(e)

# had to use terminal to put "jupyter notebook --NotebookApp.iopub_data_rate_limit=1e10"
# into Anaconda Prompt to change hits to allow to pull data


  # Use this if you just want to keep it in a dataframe
  # df2 = pd.json_normalize(data['data'])
  # df2['timestamp'] = pd.to_datetime('now')
  # df = df.append(df2)

  # Use this if you want to create a csv and append data to it
  df = pd.json_normalize(data['data'])
  df['timestamp'] = pd.to_datetime('now')
  df
  # save to csv
  if not os.path.isfile(r'/Users/jan/desktop/data/jy/future/codesamples/python/CoinMarketCap/AutomatingCryptoWebsiteAPIPull/api.csv'):
    df.to_csv(r'/Users/jan/desktop/data/jy/future/codesamples/python/CoinMarketCap/AutomatingCryptoWebsiteAPIPull/api.csv', header='column_names')
  else:
   df.to_csv(r'/Users/jan/desktop/data/jy/future/codesamples/python/CoinMarketCap/AutomatingCryptoWebsiteAPIPull/api.csv', mode='a', header=False)



import os
from time import time
from time import sleep

for i in range(333):
  api_runner()
  print('API runner completed')
  sleep(60) # sleep for i minutes
exit()

df

pd.set_option('display.max_rows', None)
df
```

Out[9]:

	id	name	symbol	slug	num_market_pairs	date_added	tags	max_supply	circulating_supply	total_supply	platform	cmc_rank
0	1	Bitcoin	BTC	bitcoin	9914	2013-04-28T00:00:00.000Z	[mineable, pow, sha-256, store-of-value, state...	2.100000e+07	1.924984e+07	1.924984e+07	NaN	1
1	1027	Ethereum	ETH	ethereum	6328	2015-08-07T00:00:00.000Z	[pos, smart-contracts, ethereum-ecosystem, coi...	NaN	1.223739e+08	1.223739e+08	NaN	2
2	825	Tether	USDT	tether	45697	2015-02-25T00:00:00.000Z	[payments, stablecoin, asset-backed-stablecoin...	NaN	6.626371e+10	7.314177e+10	NaN	3
3	3408	USD Coin	USDC	usd-coin	9365	2018-10-08T00:00:00.000Z	[medium-of-exchange, stablecoin, asset-backed-...	NaN	4.461419e+10	4.461419e+10	NaN	4
4	1839	BNB	BNB	bnb	1164	2017-07-25T00:00:00.000Z	[marketplace, centralized-exchange, payments, ...	2.000000e+08	1.599645e+08	1.599800e+08	NaN	5
5	52	XRP	XRP	xrp	871	2013-08-04T00:00:00.000Z	[medium-of-exchange, enterprise-solutions, arr...	1.000000e+11	5.034350e+10	9.998918e+10	NaN	6
6	4687	Binance USD	BUSD	binance-usd	5942	2019-09-20T00:00:00.000Z	[stablecoin, asset-backed-stablecoin, binance-...	NaN	1.651448e+10	1.651448e+10	NaN	7
7	74	Dogecoin	DOGE	dogecoin	629	2013-12-15T00:00:00.000Z	[mineable, pow, scrypt, medium-of-exchange, me...	NaN	1.326708e+11	1.326708e+11	NaN	8
							[dpos, pos,					

click to scroll output; double click to hide

```
df72 = pd.read_csv(r'/Users/jan/desktop/data/jy/future/
codesamples/python/CoinMarketCap/AutomatingCryptoWebsiteAPIPull/
api.csv')
df72

# change float display format
pd.set_option('display.float_format', lambda x: '%.5f' % x)

df3 = df.groupby('name', sort=False)
[['quote.USD.percent_change_1h','quote.USD.percent_change_24h','
quote.USD.percent_change_7d','quote.USD.percent_change_30d','quo
te.USD.percent_change_60d','quote.USD.percent_change_90d']].mean
()
df3
```



Out[10]:

	quote.USD.percent_change_1h	quote.USD.percent_change_24h	quote.USD.percent_change_7d	quote.USD.percent_change_30d	quote.USD.percent_cl
name					
Bitcoin	-0.05788	0.63699	-1.27116	-2.01917	
Ethereum	-0.09946	0.62109	-1.44728	-6.51622	
Tether	-0.00525	0.00521	-0.02790	-0.04073	
USD Coin	-0.01769	0.00261	-0.00285	0.00273	
BNB	-0.06929	0.04633	0.21326	-16.14356	
XRP	0.19762	-1.35242	-5.15773	-14.75416	
Binance USD	0.01260	0.05133	0.04272	0.07882	
Dogecoin	-0.15548	1.06450	-7.42428	-29.39069	
Cardano	-0.10892	2.15484	-3.61709	-21.27170	
Polygon	-0.09905	0.63009	-5.31364	-18.26238	

```python
# stack values
df4 = df3.stack()
df4
```

Out[11]:

```
name
Bitcoin      quote.USD.percent_change_1h     -0.05788
             quote.USD.percent_change_24h     0.63699
             quote.USD.percent_change_7d     -1.27116
             quote.USD.percent_change_30d    -2.01917
             quote.USD.percent_change_60d   -18.14344
             quote.USD.percent_change_90d   -16.15577
Ethereum     quote.USD.percent_change_1h     -0.09946
             quote.USD.percent_change_24h     0.62109
             quote.USD.percent_change_7d     -1.44728
             quote.USD.percent_change_30d    -6.51622
             quote.USD.percent_change_60d   -22.46788
             quote.USD.percent_change_90d   -10.25439
Tether       quote.USD.percent_change_1h     -0.00525
             quote.USD.percent_change_24h     0.00521
             quote.USD.percent_change_7d     -0.02790
             quote.USD.percent_change_30d    -0.04073
             quote.USD.percent_change_60d    -0.03054
             quote.USD.percent_change_90d    -0.05221
USD Coin     quote.USD.percent_change_1h     -0.01769
             quote.USD.percent_change_24h     0.00261
             quote.USD.percent_change_7d     -0.00285
             quote.USD.percent_change_30d     0.00273
             quote.USD.percent_change_60d    -0.02733
             quote.USD.percent_change_90d    -0.00868
BNB          quote.USD.percent_change_1h     -0.06929
             quote.USD.percent_change_24h     0.04633
             quote.USD.percent_change_7d      0.21326
             quote.USD.percent_change_30d   -16.14356
             quote.USD.percent_change_60d   -26.20724
             quote.USD.percent_change_90d   -15.67650
XRP          quote.USD.percent_change_1h      0.19762
             quote.USD.percent_change_24h    -1.35242
             quote.USD.percent_change_7d     -5.15773
             quote.USD.percent_change_30d   -14.75416
             quote.USD.percent_change_60d   -27.51836
             quote.USD.percent_change_90d   -27.54355
Binance USD  quote.USD.percent_change_1h      0.01260
             quote.USD.percent_change_24h     0.05133
             quote.USD.percent_change_7d      0.04272
             quote.USD.percent_change_30d     0.07882
             quote.USD.percent_change_60d    -0.01436
             quote.USD.percent_change_90d    -0.03004
Dogecoin     quote.USD.percent_change_1h     -0.15548
             quote.USD.percent_change_24h     1.06450
```

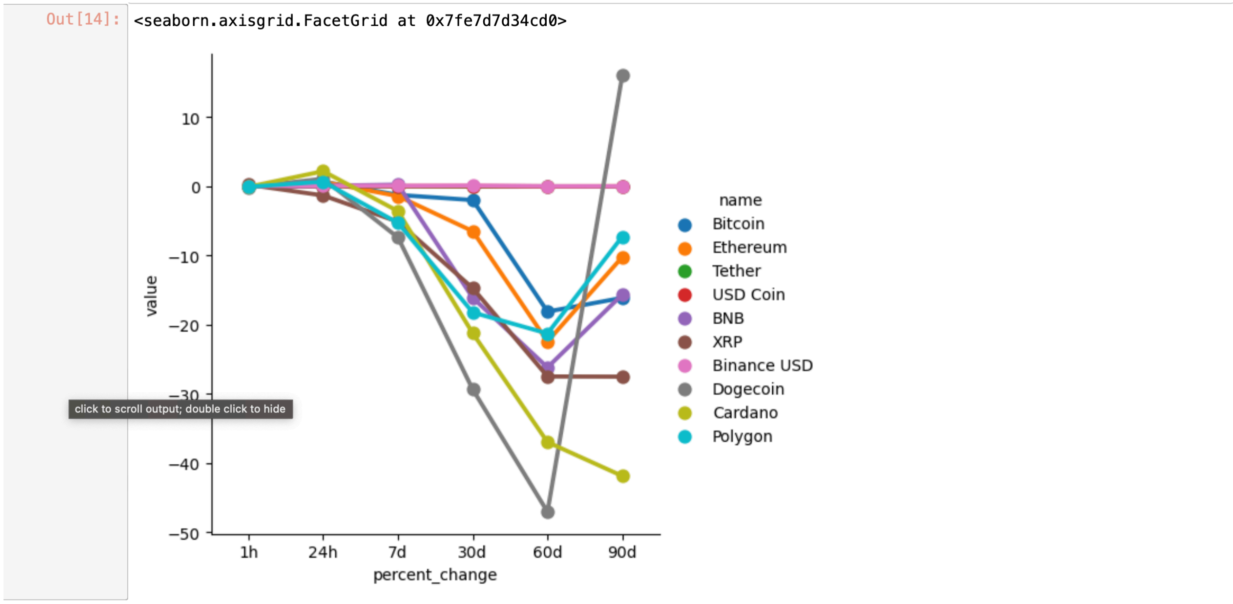
```
type(df4)
# df4 is series

# change df4 type back to data frame
df5 = df4.to_frame(name='value')
df5
```

Out[12]:

		value
name		
Bitcoin	quote.USD.percent_change_1h	-0.05788
	quote.USD.percent_change_24h	0.63699
	quote.USD.percent_change_7d	-1.27116
	quote.USD.percent_change_30d	-2.01917
	quote.USD.percent_change_60d	-18.14344
	quote.USD.percent_change_90d	-16.15577
Ethereum	quote.USD.percent_change_1h	-0.09946
	quote.USD.percent_change_24h	0.62109
	quote.USD.percent_change_7d	-1.44728
	quote.USD.percent_change_30d	-6.51622
	quote.USD.percent_change_60d	-22.46788
	quote.USD.percent_change_90d	-10.25439
Tether	quote.USD.percent_change_1h	-0.00525
	quote.USD.percent_change_24h	0.00521
	quote.USD.percent_change_7d	-0.02790
	quote.USD.percent_change_30d	-0.04073
	quote.USD.percent_change_60d	-0.03054
	quote.USD.percent_change_90d	-0.05221
USD Coin	quote.USD.percent_change_1h	-0.01769
	quote.USD.percent_change_24h	0.00261
	quote.USD.percent_change_7d	-0.00285
	quote.USD.percent_change_30d	0.00273
	quote.USD.percent_change_60d	-0.02733

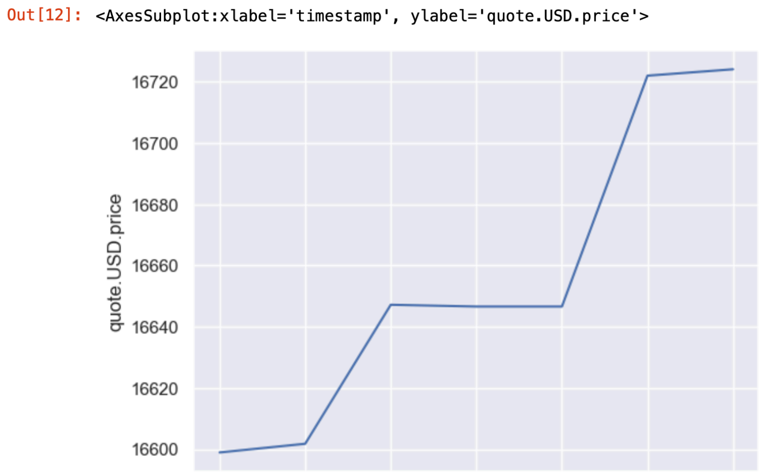
```
type(df5)

df5.count()
# there are 60 rows

index = pd.Index(range(60))
df6 = df5.reset_index()
df6
```

```
df7 = df6.rename(columns={'level_1':'percent_change'})
df7

df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_1h'],
['1h'])
df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_24h'],
['24h'])
df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_7d'],
['7d'])
df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_30d'],
['30d'])
df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_60d'],
['60d'])
df7['percent_change'] =
df7['percent_change'].replace(['quote.USD.percent_change_90d'],
['90d'])

import seaborn as sns
import matplotlib.pyplot as plt

sns.catplot(x='percent_change', y='value', hue='name', data=df7,
kind='point')
```



```
Out[14]: <seaborn.axisgrid.FacetGrid at 0x7fe7d7d34cd0>
```

```
# create a dataframe with the columns we want

df10 = df72[['name','quote.USD.price','timestamp']]
df10 = df10.query("name == 'Bitcoin'")
df10

sns.set_theme(style="darkgrid")

sns.lineplot(x='timestamp', y='quote.USD.price', data = df10)
```



```
Out[12]: <AxesSubplot:xlabel='timestamp', ylabel='quote.USD.price'>
```