

Εργασία στη Μικροδομή των Χρηματοοικονομικών Αγορών

Προσομοίωση στρατηγικών Momentum και Contrarian

Εισαγωγή

Η εργασία αυτή αποτελεί μέρος της συνολικής εξέτασης του μαθήματος της Μικροδομής των Χρηματοοικονομικών Αγορών. Το θέμα της είναι η επιρροή του κόστους συναλλαγής (spread) στην εκτέλεση των επενδυτικών στρατηγικών και αποτελεί μία προσομοίωση εκτέλεσης στρατηγικών Momentum και Contrarian λαμβάνοντας υπόψη και μη τα κόστη συναλλαγών.

Η συγκεκριμένη εκτέλεση της εργασίας δεν αφορά μία προσομοίωση των προαναφερθέντων στρατηγικών σε ορισμένες εκ των προτέρων συνθήκες αλλά την προσομοίωση του επενδυτικού περιβάλλοντος (σε στρατηγικές Momentum και Contrarian). Με λίγα λόγια δημιουργήθηκε ένας αλγόριθμος ο οποίος, λαμβάνοντας ορισμένα ορίσματα, εκτιμά τα κέρδη ή τις ζημιές τις στρατηγικής με και χωρίς κόστη συναλλαγών και ένα σύνολο χρηματοοικονομικών δεικτών, μεταξύ άλλων. Έτσι δίνεται η δυνατότητα προσδιορισμού των συνθηκών σύμφωνα με τις οποίες θα εφαρμοστεί η στρατηγική.

Για την υλοποίηση της χρησιμοποιήθηκε η γλώσσα προγραμματισμού python (version 3.7.1), το πακέτο εργαλείων (library) backtrader μεταξύ άλλων (pandas, random, datetime, os, math, numpy και matplotlib) και το περιβάλλον αριθμητικής υπολογιστικής Matlab. Το πακέτο backtrader ειδικεύεται στη προσομοίωση στρατηγικών (backtesting) με ιστορικά στοιχεία και αποτέλεσε τον πυλώνα του κεντρικού κώδικα.

Σε αυτό το κείμενο θα περιγράψουμε τις ευρύτερες συνθήκες της προσομοίωσης, την μεθοδολογία που ακολουθείται και ορισμένα αποτελέσματα προσομοιώσεων. Προκειμένου να

επικεντρωθούμε στην μεθοδολογία θα αποφύγουμε να αναλύσουμε εκτεταμένα τα τεχνικά χαρακτηριστικά του κώδικα.

Μεθοδολογία και συνθήκες προσομοίωσης

Στο τμήμα αυτό του κειμένου θα αναλυθούν οι συνθήκες υπό τις οποίες συμβαίνει η προσομοίωση. Θα περιγραφτεί η μεθοδολογία κατά στάδια και θα οριστεί το ψηφιακό επενδυτικό περιβάλλον όπου λαμβάνουν χώρα οι στρατηγικές.

Συλλογή Δεδομένων

Η φύση και το πλήθος των δεδομένων καθίστανται παράγοντες ύψιστης σημασίας για την εξαγωγή ορθών συμπερασμάτων. Όπως αναφέρθηκε και στην εισαγωγή ο αλγόριθμος δεν δέχεται συγκεκριμένα ορίσματα και επομένως δεν βασίζει το αποτελέσματα σε δεδομένα συγκεκριμένων περιουσιακών στοιχείων. Δημιουργήθηκε λοιπόν μία βάση δεδομένων με ημερήσια ιστορικά στοιχεία από τα χρηματιστήρια Nasdaq, Nyse και Amex. Τα στοιχεία περιλαμβάνουν τις ημερομηνίες, τις τιμές ανοίγματος, την χαμηλότερη τιμή, την υψηλότερη τιμή, την τιμή κλεισίματος, τον όγκο συναλλαγών και την προσαρμοσμένη τιμή κλεισίματος για κάθε εταιρεία που διαπραγματευόταν στις παραπάνω αγορές από τις 18 Φεβρουαρίου 2012 μέχρι και τις 18 Φεβρουαρίου 2019.

Τα δεδομένα συλλέχθηκαν με την βοήθεια του περιβάλλοντος Matlab¹. Περιληπτικά ο αλγόριθμος κατεβάζει τα δεδομένα ανά χρηματιστηριακή αγορά από το Yahoo Finance, ελέγχει και αφαιρεί ημερήσια στοιχεία αξιόγραφων που δεν καλύπτουν ολόκληρο το χρονικό διάστημα (1759 παρατηρήσεις) και ύστερα αποθηκεύει τα στοιχεία σε μορφή csv. Τα τελικά δεδομένα αποτελούνταν από 208 αξιόγραφα από τη Amex αγορά, 1779 αξιόγραφα από τη Nasdaq αγορά και 1761 αξιόγραφα από τη Nyse αγορά.² Τέλος να σημειωθεί ότι τα σύμβολα (tickers) των περιουσιακών στοιχείων που κατεβάσαμε αποκτήθηκαν μέσω των παρακάτω υπερσυνδέσμων (links) αιτημάτων από την ιστοσελίδα του Nasdaq:

¹ Αρχείο κώδικα: dataDownload.m

² Να αναφερθούμε στο γεγονός ότι το API του Yahoo Finance είναι ξεπερασμένο και δεν υποστηρίζεται από τα τρέχοντα πακέτα συλλογής δεδομένων. Για αυτόν τον λόγο ακολούθησε ένας περαιτέρω έλεγχος για κενές (NaN Values) οποίος αποδείχθηκε αρνητικός.

Nasdaq Quotes: <http://www.nasdaq.com/screening/companies-by-name.aspx?letter=0&exchange=nasdaq&render=download>

Nyse Quotes: <http://www.nasdaq.com/screening/companies-by-name.aspx?letter=0&exchange=nyse&render=download>

Amex Quotes: <http://www.nasdaq.com/screening/companies-by-name.aspx?letter=0&exchange=amex&render=download>

Περίληπτική περιγραφή διαδικασίας backtesting (backtrader backtesting logic)

Η προσομοίωση βασίζεται στο πακέτο εργαλείων (library) backtrader. Οι κύριες λειτουργίες της πλατφόρμας αυτής είναι η κατασκευή μίας στρατηγικής και η εισαγωγή της στην μηχανή backtesting. Από αυτό το σημείο και μετά δύναται η προσθήκη δεικτών (backtrader.Indicators), παρατηρητών (backtrader.Observers), αναλυτών (backtrader.Analyzers) και άλλων χαρακτηριστικών τα οποία εμπλουτίζουν την στρατηγική και εξάγουν πληροφορίες για την επίδοσή της. Επιπροσθέτως, ένα από τις σημαντικές λειτουργίες είναι και οι δημιουργία σύνθετων εργαλείων, βασισμένων στα προαναφερθέντα, τα οποία ευθυγραμμίζουν την λειτουργία της πλατφόρμας με αυτήν που επιθυμεί ο χρήστης.

Η πλατφόρμα διαθέτει μια κεντρική κλάση (backtrader.Cerebro) στην οποία εισάγονται τα δεδομένα (Data Feeds), η στρατηγική (backtrader.Strategy), οι παρατηρητές (backtrader.Observers), οι αναλυτές (acktrader.Analyzers) και οι καταγραφείς (backtrader.Writers). Ύστερα το cerebro εκτελεί το backtesting με τα εισαχθέντα στοιχεία, επιστρέφει τα αποτελέσματα και δίνει πρόσβαση σε δυνατότητες δημιουργίας διαγραμμάτων.

Η λογική που ακολουθεί η διαδικασία backtesting είναι η εξής:

1. Εμφάνιση τυχών αποθηκευμένων ειδοποιήσεων προς τον χρήστη.
2. Αίτημα από τη ροή δεδομένων (Data Feeds) να παραδοθεί το επόμενο tick/bar.
3. Εμφάνιση ειδοποιήσεων από τον broker (πραγματικό ή αντικείμενο κλάσης backtrader.broker) για τις εντολές, τις συναλλαγές, τα ταμειακά διαθέσιμα και την αξία του λογαριασμού.
4. Αποστολή προς τον broker των στοιβαγμένων εντολών που προκύπτουν από την επεξεργασία των νέων ροών δεδομένων.

5. Καλείται η `next` διαδικασία (method) η οποία αξιολογεί τα δεδομένα και εκτελεί τις ανάλογες πράξεις. Επιπλέον, μέσω εσωτερικών διαδικασιών εκτελούνται οι λειτουργίες των Indicators, Observers και Analyzers.

6. Τέλος, οι Writers, εφόσον υπάρχουν, καταγράφουν τα αποτελέσματα.

Προσομοίωση

Στην προηγούμενη ενότητα περιγράψαμε την λογική πάνω στην οποία βασίζεται η εν χρήση πλατφόρμα. Θα στηριχθούμε στην περιγραφή αυτή για να εξηγήσουμε τον εσωτερικό μηχανισμό της προσομοίωσης μας. Υπενθυμίζουμε ότι δεν θα περιγραφτούν εκτεταμένα χαρακτηριστικά του κώδικα τεχνικής φύσεως.

- Κλάση `momentum`: Είναι μία κλάση στρατηγικής, δηλαδή έχει κληρονομήσει όλα τα χαρακτηριστικά που απορρέουν από την parent class `backtrader.Strategy` και περιγράφει την `momentum` στρατηγική. Κατά το κάλεσμά της μπορούν να προσδιοριστούν οι παράμετροι:
 - `numberOfShares`: Ο αριθμός των περιουσιακών στοιχείων που λαμβάνεται στην `long` και `short` θέση (προεπιλογή: 10).
 - `weight`: Τα ποσοστά σε απόλυτες τιμές σύμφωνα με τα οποία θα κατανεμηθεί το επενδύόμενο κεφάλαιο (προεπιλογή: 'equal'). Υπάρχουν οι έξις τρεις πιθανές επιλογές:
 - 'equal': Όλα τα βάρη των περιουσιακών στοιχείων είναι μεταξύ τους ίσα.
 - 'returnToSumOfReturns': Ο λόγος $w_i = r_i / (r_1 + r_2 + \dots + r_n)$, όπου r_i στην συγκεκριμένη περίπτωση είναι η λογαριθμική απόδοση του εν λόγω περιουσιακού στοιχείου και n ο αριθμός των προς αγορά (ή πώληση) περιουσιακών στοιχείων.
 - 'divByStd': Ο λόγος $w_i = (1/\sigma_i) / ((1/\sigma_1) + (1/\sigma_2) + \dots + (1/\sigma_n))$, όπου σ_i η τυπική απόκλιση του εν λόγω περιουσιακού στοιχείου και n ο αριθμός των προς αγορά (ή πώληση) περιουσιακών στοιχείων.
 - `investValue`: Το μέγεθος της `short` και της `long` θέσης σε νομισματικές μονάδες.
 - `dirPath`: Το μονοπάτι (path) για το directory όπου θα σωθούν τα αποτελέσματα (προεπιλογή: το τρέχον directory (current working directory)).

Μέσα στο αντικείμενο momentum υπάρχουν οι ακόλουθες διαδικασίες (methods):

- `momentum.log(self, txt, dt=None)`: Δέχεται ως όρισμα μία αλφαριθμητική ακολουθία την οποία η συνάρτηση εμφανίζει στον χρήστη. Μέσω αυτής προβάλλονται όλα τα μηνύματα προς τον χρήστη.
- `momentum.__init__(self)`: Αποτελεί συνάρτηση εκκίνησης. Δημιουργεί τις περισσότερες μεταβλητές και δείκτες για την λειτουργία της στρατηγικής.
- `momentum.start(self)`: Παρόμοια λειτουργία με την `momentum.__init__`. Δημιουργεί τις υπόλοιπες μεταβλητές που χρειάζονται για την εκτέλεση της στρατηγικής και δημιουργεί τα αρχεία που θα καταγραφτούν το spread και οι αλφαριθμητικές αλληλουχίες που προβάλλονται στον χρήστη. Καλείται από το cerebro στην αρχή της προσομοίωσης.
- `momentum.stop(self)`: Σηματοδοτεί το τέλος του backtesting. Στην συγκεκριμένη περίπτωση απλά τερματίζει την καταγραφή στα txt αρχεία.
- `momentum.notify_order(self, order)`: Μέσω αυτής της συνάρτησης διοχετεύει το cerebro στον χρήστη τις ειδοποιήσεις για οποιαδήποτε αλλαγή στην κατάσταση των εντολών προς τον broker.
- `momentum.notify_trade(self, trade)`: Παρόμοια λειτουργία με την `momentum.notify_trade` με την διαφορά ότι ειδοποιεί το χρήστη για την εξέλιξη οποιουδήποτε trade. Επιπλέον καταγράφει το κέρδος ή ζημία της περιόδου.
- `momentum.next(self)`: Στην ουσία αποτελεί τις οδηγίες που περιγράφουν την στρατηγική μας. Αρχικά θέτει το κόστος συναλλαγής για κάθε περιουσιακό στοιχείο. Έπειτα στέλνει εντολές κλεισίματος για όλες τις ανοιχτές θέσεις και κατατάσσει τα περιουσιακά στοιχεία ανάλογα με τις λογαριθμικές αποδόσεις σε αυτά με τις υψηλότερες και αυτά με τις χαμηλότερες. Τέλος στέλνει εντολές αγοράς των στοιχείων με τις μεγαλύτερες αποδόσεις και εντολές ανοιχτής πώλησης στις μετοχές με τις μικρότερες αποδόσεις. Οι συναλλαγές που πραγματοποιούνται είναι round trip trades και η εξίσωση που υπολογίζει την ποσότητα του κάθε αξιογράφου είναι $size = wi * (investValue) * (investValue) / pi$, όπου pi η τιμή κλεισίματος του περιουσιακού στοιχείου.

- Κλάση `contrarian(momentum)`: Αφορά τη `contrarian` στρατηγική. Επίσης είναι “απόγονος” της `momentum`, δηλαδή έχει υιοθετήσει όλα τα χαρακτηριστικά της `momentum` στρατηγικής.

- `contrarian.next(self)`: Υπερκαλύπτει την διαδικασία `momentum.next(self)` της `momentum` κλάσης. Στην ουσία είναι ο ίδιος κώδικας με την διαφορά ότι δίνονται εντολές αγοράς στον `broker` για τα περιουσιακά στοιχεία με τις μικρότερες λογαριθμικές αποδόσεις και εντολές ανοιχτής πώλησης για τις εντολές με τις μεγαλύτερες λογαριθμικές αποδόσεις.

- Η συνάρτηση `runstrat(strategy, cash, weight, numberOfShares, investValue, exchange, sample, timeframe, numData=50, plot=True, standardStats=False, coc=True, int2pnl=False)` εκτελεί την στρατηγική μας σύμφωνα με τα ορίσματα που δίνονται από τον χρήστη. Ακολουθεί ανάλυση των ορισμάτων:

- `cash`: Τα μετρητά που θα έχει ο λογαριασμός του χρήστη στον `broker` κατά την εκκίνηση του `backtesting`. Υποστηρίζεται η χρήση μεγάλου αριθμού διότι σε περίπτωση ζημιών που ξεπερνούν τα μετρητά στον λογαριασμό ο `broker` θα κλείσει αυτόματα την θέση μας. Αυτό δεν είναι επιθυμητό καθώς σκοπός μας είναι η καταγραφή των αποτελεσμάτων. Σε σενάριο μηδενικού αποτελέσματος (μηδενικών κερδών ή ζημιών) ο λογαριασμός των μετρητών θα παραμείνει ο ίδιος καθώς οι συναλλαγές είναι της μορφής `round trip trades`. Τα κέρδη ή ζημιές καταγράφονται σε ξεχωριστή μεταβλητή όπου και τοκίζονται με το `OverNight Libor`.

- `weight`: Ο τρόπος υπολογισμού των βαρών όπως τον ορίσαμε παραπάνω.

- `numberOfShares`: Ο αριθμός των περιουσιακών στοιχείων που λαμβάνεται για τη `long` και `short` θέση.

- `investValue`: Το μέγεθος των θέσεων στην `long` και `short` κατεύθυνση.

- `exchange`: Το χρηματιστήριο του οποίου τα περιουσιακά στοιχεία θα ενσωματωθούν στο `cerebro` για την εκτέλεση του `backtesting`. Εν δυνάμει τιμές για το συγκεκριμένο όρισμα:

- ‘NASDAQ’ για το χρηματιστήριο Nasdaq.
 - ‘AMEX’ για το χρηματιστήριο Amex.
 - ‘NYSE’ για το χρηματιστήριο Nyse.

- `sample`: Το όρισμα αυτό παίρνει τις τιμές `True` ή `False`. Εάν είναι `True` επιλέγεται ένας ορισμένος από τον χρήστη αριθμός περιουσιακών στοιχείων. Τα περιουσιακά στοιχεία επιλέγονται τυχαία μέσω του πακέτου εργαλείων (library) `random`. Εάν πάρει την τιμή `False` η στρατηγική εφαρμόζεται στο σύνολο των περιουσιακών στοιχείων της επιλεχθείσας χρηματιστηριακής αγοράς.
- `timeframe`: Ο χρονικός ορίζοντας που μεσολαβεί ανάμεσα από κάθε rebalancing. Το όρισμα πρέπει να είναι τύπου `backtrader.TimeFrame`. Εν δυνάμει τιμές:
 - `backtrader.TimeFrame.Months` για μηνιαία διαστήματα.
 - `backtrader.TimeFrame.Weeks` για εβδομαδιαία διαστήματα.
 - `backtrader.TimeFrame.Days` για ημερήσια διαστήματα.
- `numData`: Ο αριθμός των ρών δεδομένων των περιουσιακών στοιχείων που θα εισαχθεί στον `cerebro` (Προεπιλογή: 50)
- `plot`: Εάν εισαχθεί η τιμή `True` ενεργοποιούνται οι δυνατότητες δημιουργίας διαγραμμάτων του `cerebro` (Προεπιλογή: `True`).
- `standardStats`: Εάν εισαχθεί η τιμή `True` το `cerebro` θα δημιουργήσει διαγράμματα που θα περιγράφουν την συμπεριφορά των μεταβλητών των μετρητών, της αξίας του λογαριασμού και των τιμών των περιουσιακών στοιχείων (Προεπιλογή: `False`).
- `coc`: Cheat on Close. Αφορά την συμπεριφορά του broker κατά την εκτέλεση των εντολών μας. Εάν το όρισμα έχει τιμή `True` ο broker θα εκτελέσει τις εντολές μας στις τιμές κλεισίματος των `tick/bars` που στάλθηκε η εντολή στον broker (Προεπιλογή: `True`). Αν και μη ρεαλιστική συνθήκη, μας βοηθάει να προσεγγίσουμε καλύτερα την τακτική του `round trip trade` διότι η τιμή αγοράς (ή πώλησης) είναι η ίδια με αυτήν που αξιοποιήθηκε για την εκτίμηση του μεγέθους της θέσης³. Εάν στο όρισμα δινόταν η τιμή `False` η εντολή θα εκτελούνταν στην τιμή ανοίγματος του επόμενου `tick/bar` η οποία ενδέχεται να είναι και διαφορετική από την τιμή κλεισίματος που χρησιμοποιήθηκε στον υπολογισμό.
- `int2pnl`: Αφορά την εφαρμογή προμήθειας από τον broker ως ποσοστό στα κέρδη ή ζημιές από τις συναλλαγές (Προεπιλογή: `True`).

³ βλέπε εξίσωση στην περιγραφή του διαδικασίας `momentum.next()`

Ο σκοπός αυτής της συνάρτησης είναι μέσω της εισαγωγής των απαραίτητων δεδομένων (Data Feeds), στοιχείων (Analyzers, Observers) και χαρακτηριστικών να εκτελεστεί το backtesting. Η runstrat επιστρέφει δύο μεταβλητές με τα αποτελέσματα και την ανάλυση των αποτελεσμάτων αντίστοιχα καθώς και δύο διαγράμματα που περιγράφουν την απόδοση της επένδυσης και την τιμή drawdown του χαρτοφυλακίου ανά περίοδο. Τέλος δημιουργείται ένας φάκελος στο current working directory με όνομα που απορρέει από τα ορίσματα της προσομοίωσης και περιεχόμενο αρχεία txt που περιέχουν τις τιμές των δεικτών και δύο διαγράμματα που αναλύουν την συμπεριφορά των κερδών ή ζημιών και του λογαριασμού που κατατίθενται αυτά.

Κόστος Συναλλαγών

Καθώς είναι το αντικείμενο της μελέτης, δόθηκε έμφαση στην εκτίμηση του κόστους συναλλαγών έτσι ώστε να ενσωματωθεί ορθά στο υπόλοιπο σύστημα. Σε θεωρητικό πλαίσιο χρησιμοποιήθηκε το μοντέλο του Roll. Πιο συγκεκριμένα, για να υπολογίσουμε το κόστος συναλλαγών αρχικά θα εκτιμήσουμε την αυτοδιακύμανση πρώτης τάξης των λογαριθμικών αποδόσεων των περιουσιακών στοιχείων. Ύστερα το κόστος συναλλαγών δίνεται από τους παρακάτω τύπους:

$$c = \sqrt{-\gamma_1}$$

$$\gamma_1 = cov[\Delta \log(P_t) \Delta \log(P_{t-1})]$$

όπου το c είναι το one-half effective spread ως ποσοστό της τιμής. Σε περίπτωση που το γ δεν οριστεί, το one-half effective spread δεν ορίζεται και του αποδίδεται η τιμή 0.

Δημιουργήθηκε λοιπόν ένας δείκτης ο οποίος αντλεί τιμές από τις λογαριθμικές αποδόσεις των περιουσιακών στοιχείων και ένας δείκτης ο οποίος αξιοποιεί τις τιμές του προηγούμενου για τον υπολογισμό του c για κάθε περιουσιακό στοιχείο⁴. Σε αυτό το σημείο αρμόζει να αναφέρουμε ότι το σύστημα λαμβάνει την περίοδο υπολογισμού των τιμών ως “περίοδο ωρίμανσης”. Με άλλα λόγια, εωσότου να είναι δυνατή η εκτίμηση των τιμών των δεικτών δεν καλείται η ιδιότητα next και άρα δεν ξεκινάει η εφαρμογή της στρατηγικής.

⁴ βλέπε αρχείο κώδικες, MyIndicators.py.

Αποτελέσματα

Αφού αναλύθηκε η μεθοδολογία και οι συνθήκες τις προσομοίωσης, θα παρουσιαστούν τα αποτελέσματα από την διεξαγωγή δύο προσομοιώσεων. Ως ορίσματα επιλέχθηκαν τα ακόλουθα:

- strategy: momentum και contrarian
- cash: 1000000
- weights: divByStd
- numberOfShares: 10
- investValue: 1000
- exchange: NYSE
- sample:100
- timeframe: backtrader.TimeFrame.Months

Το σενάρια αυτά επιλέχθηκαν διότι αναπαριστούν ικανοποιητικά τις συνθήκες της αγοράς. Επίσης σε αυτά είναι εμφανή η επίδραση του κόστους συναλλαγών και η αντίθετη φύση των δύο αυτών στρατηγικών καθώς η μία παράγει κέρδη και η άλλη ζημιές.

Οι εντολές που πρέπει να εκτελεστούν για την πραγματοποίηση των προσομοιώσεων είναι⁵:

- Momentum:

```
results, analysis = runstrat(momentum, 1000000, 'divByStd', 10, 1000, 'NYSE',  
False, bt.TimeFrame.Months, numData=50, plot=True, standardStats=False, coc=True,  
int2pnl=False)
```

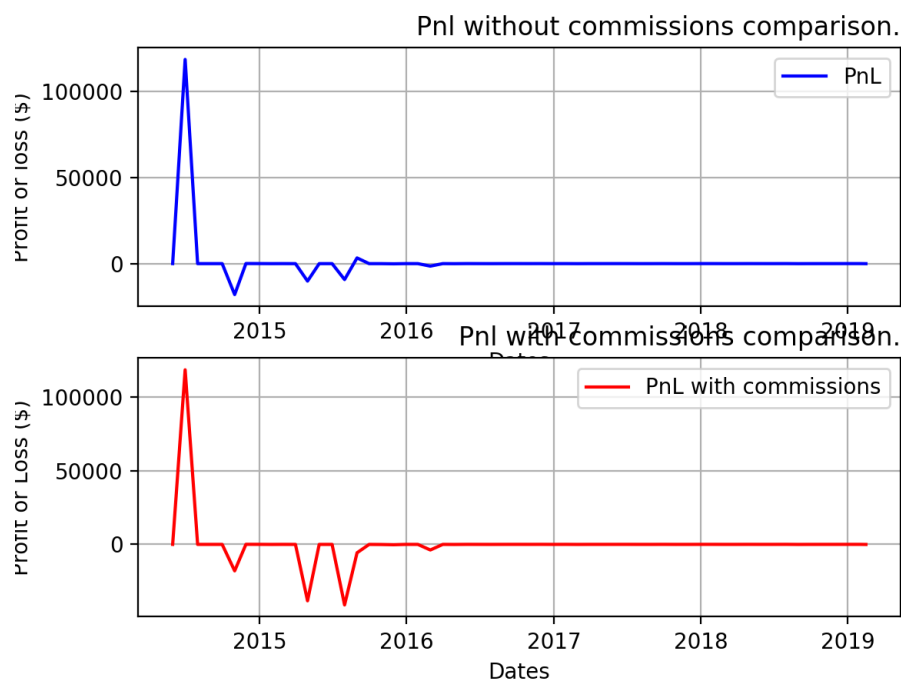
- Contrarian:

```
results, analysis = runstrat(contrarian, 1000000, 'divByStd', 10, 1000, 'NYSE',  
False, bt.TimeFrame.Months, numData=50, plot=True, standardStats=False, coc=True,  
int2pnl=False)
```

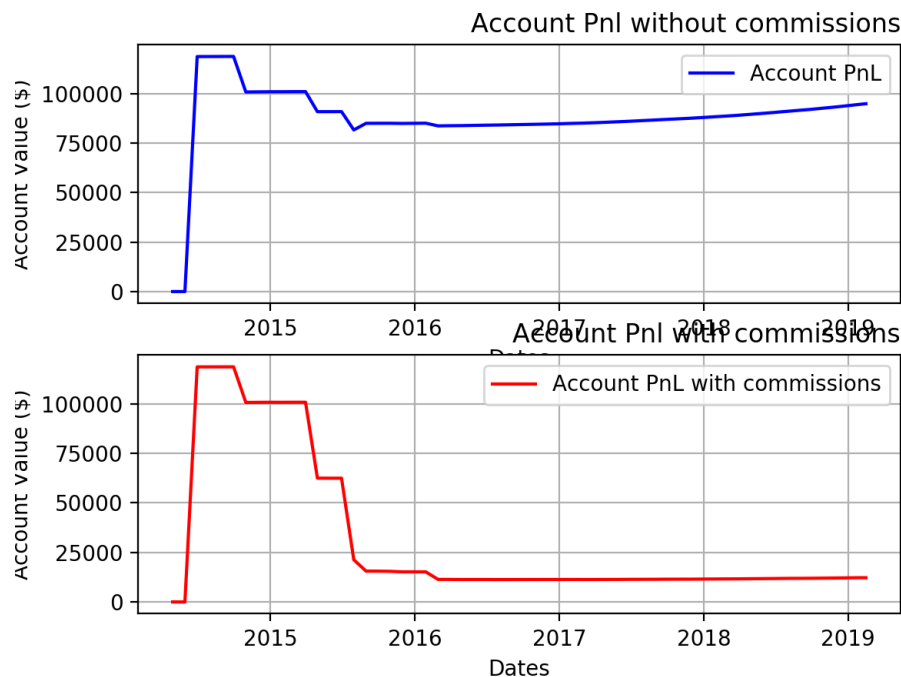
Momentum

⁵ εφόσον έχει προηγηθεί η εντολή: `from Simulation import *`

Περιγραφή: Αναλύοντας τα ιστορικά στοιχεία όλων των -διαθέσιμων⁶- περιουσιακών στοιχείων της χρηματιστηριακής αγοράς Nyse, ξεχωρίζουμε τα δέκα αξιόγραφα με την μεγαλύτερη λογαριθμική απόδοση και τα δέκα με την μικρότερη. Ύστερα, για διάστημα ενός μήνα, πουλάμε ανοιχτά την κατηγορία με την μικρότερη απόδοση και αγοράζουμε την κατηγορία με την μεγαλύτερη. Το βάρος του κάθε περιουσιακού στοιχείου στο σύνολο του χαρτοφυλακίου ανά κατεύθυνση (long και short) θα εξαρτηθεί από τον κίνδυνο αυτού συγκριτικά με τον κίνδυνο των υπολοίπων. Τέλος επενδύονται 1000 \$ στις long θέσεις, τα οποία χορηγούνται από την ανοιχτή πώληση της αντίστοιχης short. Η διαδικασία επαναλαμβάνεται σε μηνιαία βάση για 5 έτη. Να θυμίσουμε ότι ο λογαριασμός κερδών/ζημιών τοκίζεται με το επιτόκιο Overnight Libor σε ημερήσια βάση.



⁶ ύπαρξη ιστορικών στοιχείων για διάστημα 7 ετών



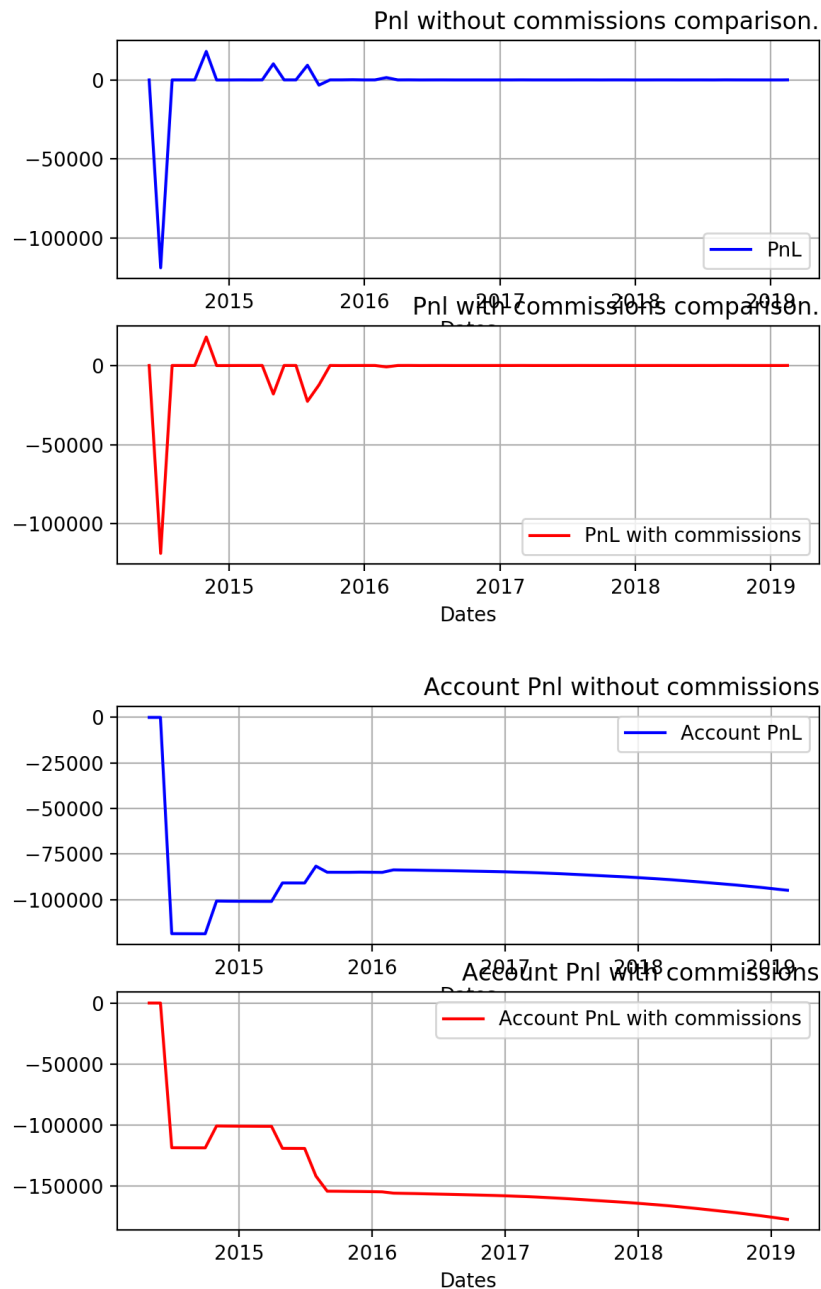
Εάν ένας επενδυτής ακολουθούσε αυτήν την στρατηγική θα απολάμβανε σημαντικά κέρδη το πρώτο χρόνο της επένδυσης. Στον υπόλοιπο χρονικό ορίζοντα, τα κέρδη και ζημιές που θα είχε το χαρτοφυλάκιο κάθε μήνα προσεγγίζουν το μηδέν. Ωστόσο, ο αναγνώστης θα παρατηρήσει μια σημαντική εκροή κεφαλαίων από τον λογαριασμό του μετά το πρώτο τετράμηνο του 2015. Αυτό το συμβαίνει διότι δεν παράγονται κέρδη με αποτέλεσμα την μη κάλυψη των κοστών συναλλαγών από την αναδιάρθρωση. Προς το τέλος της στρατηγικής, εμφανίζεται μια μικρή άνοδος στην αξία του λογαριασμού η οποία “χορηγεί” τα κόστη συναλλαγών της αναδιάρθρωσης και κράτα στάσιμο το υπόλοιπο του λογαριασμού. Αν και η αύξηση των κερδών αποτελεί εν μέρει εξήγηση του φαινομένου, η αντίσταση του λογαριασμού στην πτώση από το 2018 και μετά οφείλεται στην ραγδαία αύξηση του επιτοκίου Overnight Libor με το οποίο και ανατοκίζουμε τον λογαριασμό υπόλοιπου⁷. Συμπερασματικά, αν και η στρατηγική εμφάνισε κέρδη, τα κόστη συναλλαγών οδήγησαν σε σημαντική εκροή κεφαλαίων, η οποία σταθεροποιήθηκε λόγω συγκυριακών παραγόντων.

⁷ Μεγάλη πτώση των χρηματιστηριακών αγορών κατά τον τελευταίο χρόνο. Η fed ανακοινώνει απρογραμμάτιστη αύξηση των επιτοκίων.

Contrarian

Περιγραφή: Η στρατηγική ακολουθεί ταυτόσημη μεθοδολογία με εξαίρεση μία ειδοποιός διαφορά: πουλιέται ανοιχτά το long χαρτοφυλάκιο και αγοράζεται το short.

Ακολουθούν τα διαγράμματα κερδών/ζημιών και μεταβολής της αξίας του λογαριασμού ανά περίοδο.



Όπως ήταν αναμενόμενο, τα αποτελέσματα είναι αντίθετα σε σχέση με την momentum. Η αρχική υπεραπόδοση της προηγούμενης στρατηγικής μεταφράζεται σε ζημιές στην contrarian, γεγονός που είναι λογικό εάν αναλογιστούμε ότι λαμβάνονται οι αντίθετες θέσεις. Όμως τα κόστη συναλλαγών έχουν όμοιες επιδράσεις στην τιμή του λογαριασμού: επιβαρύνουν σημαντικά τον επενδυτή οδηγώντας σε εκροές κεφαλαίων της τάξεως των 87% επιπλέον στην περίπτωση που συνυπολογίζονται τα κόστη συναλλαγών στα τελικά υπόλοιπα των λογαριασμών, παρόλο που οι ζημιές ανά περίοδο είναι πολύ μικρές.

Round Trip Trades

Κοινό χαρακτηριστικό των παραπάνω συναλλαγών είναι ότι εκτελούνται με την μορφή round trip trades. Ειδικότερα, οι long θέσεις του χαρτοφυλακίου χρηματοδοτούνται από τις short εξαλείφοντας την ανάγκη αρχικού κεφαλαίου. Σε αυτό το σημείο είναι συνετό να υπενθυμίσουμε την συνθήκη Cheat on Close που αναλάβαμε προκειμένου να προσεγγίσουμε αυτήν την υπόθεση. Ο πίνακας που παρατίθεται στην επόμενη σελίδα περιέχει τις τιμές του δείκτη Gross Leverage για τις πρώτες 30 περιόδους που λαμβάνουν χώρα οι στρατηγικές. Από τις τιμές του δείκτη επιβεβαιώνουμε ότι οι στρατηγικές εκτελούνται με την μορφή round trip trades. Οι τιμές προσεγγίζουν το μηδέν όχι μόνο στις αναλύσεις που προκύπτουν από τις συγκεκριμένες στρατηγικές αλλά και από όλες τις στρατηγικές που πραγματοποιούνται μέσω του συστήματος.

	Momentum	Contrarian
Date	GrossLeverage	GrossLeverage
5/1/14	5.22E-06	-5.22E-06
5/31/14	1.06E-01	-0.134620219
7/1/14	-9.51E-06	1.21E-05
8/1/14	-1.01E-05	1.28E-05
8/30/14	-7.70E-06	9.77E-06
10/1/14	-1.64E-02	0.020010256
11/1/14	4.37E-05	-5.35E-05
11/29/14	1.84E-05	-2.25E-05
1/1/15	-1.70E-05	2.08E-05
1/31/15	2.37E-06	-2.91E-06

2/28/15	-2.12E-05	2.60E-05
4/1/15	-9.33E-03	0.011185802
5/1/15	-1.97E-05	2.37E-05
5/30/15	-1.75E-05	2.11E-05
7/1/15	-8.81E-03	0.010411986
8/1/15	3.22E-03	-0.003860461
9/1/15	-1.88E-05	2.26E-05
10/1/15	-2.25E-05	2.69E-05
10/31/15	-1.16E-04	0.000139655
12/1/15	-3.88E-06	4.65E-06
1/1/16	-1.31E-05	1.57E-05
1/30/16	-1.44E-03	0.001726066
3/1/16	-8.11E-06	9.70E-06
4/1/16	-4.29E-05	5.13E-05
4/30/16	1.69E-05	-2.02E-05
6/1/16	-3.36E-06	4.02E-06
7/1/16	-1.61E-05	1.92E-05
7/30/16	8.55E-06	-1.02E-05

Συμπεράσματα

Η δημιουργία ενός συστήματος backtesting στρατηγικών Contrarian και Momentum και η συλλογή ενός μεγάλου δείγματος ιστορικών στοιχείων παρέχουν την δυνατότητα ενδελεχούς εξέτασης των επιπτώσεων του κόστους συναλλαγών στις επενδυτικές στρατηγικές. Στο παρόν κείμενο περιγράφηκε αναλυτικά η μεθοδολογία πραγματοποίησης του backtesting, το ψηφιακό περιβάλλον που προσομοιάζεται καθώς και ένα παράδειγμα εφαρμογής backtesting. Να σημειωθεί ότι έχει πραγματοποιηθεί ανάλυση πλήθους πιθανών σεναρίων και τα αποτελέσματα έχουν καταγραφεί. Ενθαρρύνουμε τον αναγνώστη να εκτελέσει ο ίδιος τον αλγόριθμό και να τα αποτελέσματα προσομοιώσεων που θα συνοδεύουν την παρούσα εργασία.