ניתוח נתונים אוטומטי Computational Data Analysis

עבודת גמר - 2019
מרצה: ד״ר יהונתן שלר

מבוא

מטרת המשימה היא לעצב ולפתח מערכת לניתוח וחישוב השקעות ומדדים באמצעות שערי מניות היסטוריים. המערכת מורכבת ממודול, המכיל את רוב הפונקציונליות, ו-4 תוכניות עזר, המאפשרות למשתמשים להשתמש בכל הפונקציונליות.

המשימה

המודול TickersProcess

כתיבת מודול המבצע אחזור (הורדה), שמירה (על דיסק), וניתוח של שערי מניות ההיסטוריים. המודול צריך לספק את הפונקציות הבאות:

fetchTicker(ticker_name,timerange='''')

הפונקציה מביאה את הנתונים עבור מניית ticker_name. במידה והמניה לא קיימת הפונקציה זורקת exception. הפרמטר timerange הינו ריק (בברירת המחדל) ואז מביא נתונים מ100 יום אחרונים, במידה והוא שווה ל 'full', אזי יביא את כל הנתונים ההיסטורים. הפונקציה תישמור קובץ csv עם תוצאות השליפה בספריה data על הדיסק (במידה והספריה לא קיימת - תיצור אותה). במידה והקובץ קיים עם נתונים עבור התקופה המבוקשת, אין להביא שוב את הנתונים. (כדאי לחשוב על איך לציין את שם הקובץ ותאריכי הנתונים שנמצאים בו - ע"מ לאפשר פעולה יעילה)

getDataForTickerInRange(ticker_name,from_date,to_date, data_type)

הפונקציה מקבלת מזהה של מניה (ticker) תאריך התחלה, תאריך סיום, ורשימה של סוג נתונים (data_type) ומחזירה את אוביקט מסוג Pandas.DataFrame שמכיל טבלה עם הנתונים המבוקשים. לטבלה יש את השדות הבאים: תאריך, מזהה מניה, ורשימה של סוגי הנתונים. סוגי הנתונים האפשריים הם: open,close,high,low,volume. במידה ויש סוג נתונים לא תקין, או תאריך לא תקין (בין אם פורמט לא תקין או תאריך שאינו קיים) המודול יזרוק exception. הפונקציה תביא את הנתונים מקובץ ה csv ששומר את הנתונים על הדיסק, במידה והקובץ אינו קיים או שהנתונים בטווח התאריכים הנתון אינו קיים בו - היא תקרא לפונקציה fetchTicker ע״מ להביא את הנתונים.

getProfitForTickerInRange(ticker_name,from_date,to_date, accumulated=false)

הפונקציה מקבלת מזהה של מניה (ticker) תאריך התחלה, תאריך סיום, ופרמטר accumulated שיכול לקבל ערך true או false (ברירת מחדל). הפונקציה מחזירה אוביקט מסוג Pandas.DataFrame שמכיל טבלה עם נתונים הרווח בטווח המבוקש. לטבלה יש את השדות הבאים: תאריך, מזהה מניה, והרווח. רווח יומי מחושב ע״י חישוב היחס בין מחיר הסגירה (close) ביום מסוים למחיר הסגירה ביום המסחר הקודם. במידה ומתבקש רווח מצטבר (accumulated=true) אזי יש להחזיר רווח מצטבר (כחישוב ריבית דריבית). למשל אם הרווח ביום הראשון הוא 1% וביום השני 2% אזי הרווח המצטבר ביום השני הוא 1.01*1.02 = 1.0302. הרווח ידווח באחוזים (אפשר כשבר עשרוני) - בדוגמא דלעיל הערך המצטבר ביום השני יהיה 3.02% (או 0.0302)

במידה ויש סוג נתונים לא תקין, או תאריך לא תקין (בין אם פורמט לא תקין או תאריך שאינו קיים) המודול יזרוק exception. הפונקציה תביא את הנתונים מקובץ הcsv ששומר את הנתונים על הדיסק, במידה והקובץ אינו קיים או שהנתונים בטווח התאריכים הנתון לא נמצאים בקובץ - היא תקרא לפונקציה fetchTicker ע״מ להביא את הנתונים.

geP2vForTickerInRange(ticker_name,from_date,to_date)

הפונקציה מקבלת מזהה של מניה (ticker) תאריך התחלה, תאריך סיום. הפונקציה מחזירה את ערך ה peak_to_valley בטווח המבוקש. בנוסף הפונקציה מחזירה את ערך הpeak את ערך ה valley ואת מספר ימי המסחר שבינהם. Peak to valley מוגדר כפער הגדול ביותר של שיעור הירידה משווי ערך הסגירה (close) הגבוה ביותר של המניה (שיא) לשווי הסגירה הנמוך ביותר <u>לאחר השיא.</u>

במידה ויש סוג נתונים לא תקין, או תאריך לא תקין (בין אם פורמט לא תקין או תאריך שאינו קיים) המודול יזרוק exception. הפונקציה תביא את הנתונים מקובץ הcsv ששומר את הנתונים על הדיסק, במידה והקובץ אינו קיים או שהנתונים בטווח התאריכים הנתון לא נמצאים בקובץ - היא תקרא לפונקציה fetchTicker ע״מ להביא את הנתונים.

<u>תכניות העזר</u>

יש לכתוב 4 תכניות עזר (לפי השמות שיפורטו) כל תכנית תשתמש במודול TickersProcess ותבצע את את הנדרש ממנה:

1. tickersOnDate

התכנית קולטת תאריך (בפורמט yyyy-mm-dd) מהמשתמש ורשימה של tickers (מופרדת בפסיקים) ומדפיסה עבור כל מניה את מחיר הסגירה ואת הרווח (בהשוואה ליום הקודם). במידה ואחת המניות לא קיימת או שיש בעיה בטווח התאריכים תודפס הודעה מתאימה למשתמש.

2. getFileForTicker

התכנית קולטת מהמשתמש תאריך התחלה ותאריך סיום (בפורמט yyyy-mm-dd) שם של ticker, שם (או מסלול ושם) של קובץ ופורמט הקובץ (csv או json). התכנית שומרת את כל נתוני המניה בטווח המבוקש, בפורמט המבוקש לתוך קובץ בשם ובמסלול המבוקש. במידה ואין מסלול - הקובץ ישמר בספריה הנוכחית. במידה והמניה לא קיימת או שיש בעיה בטווח התאריכים, או בעיה בשם/מסלול הקובץ תודפס הודעה מתאימה למשתמש.

3. compareTickersInRange

התכנית קולטת מהמשתמש תאריך התחלה ותאריך סיום (בפורמט yyyy-mm-dd) רשימה של ticker (מופרדת בפסיקים). התכנית תציג טבלה עם סיכומי נתונים עבור כל מניה שברשימה. הטורים של הטבלה הם: שם המניה (ticker), רווח כולל (מחיר סוף יום אחרון בטווח חלקי מחיר סוף יום ראשון בטווח), ערך peak_to_valley, מחיר גבוה ביותר בתקופה, מחיר נמוך ביותר בתקופה, מחיר ממוצע בתקופה, וסטיית תקן של המחיר בתקופה. (מחיר = הכוונה למחיר המניה בסוף היום).

במידה ואחת המניות לא קיימת או שיש בעיה בטווח התאריכים, או בעיה אחרת תודפס הודעה מתאימה למשתמש.

4. tickCompare

התכנית קולטת מהמשתמש תאריך התחלה ותאריך סיום (בפורמט yyyy-mm-dd) רשימה של ticker (מופרדת בפסיקים) ומדד (אחד) להצגה. התכנית תיצור גרף בתקופה הנתונה, המשווה את המדד (הנבחר) עבור המניות שברשימה.

המדדים האפשריים להצגה הם (המשתמש צריך לבחור אחד מהם): מחיר (סוף היום), רווח (יומי), רווח (מצטבר מתחילת תקופה), מחיר מינימום (יומי), מחיר מקסימום (יומי). התכנית תציג טבלה עם סיכומי נתונים עבור כל מניה שברשימה. הטורים של הטבלה הם: שם המניה (ticker), רווח כולל (מחיר סוף יום אחרון בטווח חלקי מחיר סוף יום ראשון בטווח), ערך peak_to_valley, מחיר גבוה ביותר בתקופה, מחיר נמוך ביותר בתקופה, מחיר ממוצע בתקופה, וסטיית תקן של המחיר בתקופה. (מחיר = הכוונה למחיר המניה בסוף היום).

במידה ואחת המניות לא קיימת או שיש בעיה בטווח התאריכים, או בעיה אחרת תודפס הודעה מתאימה למשתמש.

הוראות

<u>תאריך להגשה</u>

תאריך אחרון להגשת העבודה הוא 19-02-2019 בשעה 12:00 (בצהריים)

<u>פורמט הנתונים</u>

ההוראות שלהלן מתארות כיצד יש לעבד את הנתונים.

שמות של מניות (Tickers) - המשתמש יכול להכניס אותם או באותיות גדולות או קטנות. כדאי להמיר בתחילת הקלט לצורה אחידה ע״מ לשמור על אחידות בתכנית. גם הפונקציות צריכות לדעת להתמודד עם שמות מניות ללא קשר לאות גדולה/קטנה

מחירי מניות - מספר עשרוני עם עד 4 ספרות עשרוניות (אחרי הנקודה)

רווח - מיוצג כשבר עשרוני, ביחס למחיר הבסיס. נניח שמחיר מניה בתקופה עלה ב 3.54%

תאריכים - לקליטה מהמשתמש, ולצרכי הדפסה יש להשתמש בתבנית "YYYY-MM-DD", כאשר YYYY היא שנה בת ארבע ספרות, MM הוא חודש דו-ספרתי, ו- DD הוא יום דו ספרתי, כל אחד מהם שוכן אלה מופרדים על ידי מקף. לדוגמה, 31. ינואר 1956. מיוצג כמחרוזת "1956-01-31".

עם זאת, בין קלט ופלט (כלומר, כאשר עובדים עם הנתונים), אפשר לשמור את התאריכים כמו תאריך או אובייקט datetime, אבל אתם רשאים לעשות את זה גם בכל דרך אחרת (כמו מחרוזות, כמו (שנה, חודש, יום) tuples, וכו ').

<u>קבצים</u>

יש לשמור את כל הנתונים בספריית משנה עם השם "data" בספרייה הנוכחית. הקבצים עם נתוני המניות ייקראו TICKER_NAME_fromDATE_toDATE.csv (כאשר TICKER_NAME יוחלף בשם המניה באותיות קטנות והתאריכים ייוצגו כמספרים בני שמונה ספרות בצורה YYYYMMDD).

בעת עבודה עם קבצים בתיקיות המשנה, הקפד להשתמש בפונקציות os.path.join ובפונקציות os.path אחרות, שייתכן שתצטרך לבנות את שמותיהן המלאות.
חשוב: אל תשתמש בקו נטוי הפוך \ "כמפריד לרכיבי נתיב ועבוד רק עם נתיבים יחסיים, שכן התוכניות שלך לא יפעלו במחשב שאינו Windows והנתיבים המוחלטים שלך לא יפעלו.

הערות

תכנון המודול והתוכניות

כדי לאפשר את הבאת נתוני המניות מהאינטרנט, צריך להרשם (בחינם) באתר quandl כאשר הלינק לרישום הינו https://www.quandl.com/sign-up-modal?defaultModal=showSignUp (תבחרו בצורת השימוש האקדמית). השימוש הבסיסי בשירות זה מתואר בתיעוד של ה- API של האתר כאן: https://www.quandl.com/data/EOD-End-of-Day-US-Stock-Prices/usage/quickstart/api (צריך להשתמש ב API ולא באחת המעטפות שמתוארות שם). יש להשתמש בצורה של " Filter by a date range" (כאשר אם רוצים נתונים עבור כל התקופה, יש לתת רק תאריך הסיום ולא לתת תאריך התחלה). באמצעות הגדרה של פרמטר data.csv ניתן לקבל נתונים בצורה של CSV (ולא JSON)

השימוש ב API מצריך שימוש ב api_key. לכל משתמש יש api_key שונה. יש לשמור את ה api_key שלכם בקובץ api.key מהספריה בה נמצאים הקבצים ולהשתמש בו בתכנית. באופן זה ניתן בקלות להחליף את ה api_key שלכם ב api_key של משתמש אחר.

מניות (תקינות) שלא קיים להם ערך בתאריכים מסויימים

לפעמים, עבור חלק מהמניות, לא כל הנתונים קיימים עבור כל יום (למשל אם לא התקיים מסחר במניה מסוימת בתאריך מסוים, או שמניה התחילה רק להסחר החל מתאריך כלשהו). במידה ולא קיים נתון בתאריך מסויים, עבור מידע טבלאי ניתן להשתמש בערך na בהצגה גראפית ממליץ להשתמש בפונקציה fillna ע״מ שלא תהינה נקודות אי-רציפות.

הגשה

יש להגיש את המודול והתוכניות באמצעות מערכת Moodle, מבלי לשלוח קבצים נוספים (כלומר, להגיש רק את חמשת הקבצים .py המתוארים למעלה, אין קובצי נתונים, ולא קבצים אחרים). שם המודול ואת קבצי התוכנית בדיוק כפי שהם נקראו בתרגיל, תוך שימוש רק באותיות קטנות עבור שמות קבצים ותוספים (לדוגמה, התוכנית הרביעית צריכה להיות "tickcompare.py", ולא "HMG.PY", "TickCompare.py", "tickcompare.Py", "tickcompare.PY", וכו ')

ציון

כדי לקבל ציון, כל הקוד חייב לרוץ כנדרש בסביבת Python 3. כדי לוודא כי התקנת Python שלך היא באמת Python 3 (ולא Python 2), אפשר להפעיל PyVer ולראות מה זה מדפיס.

המודול והתוכניות חייבים לעבוד. ניתן להשתמש בכל המודולים בספריית פייתון 3, יחד עם המודולים והחבילות של SciPy. אם אתם מתכננים להשתמש בכל מודול אחר, אנא שאלו באמצעות דואר אלקטרוני אם זה בסדר.

כל התוכניות יופעלו דרך שורת הפקודה (ולכן יש להפעיל בסביבה זו). עבור כל תוכנית ודאו כי כאשר אתם מקלידים python cmd myscript.py זה עובד (כאשר myscripts הוא אחד מתוך 5 קבצים ששלחתם)

כל תכנית מקבלת ניקוד (ציון) בנפרד, באופן אחיד (מודול 20% מהציון, וכל אחת מ-4 התכניות גם 20% מהציון).

זכרו, כל מודול, קובץ ופונקציה חייבים להיות מתועדים היטב. תיעוד מינימלי מורכב מתחילת הקובץ, תחילת מודול, תיעוד עבור כל פונקציה, וכל רכיב קוד לא טריוויאלי.

התרגיל הוא תרגיל עצמאי - זאת אומרת כל סטודנט נדרש לבצע את המשימה בעצמו. אפשרי לשאול ולהתיעץ עם חברים, אבל הקוד הסופי צריך להיות שלך - אתם תהיו גאים בעצמכם מאד בסיום התרגיל! תלמידים שיגישו קוד זהה - שני התלמידים יקבלו 0 נקודות.

בהצלחה רבה! ותשתדלו גם להנות מהדרך...

יהונתן