OEC 2018 Programming

University of Toronto A

Problem Description

Try to make the most money using a stock trading bot

Given: current and past prices of a set of stocks every minute

Unknown: future values of stocks

Problem Analysis

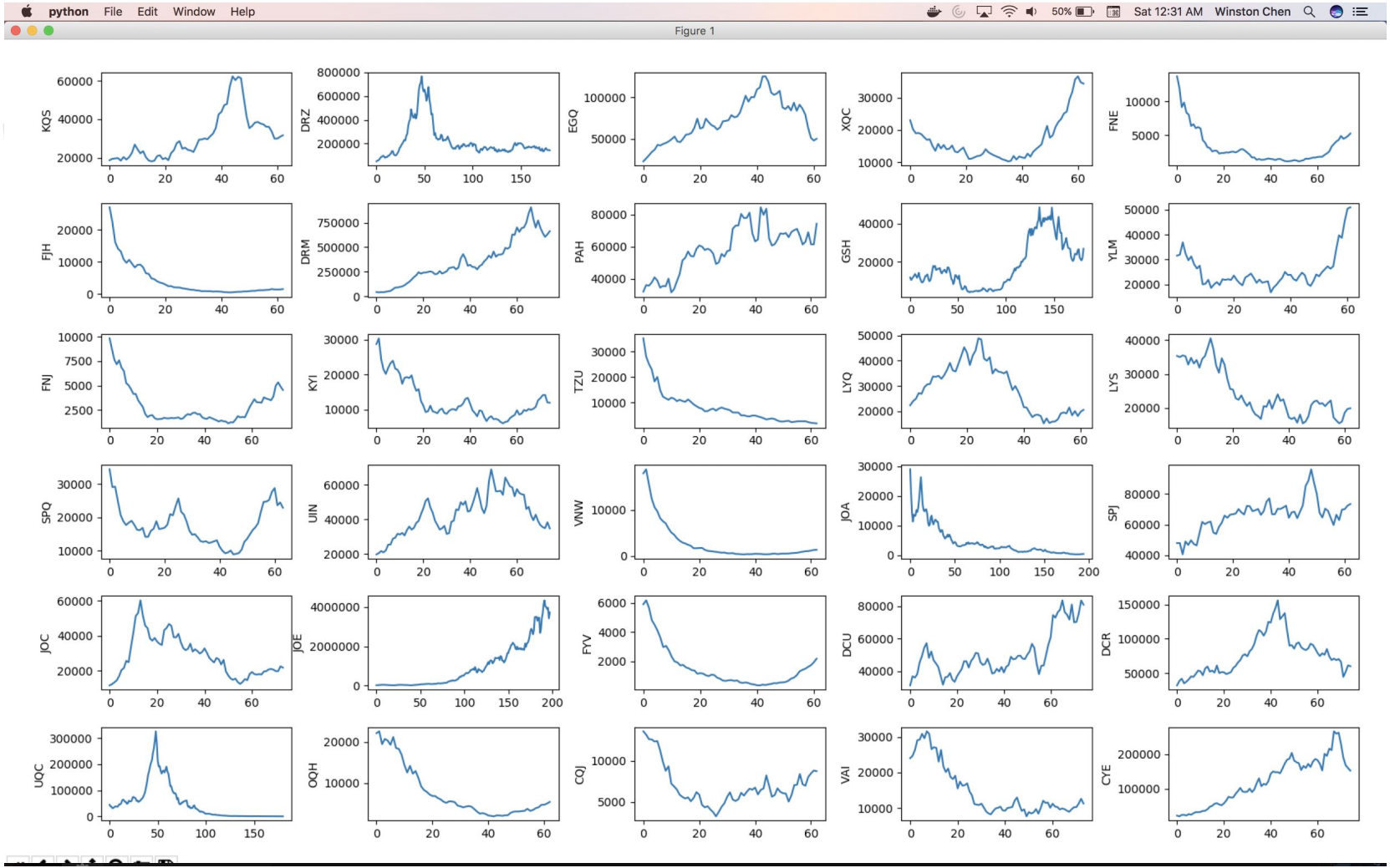
Information given is purely numerical

Market cap, company metrics (earnings, debt) are not given

Observed that the majority of stocks had either “smooth” or “unsmooth” price histories

Smooth stocks did not transition into unsmooth and vice versa

Idea- if volatile stocks appear to stay volatile, can take advantage of this via short-term transactions


python File Edit Window Help
50% Sat 12:31 AM Winston Chen
Figure 1
KQS
DRZ
EGQ
XQC
FNE
FJH
DRM
PAH
GSH
YLM
FNJ
KYI
TZU
LYQ
LYS
SPQ
UIN
VNW
JOA
SPJ
JOC
JOE
FYV
DCU
DCR
UQC
OQH
CQJ
VAI
CYE

Solution

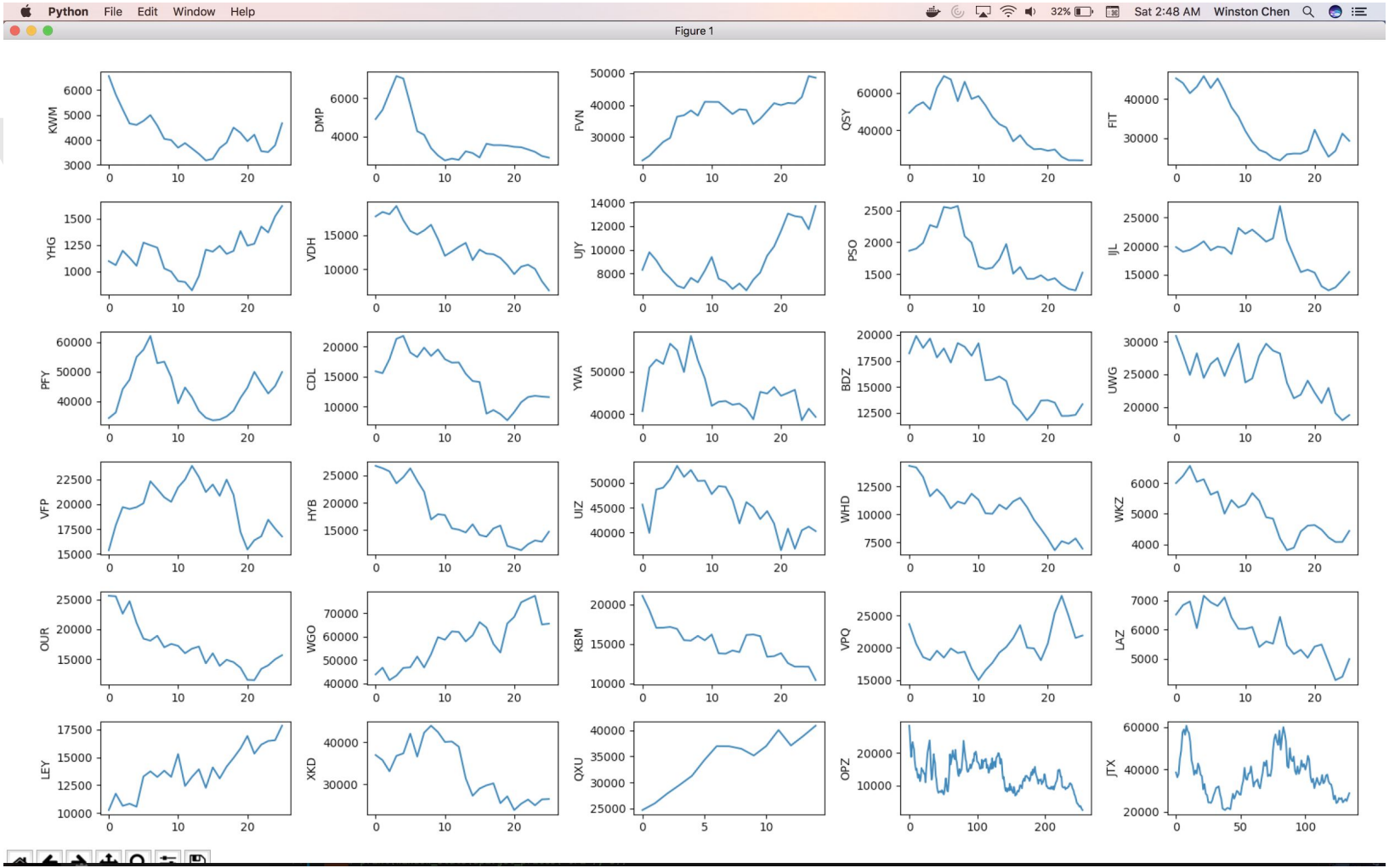
Solution takes advantage of volatile nature of certain stocks

Buy when the stock starts to rise

Sell when the stock starts to fall

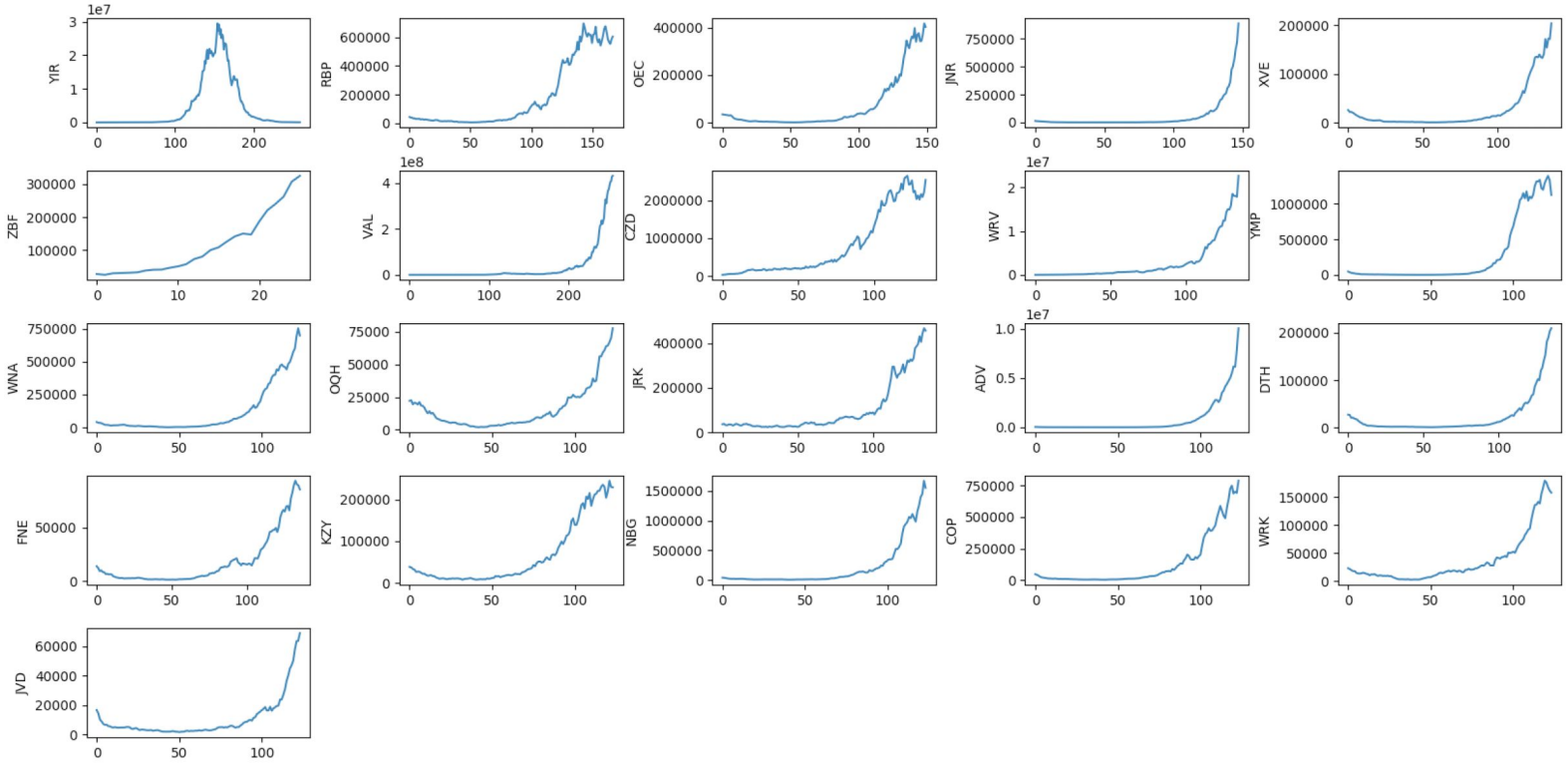
Avoid “smooth” stocks- no potential for short-term gains

Smoothness determined by autocorrelation coefficient


Python File Edit Window Help
32% Sat 2:48 AM Winston Chen
Figure 1
KWM
DMP
FVN
QSY
FIT
YHG
VDH
UJY
PSO
JIL
PFY
CDL
YWA
BDZ
UWG
VFP
HYB
UIZ
WHD
WKZ
OUR
WGO
KBM
VPQ
LAZ
LEY
XKD
QXU
OPZ
JTX

Python File Edit Window Help 32% Sat 2:50 AM Winston Chen

Figure 1

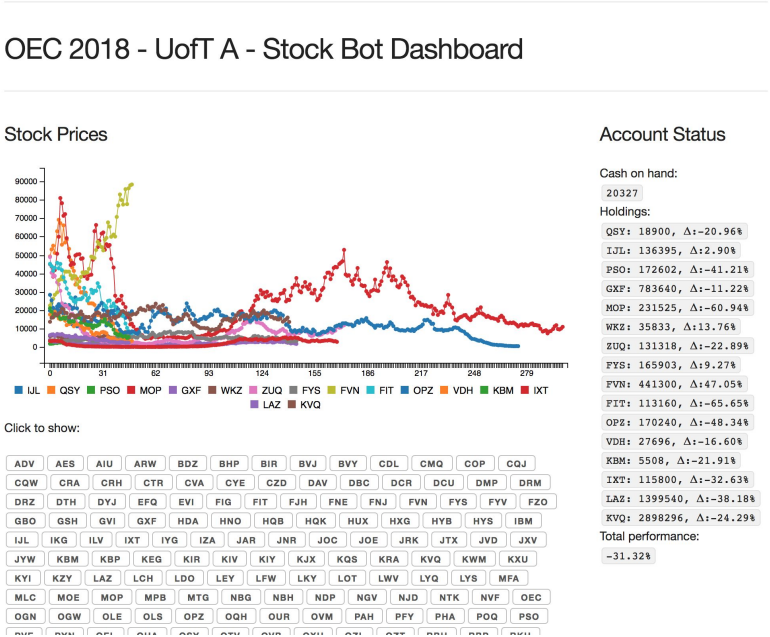


Prediction Algorithm

- Buy low sell high
- Smoothed first and second derivatives
- Also considered autocorrelation

User Dashboard

Real-time status of our account status as well as the historical prices of any stock



Testing (Simulation)

Testing was done on historical pricing data

Historical pricing resulted in a 2400x increase in net value to $240 mil

Simulation testing performed on a 173-step period