회귀분석을 활용한

KOSPI200과 K10 관계 도출

인천대학교

과목	R과 함께하는 통계학 응용
담당 교수	김성연
학번	201600779
학과	㈜경제학과, (부)컴퓨터공학부
이름	김영민

목차

1. 서론

- 연구의 목적 및 필요성

핀테크(FINTECH) 기술이 발전하면서 투자에 대한 수요가 빠르게 급증하고 있다. 또한 바이오 산업이 급성장하면서 많은 사람들이 너도 나도 할 것 없이 주식 투자에 많은 관심을 갖고 있다. 그래서 다양한 주식 투자 분석을 행하는 사람이 늘고 있는 추세이다. 하지만 주식 분석이란 매우 변수가 많고 다양해 쉽게 주가를 예측 할 수 없는 한계점이 있다. 따라서 필자는 대표적인 대한민국의 주식 KOSPI200(상위 종목 200개)와 상위 종목 10개 주식인 K10의 영향도를 분석하여 주식 시장의 흐름을 간단하게 알아보고자 한다.

이러한 분석을 하기 위해 KDD 분석 절차(Knowledge Discovery in Database)를 따라서 분석 하고자 한다. KDD분석 방법론은 기술과 데이터베이스를 중심으로 한 Insight를 발견하기 위한 분석 방법론이다. 이 분석 방법은 데이터 선택, 데이터 전처리, 데이터 변환, 데이터 마이닝, 데이터 평가 순으로 5가지 절차로 이뤄져 있다.

이 때 데이터 수집은 Ipython의 Jupyter NoteBook을 이용해 BeautifulSoup 패키지를 사용하여 웹 크롤링을 하여 데이터를 수집하고 분석 모델은 선형회귀분석 모델을 사용하여 얼마나 큰 영향을 미치는지 수치로써 나타내고, 회귀식의 가정에 대해 만족하는지 알아볼 것이다.

- 이론적 배경

김용환 금융공학자는 KOSPI200 지수와 K10 지수의 상관관계를 분석하고 , KOSPI200 과 미국의 주가인 S&P500 의 회귀분석을 실시한 연구를 하였다. 그 결과 KOSPI200 과 K10 지수는 상관관계가 높게 나왔고 또한 KOSPI200 과 S&P500 의 회귀분석 결과 S&P500 이 KOSPI200 에 영향을 준다고 결과가 도출하였다.

또한, 다른 연구는 알고리즘 트레이딩에 대해 강의하는 유튜브 크리에이터인 싱크알고는 python 의 scikit-learn 패키지를 통해 선형회귀를 사용하여 주가 분석을 하고 또한 군집 분석 및 트리 구조를 이용해 주가 분석을 하였다.

마지막으로 메리츠 금융 증권에서 사용하는 머신 러닝의 롱숏 모델을 통해 주식을 고르는 사례가 있다. 이 연구는 과거 10 년에 대해 총 150 개 펀더멘탈 및 주가 관련 데이터와 수익률 간의 연관성 분석을 통하여 증시 상황에 따라 투자 지표를 선택하는 결과를 도출해 낸 연구이다.

필자는 김용한 금융공학자의 연구와 싱크알고의 알고리즘 트레이딩의 강의를 응용하여 K10 과 KOSPI200 의 회귀식을 도출해 낼 것이다.

2. 연구 방법

– 데이터 수집

주가 데이터는 실시간으로 변하기 때문에 데이터의 수집 시점을 잘 잡는 것이 중요하다. 따라서 필자는 2019년 10월 31일 이전의 데이터를 수집 할 것이다. 주가 데이터를 수집하기 위해서 가장 쉽게 접할 수 있는 NAVER 포털에 있는 금융 사이트에 접속하여 python의 BeautifulSoup를 이용해 웹에 있는 정보를 갖고 오는 웹 크롤링 기술을 이용할 것이다.

```
from urllib.request import urlopen
import bs4
def historical_index_naver(index_cd,start_date='',end_date='',page_n=1,last_page=0):#날짜 지정하면 그 날의 종가 지수 출력(kospi)
    if start_date:
        start_date=date_format(start_date)  #date포맷으로 변환
    else:
        start_date=dt.date.today()  #없으면 오늘로 지정
    if end_date:
        end_date=date_format(end_date)
    else:
        end_date=dt.date.today()

    naver_index='https://finance.naver.com/sise/sise_index_day.nhn?code='+index_cd+'&page='+str(page_n)

    source=urlopen(naver_index).read()  #지정한 페이지에서 코드 읽기
    source=bs4.BeautifulSoup(source,'lxml') #뷰티풀 수프로 태그별로 코드 분류

    dates=source.find_all('td',class_='date')  #<td class='date'>태그에서 날짜 수집
    prices=source.find_all('td',class_='number_1')  #<td class='number_1'>태그에서 종가 수집

    for n in range(len(dates)):  #dates 개수만큼 반복
        if dates[n].text.split('.')[0].isdigit(): # 날짜가 숫자형이면
            #날짜처리
            this_date=dates[n].text
            this_date=date_format(this_date)

            if this_date<= end_date and this_date>=start_date:  #start_date 와 end_date 사이에서 데이터 처리
                #종가처리
                this_close=prices[n*4].text
                this_close=this_close.replace(',','')
                this_close=float(this_close)

                #딕셔너리에 저장
                historical_prices[this_date]=this_close
            elif this_date<start_date:
                return historical_prices
    #페이지 네비게이션
    if last_page==0:
        last_page = source.find('td',class_='pgRR').find('a')['href']
        #마지막 페이지 주소추출
        last_page=last_page.split('&')[1]
        last_page=last_page.split('=')[1]
        last_page=int(last_page)
    if page_n<last_page:
        page_n+=1
        historical_index_naver(index_cd,start_date,end_date,page_n,last_page)

    return historical_prices
```

이와 같은 함수를 통해 KOSPI200의 종가를 2018.05.04일부터 2019.10.31일까지의 종가를 모두 받아올 것이다. K10 종목에 삼성전자우 라는 종목이 있는데 이는 삼성전자 주식을 50분에 1로 액면분할 한 것이다. 그래서 2018.05.04일부터 삼성전자우 주식이 발행되기 시작했으므로 2018.05.04일부터 받아오는 이유이다.

다음으로 K10을 받아올 때에는 NAVER 증권에 있는 정보와 추가로 발행주식 수와 유통비율을 불러와야 된다.

http://companyinfo.stock.naver.com/v1/company/c1010001.aspx?cmp_cd=005930

위 사이트를 통해 발행주식 수와 유통비율 그리고 종목의 이름을 받아 올 것이다.

```
def stock_info(stock_cd): # 상장주식과 유동비율 주가 이름을 구하는 함수
    url_float='http://companyinfo.stock.naver.com/v1/company/c1010001.aspx?cmp_cd='+stock_cd
    source = urlopen(url_float).read()
    soup = bs4.BeautifulSoup(source,'lxml')

    # //*[@id="cTB11"]/tbody/tr[7]/td
    tmp = soup.find(id='cTB11').find_all('tr')[6].td.text
    tmp= tmp.replace('₩r','')
    tmp= tmp.replace('₩t','')
    tmp= tmp.replace('₩n','')

    tmp=re.split('/',tmp)

    outstanding=tmp[0].replace(' ','')
    outstanding=outstanding.replace('주','')
    outstanding=outstanding.replace(',','')
    outstanding=int(outstanding)

    floating= tmp[1].replace(' ','')
    floating=floating.replace('%','')
    floating=float(floating)

    name=soup.find(id='pArea').find('div').find('div').find('tr').find('td').find('span').text

    k10_outstanding[stock_cd]=outstanding
    k10_floating[stock_cd]=floating
    k10_name[stock_cd]=name
```

위 함수를 통하여 상위 10개 종목인 K10의 유동비율 주가와 상장주식 수, 이름을 불러온다. 삼성전자, SK하이닉스, 삼성전자우, 삼성바이오로직스, 셀트리온, 현대차, NAVER, 현대모비스, LG화학, 신한지주 가 우리나라 KOSPI의 10개 주식 목록이다. 그리고 종가 지수를 구하기 위해 다시 위 네이버 금융 사이트

'http://companyinfo.stock.naver.com/v1/company/c1010001.aspx?cmp_cd='+stock_cd에 접속한 후 주가 별 , 날짜 별 종가를 받아올 것이다.

```python
def historical_stock_naver(stock_cd,start_date='',end_date='',page_n=1,last_page=0):   #날짜 지정하면 그 날의 종가 지수 출력
    if start_date:
        start_date=date_format(start_date)  #date포맷으로 변환
    else:
        start_date=dt.date.today()  #없으면 오늘로 지정
    if end_date:
        end_date=date_format(end_date)
    else:
        end_date=dt.date.today()

    naver_stock='https://finance.naver.com/item/sise_day.nhn?code='+stock_cd+'&page='+str(page_n)

    source=urlopen(naver_stock).read()  #지정한 페이지에서 코드 읽기
    source=bs4.BeautifulSoup(source,'lxml') #뷰티풀 수프로 태그별로 코드 분류

    dates=source.find_all('span',class_='tah p10 gray03')  #날짜 수집
    prices=source.find_all('td',class_='num')  #종가 수집

    for n in range(len(dates)):  #dates 개수만큼 반복
        if len(dates)>0:
            #날짜처리
            this_date=dates[n].text
            this_date=date_format(this_date)

            if this_date<= end_date and this_date>=start_date:  #start_date 와 end_date 사이에서 데이터 처리
                #종가처리
                this_close=prices[n*6].text
                this_close=this_close.replace(',','')
                this_close=float(this_close)

                #딕셔너리에 저장
                #historical_prices=dict()
                historical_prices[this_date]=this_close
            elif this_date<start_date:
                return historical_prices
    #페이지 네비게이션
    if last_page==0:
        last_page = source.find_all('table')[1].find('td',class_='pgRR').find('a')['href']
        #마지막 페이지 주소추출
        last_page=last_page.split('&')[1]
        last_page=last_page.split('=')[1]
        last_page=float(last_page)
    if page_n<last_page:
        page_n+=1
        historical_stock_naver(stock_cd,start_date,end_date,page_n,last_page)

    return historical_prices
```

위 historical_stock_naver 함수를 통해 종가지수를 받아오고 그 결과는 데이터 프레임 형식으로 저장한다.

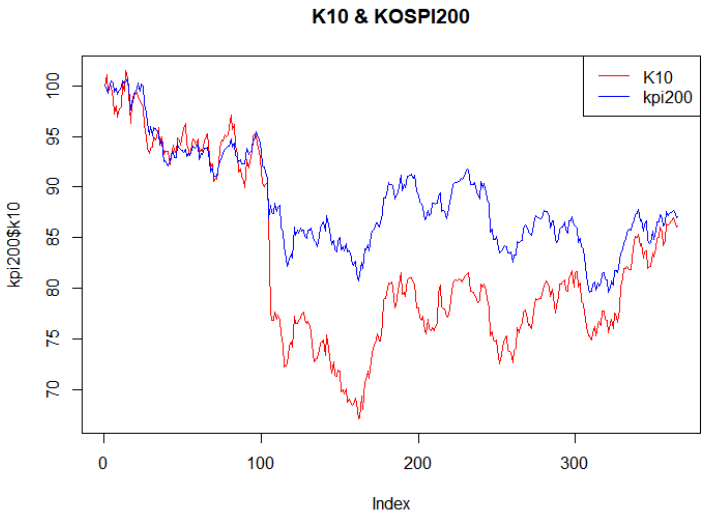
	Outstanding	Floating	Price	Name
000660	728002365	73.63	83000.0	SK하이닉스
005380	213668187	65.59	158000.0	현대차
005930	5969782550	74.62	51900.0	삼성전자
005935	5969782550	74.62	41050.0	삼성전자
012330	95306694	68.00	235000.0	현대모비스
035420	164813395	77.15	717000.0	NAVER
051910	70592343	64.28	341000.0	LG화학
055550	474199587	80.56	46800.0	신한지주
068270	128336328	75.83	250000.0	셀트리온
207940	66165000	24.76	359500.0	삼성바이오로직스

– 데이터 전처리 및 변환

위와 같이 불러온 데이터를 분석하기 쉬운 방식으로 변환하기 위해 전처리 과정이 필요하다. KOSPI200은 종가가 316.75인 반면에 K10 지수의 가장 낮은 종목의 종가도 46800이어서 단면적으로 분석하기에는 어려움이 있다. 이를 통해 데이터들을 가공해서 표준화를 시켜야 한다. 그 방법으로 지수 방법을 택하였다. 지수는 비교시점 시가총액/기준시점 시가총액 * 100 의 식을 가지며 2018.5.4일이 시작점이기 때문에 기준시점을 2018.5.4일로 잡고 계산을 한다.

Unnamed: 0	K10	kpi200
2018-05-04	100.000000	100.000000
2018-05-08	101.155225	99.722178
2018-05-09	99.319717	99.264404
2018-05-10	100.124989	100.176796
2018-05-11	99.986664	100.555643

위와 같은 라는 결과가 나온다.



위의 데이터프레임의 결과를 그래프로 나타냈을 때 위와 같은 그래프가 그려진다.

3. 연구 결과

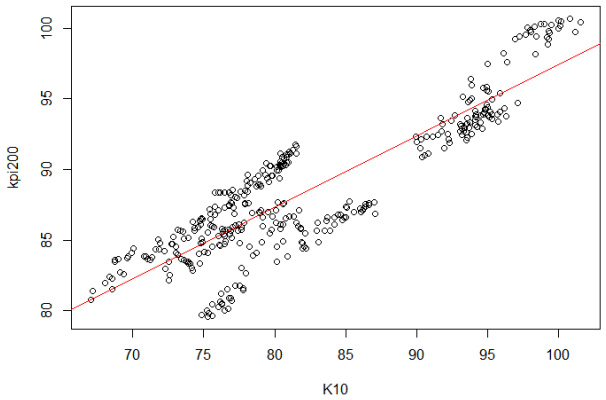
K10 지수가 얼마나 KOSPI200 지수에 영향을 끼치는지 알아보기 위해서 회귀모델을 분석 모델로 선정했다. 따라서 KOSPI200 지수를 종속변수로 K10 지수를 독립변수로 설정하여 회귀 분석을 실시하였다.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 46.82813    1.15963   40.38   <2e-16 ***
K10          0.50585    0.01396   36.23   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.313 on 363 degrees of freedom
Multiple R-squared:  0.7834,    Adjusted R-squared:  0.7828
F-statistic:  1313 on 1 and 363 DF,  p-value: < 2.2e-16
```

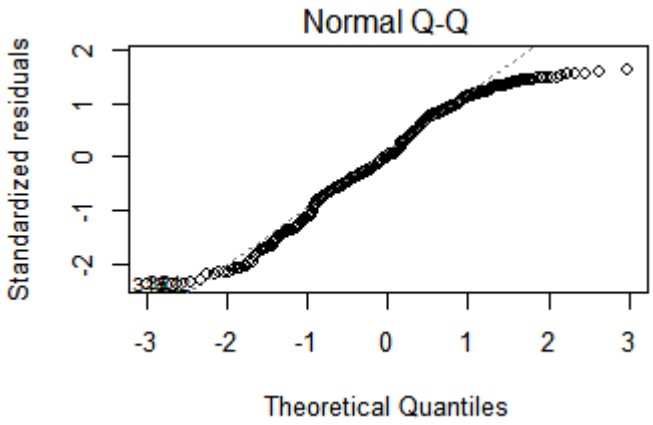
위와 같은 결과가 도출되어 독립변수인 K10의 변수의 통계적 유의성은 0.05보다 작으므로 유의하다고 말할 수 있다. 또한, 모델의 p-value값이 0.05보다 작으므로 통계적으로 유의하다 할 수 있다. 이 모델의 회귀식은 kpi200=46.82813+0.50585*K10 이라는 식이 도출된다. 이 모델의 설명력은 78.34%로 설명력이 높게 나왔다.

이를 그래프로 한 번 나타내보았다.



빨간선은 회귀 직선이며 실제 데이터는 회귀식의 선과 거의 유사하게 분포되어 있음을 알 수 있다.

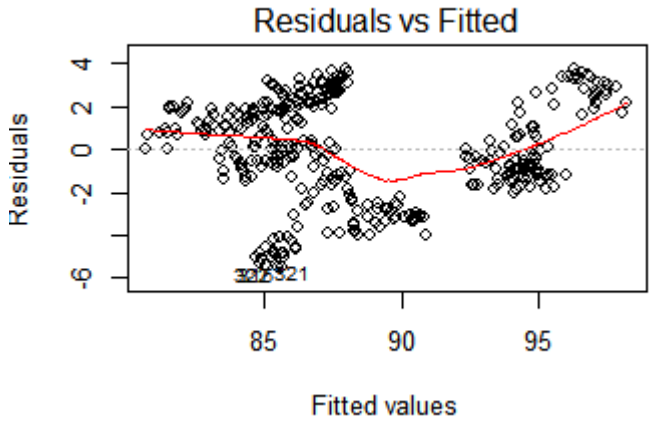
이어서 회귀식의 기본 가정 4가지를 만족하는지 알아볼 것이다. 이는 정규성, 독립성, 등분산성, 선형성이 있다. 이 때 독립성은 단순선형회귀 분석에는 중요한 가정이 아니므로 제외한다. 정규성은 오차의 분포가 정규분포를 따른다라는 것이다. 이는 Q-Q Plot으로 확인 할 수 있다.



위와 같은 그래프가 나오므로 정규성을 만족한다고 할 수 있다.

또한 선형성은 입력변수와 출력변수의 관계가 선형이다 라는 가정인데 이는 위에 그렸던 산점도로 상관관계가 높게 나왔기 때문에 이 가정을 성립한다.

다음으로 등분산성이다. 등분산성은 오차의 분산이 입력변수와 무관하게 일정하다 라는 것이다.



위와 같은 그래프로 등분산성은 만족하지 못한다는 것을 알 수 있다.

따라서 위 연구 결과는 회귀식이 kpi200=46.82813+0.50585*K10으로 도출되었고, 두 변수 사이의 선형적인 관계가 있음을 알아냈다. 하지만 한계점으로는 주가가 시계열 데이터이기 때문에 등분산성을 만족하지 못하였다.

4. 참고문헌

서론

https://www.youtube.com/user/ATssarabiya

https://creativeprm.tistory.com/84

금융공학 레시피 - 김용환 저

본론

http://companyinfo.stock.naver.com/v1/company/c1010001.aspx?cmp_cd=005930

https://finance.naver.com/sise/sise_index.nhn?code=KPI200)