

[Contest](#)[Community](#)[Research](#)[Algorithms](#)[Help](#)

Code your algorithm. Make money with it.

Quantopian provides capital, technology, data, and community. Everything you need to be successful.

Start Coding

Quantopian

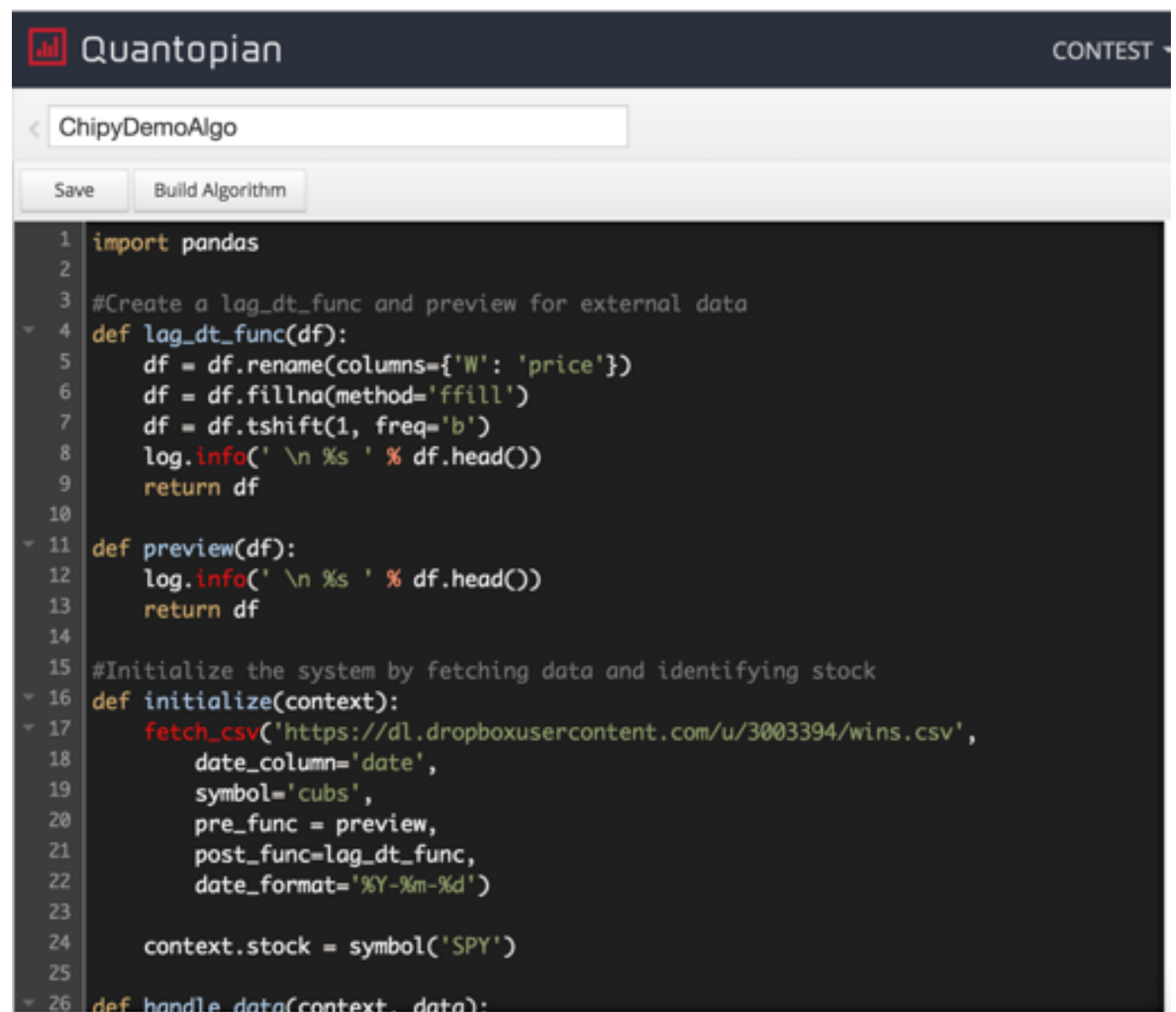
Chipy - September 2015

Setup

- Github: davidkunio/chipy-quantopian
- iPython Notebook
- Quantopian IDE/Backtest: Research backtests
- Slack: <https://chipy-slack-invites.herokuapp.com/>

Agenda

- Quantopian Overview
- Quantopian IDE Example
- Backtest Examples



The screenshot displays the Quantopian IDE interface. At the top, the 'Quantopian' logo is on the left, and a 'CONTEST' dropdown menu is on the right. Below the header, a search bar contains the text 'ChippyDemoAlgo'. Underneath the search bar are two buttons: 'Save' and 'Build Algorithm'. The main area of the IDE is a dark-themed code editor showing a Python script. The script is numbered from 1 to 26. It starts with 'import pandas'. Then, there is a comment '#Create a lag_dt_func and preview for external data'. This is followed by a function definition 'def lag_dt_func(df):' which contains several lines of code: 'df = df.rename(columns={'W': 'price'})', 'df = df.fillna(method='ffill')', 'df = df.tshift(1, freq='b')', 'log.info('\n %s ' % df.head())', and 'return df'. Next is another function definition 'def preview(df):' with 'log.info('\n %s ' % df.head())' and 'return df'. Then, a comment '#Initialize the system by fetching data and identifying stock' is followed by a function definition 'def initialize(context):'. Inside this function, there is a call to 'fetch_csv' with a URL, and several parameters: 'date_column='date'', 'symbol='cubs'', 'pre_func = preview', 'post_func=lag_dt_func', and 'date_format='%Y-%m-%d''. After the 'initialize' function, there is a line 'context.stock = symbol('SPY')'. The script ends with a function definition 'def handle_data(context, data):'.

```
1 import pandas
2
3 #Create a lag_dt_func and preview for external data
4 def lag_dt_func(df):
5     df = df.rename(columns={'W': 'price'})
6     df = df.fillna(method='ffill')
7     df = df.tshift(1, freq='b')
8     log.info('\n %s ' % df.head())
9     return df
10
11 def preview(df):
12     log.info('\n %s ' % df.head())
13     return df
14
15 #Initialize the system by fetching data and identifying stock
16 def initialize(context):
17     fetch_csv('https://dl.dropboxusercontent.com/u/3003394/wins.csv',
18             date_column='date',
19             symbol='cubs',
20             pre_func = preview,
21             post_func=lag_dt_func,
22             date_format='%Y-%m-%d')
23
24     context.stock = symbol('SPY')
25
26 def handle_data(context, data):
```

Quantopian is...

- Platform for developing investment strategies
- Jupyter based IDE + Custom Backtest Module - no install necessary. High powered batteries included
- Built on open-source: Zipline
- Connected to a broker (IB) for live trading
- Surprisingly active forum

Quantopian isn't...

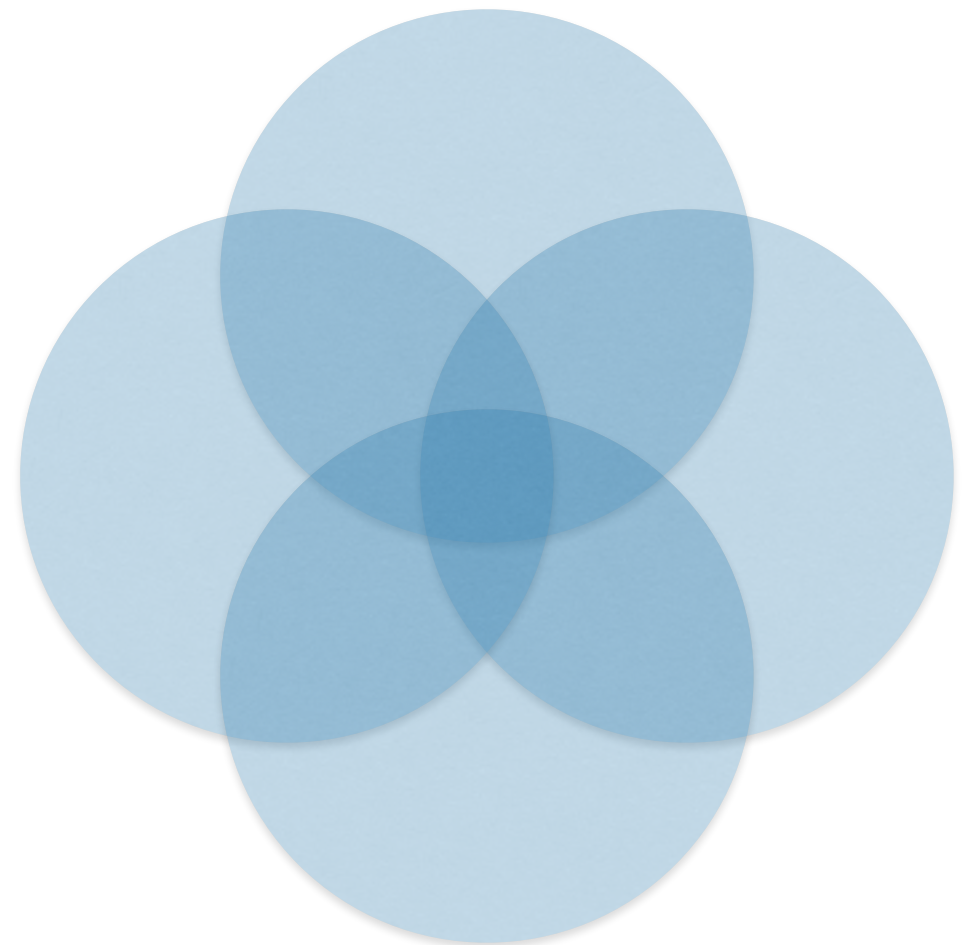
- Setup for High Frequency Trading (HFT) - per minute
- Fully open to the python development stack. There are integrated packages and a package white list system
- Multi asset class. US Equity/ETF Cash
- Limited Data: Namex per minute + Morningstar Fundamentals (Can access http hosted CSV files)

Quantopian is trying to...

- Crowd source a hedge fund
- Incentivize people to share their strategies.
- Contest - Run their money.

Quantopian needs a multi-dimensional person.

- Finance knowledge (or at least some intuition)
- Python skills
- Hacker skills
- Access to capital



Researching in Quantopian

Strength	Weakness
Access to Quantopian API	Limited package support
Integrated Dynamic Coding tools	No environment support
Strong user support	

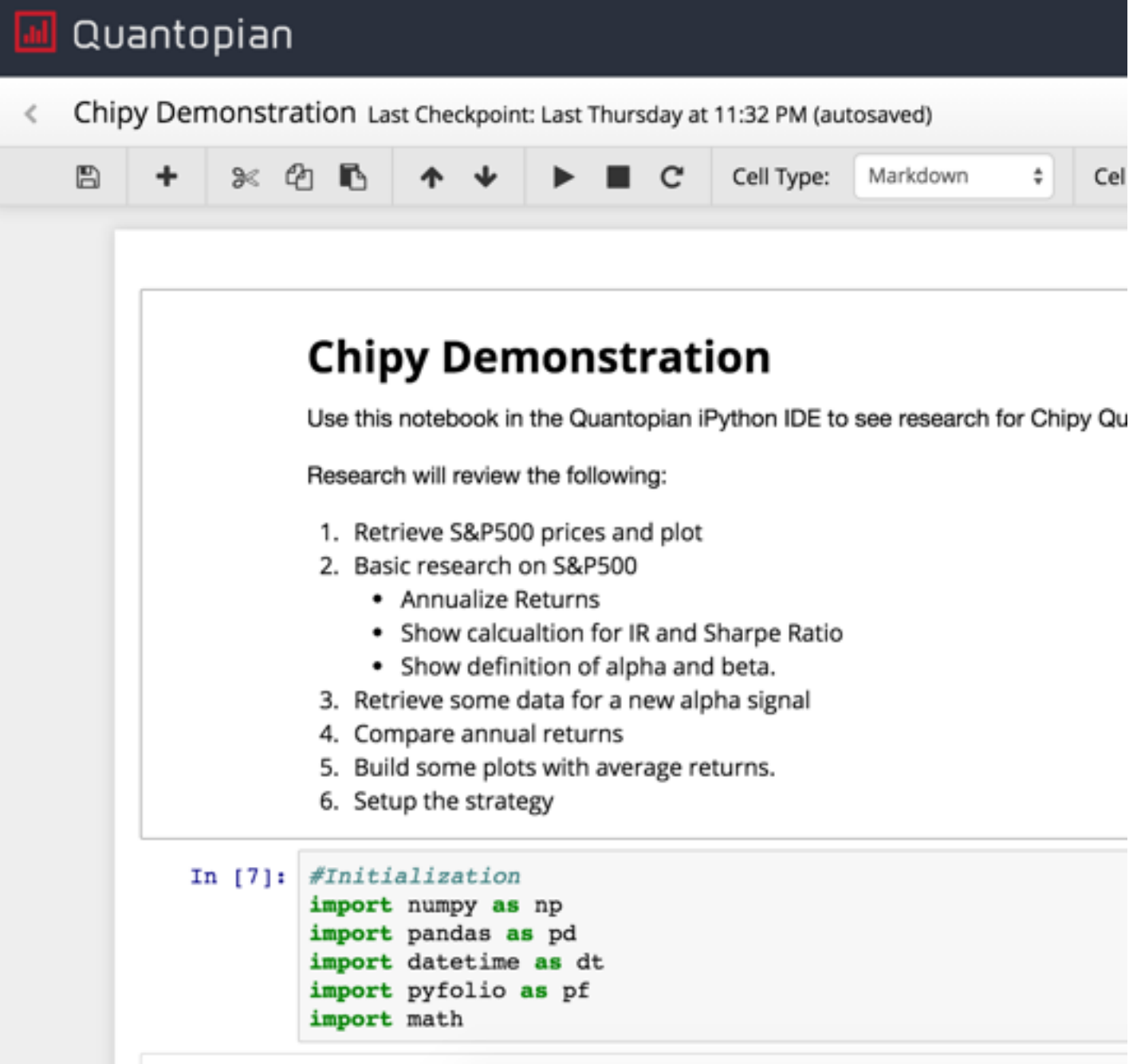


Quantopian

Lets take a tour

Research

- Explore in the tool
- External Data
- Researching a strategy
- Live Code Example!

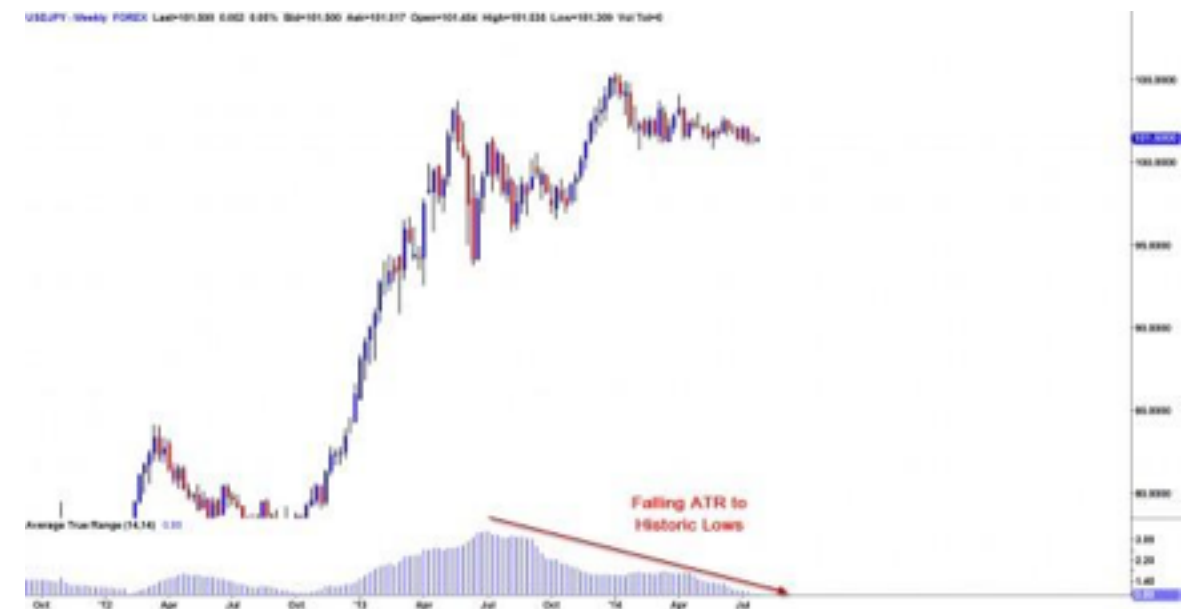


The screenshot displays the Quantopian iPython IDE interface. At the top, the header shows the Quantopian logo and the notebook title 'Chipy Demonstration' with a subtitle 'Last Checkpoint: Last Thursday at 11:32 PM (autosaved)'. Below the header is a toolbar with icons for file operations (save, new, open, close, copy, paste) and execution (run, stop, refresh). The 'Cell Type' dropdown is set to 'Markdown'. The main content area shows a notebook cell with the title 'Chipy Demonstration' and a description: 'Use this notebook in the Quantopian iPython IDE to see research for Chipy Qu'. Below this, it states 'Research will review the following:' followed by a numbered list of six tasks: 1. Retrieve S&P500 prices and plot, 2. Basic research on S&P500 (with sub-points: Annualize Returns, Show calculation for IR and Sharpe Ratio, Show definition of alpha and beta), 3. Retrieve some data for a new alpha signal, 4. Compare annual returns, 5. Build some plots with average returns, and 6. Setup the strategy. At the bottom, a code cell is shown with the prompt 'In [7]:' followed by Python code for initialization:

```
#Initialization
import numpy as np
import pandas as pd
import datetime as dt
import pyfolio as pf
import math
```

Robust Strategy

- Intuitive connection to market economic thesis
- Diversified: broad market, good spread of out performance in a variety of market conditions
- Reproducible: Backtest, Out of sample testing, live monitoring
- Risk Optimized: Factor Exposures,



Some Closing thoughts

- Use the forums: Quants are typically reserved, but the forums are quite active
- Clone Algorithms
- Fundamental Strategies are somewhat unexplored
- Developing in the IDE has perks, but more formalized research (iteration, optimization) can be better handled outside of the product.
- Backtests aren't "real". There are a lot of other considerations: transaction cost, margins, slippage, tax, etc.

Backtest Challenge



Places to start:

1. Improve the model
2. Alternative trading strategy (Going to cash isn't ideal)
3. Leverage - Leveraged ETF
4. Optimization

Other Resources

- [This Presentation on Github](#)
- [Quantopian Overview: Jess Stauth](#)
- [General Quantopian Resources](#)
- [Strategy Building: Dan Dunn](#)
- [Slack for the Group](#)