|  |  |
| --- | --- |
| **프로젝트 결과 보고서** | |
| 작 품 명 | 기술적 분석 지표를 이용한 주식 데이터 분석 |
| 팀 명 | (6조) 골든크로스 |
| 제 작 기 간 | 2018년 8월 06일 ~ 2018년 8월 16일 |
| 프로젝트 매니저 | 박지용 (인) |

|  |  |  |  |
| --- | --- | --- | --- |
| 조 원 | | | |
| No. | 성 명 | 연 락 처 | E-mail |
| 1 | 정승원 | 010-3958-0106 | seungwonjung0106@gmail.com |
| 2 | 박지용 | 010-4941-6000 | archymi77@gmail.com |
| 3 | 정해명 | 010-9119-2831 | hmchung2@gmail.com |
| 4 | 김명회 | 010-9771-2540 | kmh9771@gmail.com |
| 5 | 김준국 | 010-3724-7759 | ehehwnwjs546@naver.com |

**1. Team 소개**

**1-1) 팀명**

골든크로스는 주식용어로서 단기 이동평균선이 장기 이동평균선을 아래에서 위로 돌파하는 것을 이야기하며 이는 주가가 강세로 접어들었다고 볼 수 있음을 의미합니다.

골든크로스를 팀명으로 선정한 이유는 현재 진행하고 있는 한양대학교 빅데이터 과정이라는 단기 이동평균선이 저희가 그동안 받아온 교육과 시너지를 만들어 증시에서 이야기하는 골든크로스처럼 장기이동평균선을 아래에서 위로 돌파하기를 바라는 마음과 각오를 담아 선정하였습니다.

**1-2) 주요 업무 : 사용 언어 및 사용 프로그램**

**데이터 서버 구축 : Docker, Linux(CentOS7), MariaDB, Apache Tomcat**

**데이터 수집 : Python**

**데이터 정제&가공 : Logstash, Elasticsearch**

**데이터 시각화 : Kibana, R Studio**

**보고서 작성 : Google word processor**

**1-3) 업무 분담**

[PM] 박지용 : 데이터 서버 구축 / 데이터 수집 / 데이터 정제&가공 / 데이터 시각화 / 보고서 작성

[조장] 정승원 : 데이터 서버 구축 / 데이터 정제&가공 / 보고서 작성

[조원] 정해명 : 데이터 서버 구축 / 데이터 수집 / 데이터 정제&가공 / 데이터 시각화 / 보고서 작성

[조원] 김명회 : 데이터 서버 구축 / 데이터 정제&가공 / 데이터 시각화 / 보고서 작성

[조원] 김준국 : 데이터 서버 구축 / 데이터 정제&가공 / 보고서 작성

**2. Project 주제 : 주식 데이터를 객관적인 지표로 분석한다**

**2-1) 주제의 개요**

기술적 분석 지표를 통해 주식 데이터를 분석하여 각 주식종목의 매수,매도 신호를 판별하여 투자자의 의사결정에 도움을 준다.

**3. Project 개요**

**3-1) 소개**

주식 데이터를 바탕으로 객관적인 지표를 바탕으로 매수, 매도 신호를 포착하여 투자자에게 도움을 준다. 분석에 사용된 기술적 분석 지표는 다음과 같다.

1. 이동평균선(Moving Average Line) : 일정기간 동안의 주가를 산술 평균한 값인 주가이동평균을 연결한 선

(1) 5일 이동평균선 : 해당일을 포함하여 과거 4일동안의 가격의 평균

(2) 20일 이동평균선 : 해당일을 포함하여 과거 19일동안의 가격의 평균

(3) 60일 이동평균선 : 해당일을 포함하여 과거 59일동안의 가격의 평균

1. 볼린저 밴드(Bollinger Band): 볼린저 밴드는 하기와 같이 총 3개의 선으로 이루어져 있다

(1) 중심선 : 20일 이동평균선

(2) 상한선 : 20일 이동평균선 + 2σ (σ : 20일 동안의 표준편차)

(3) 하한선 : 20일 이동평균선 - 2σ (σ : 20일 동안의 표준편차)

1. 엔벨로프(Envelope) : 엔벨로프는 총 2개의 선으로 이루어져 있다.

(1) 엔벨로프 상단 : 20일 이동평균선 + (20일 이동평균) \* 0.03

(2) 엔벨로프 하단 : 20일 이동평균선 - (20일 이동평균) \* 0.03

1. 골든크로스(Golden Cross) : 주가나 거래량의 단기 이동평균선이 중장기 이동평균선을 돌파해 올라가는 현상

(1) 단기 골든크로스 : 5일 이동평균선이 20일 이동평균선을 돌파하는 것

(2) 중장기 골든 크로스 : 20일 이동평균선이 60일 이동평균선을 돌파하는 것

1. 이격도(Disparity) : 주가와 이동평균선 사이가 얼마나 떨어져 있는지를 나타내는 지표로, 당일의 주가를 이동평균치로 나눠 계산하며 단기 투자시점을 포착하는 지표로 활용

(1) 이격도과열 : 이격도가 100% 이상이면 당일의 주가가 이동평균선보다 위에, 100% 이하면 아래에 위치한 상태를 의미한다. 즉 이격도가 100% 이상이면 단기적으로 주가가 최근 상승했음을 뜻한다.

- 여기서는 당일 주가를 20일 이동평균값으로 나눈 비율이 120% 이상일 경우의 종목을 찾는다.

- 해당값이 120% 이상일 경우 “매수” 신호로 판단하고 미만일 경우 “중립” 으로 판단한다.

**3-2) 주제 선정 이유**

하루가 다르게 변하는 주가를 대하면서 투자자에게 객관적인 지표를 바탕으로 투자정보를 제공하고 싶었습니다.

또한 Kibana와 Elasticsearch를 동시에 사용하며 보다 시각적으로 정보를 제공하고 싶었습니다.

**4. 활용 범위**

**4-1) 주식 투자&증권 분석 분야**

증권사의 HTS(Home Trading System)에 서브 프로그램으로서 증시에 대한 정보를 사용자에게 제공해줄 수 있을것이라고 사료 됩니다.

해당 조건에 맞는 주식을 매수, 매도 신호로 간단하게 분류함으로써 여러 지표로 분석되는 자료를 복잡한 수학적 지식 없이 사용자가 매수 매도에 대한 판단할 수 있도록 돕습니다.

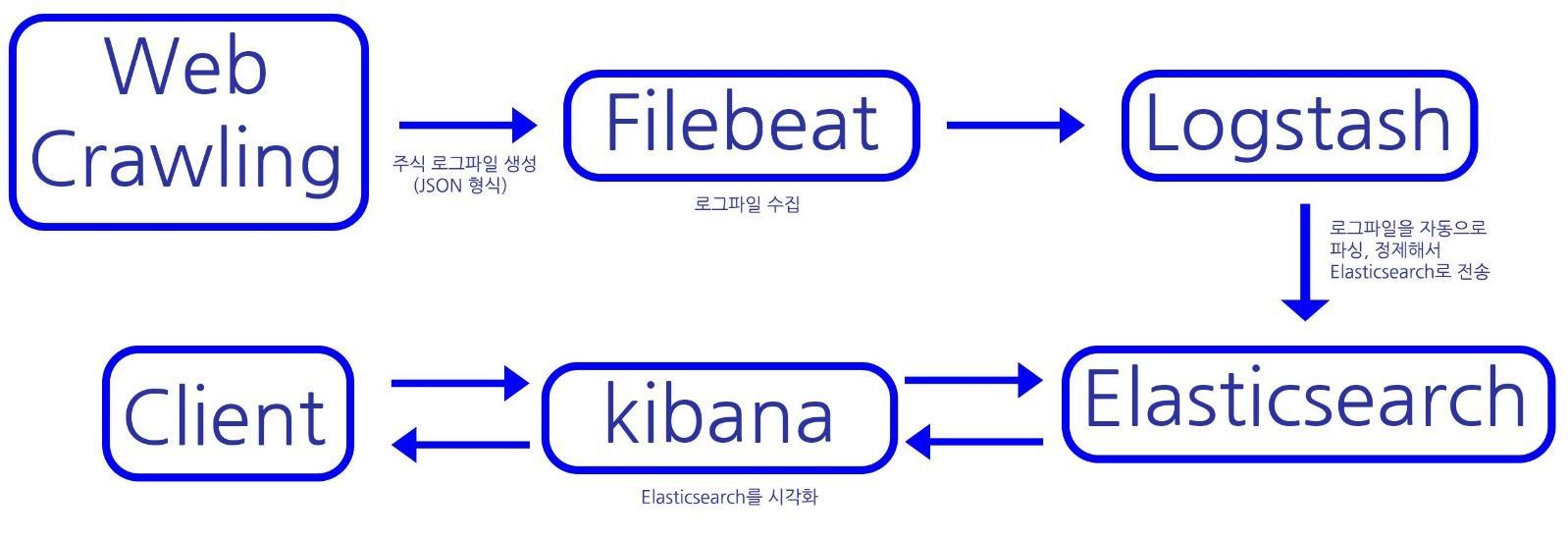
**4-2) 증권 방송 분야**

증권 관련 매체의 특성상 각종 자료를 분석 하여 시청자들에게 정보를 제공하는 업무를 주로 하게 되는데 이러한 곳에 저희의 분석과정이 활용이 될 수 있을것이라 사료 됩니다.

아울러 저희의 분석 내용은 실재하는 데이터를 기반으로 수집하여 데이터를 정제한 것이므로 조사자의 주관적 판단이 배제되었습니다. 그렇기 때문에 사용자에게 보다 신뢰성 있는 정보를 제공할 수 있다는 장점이 있습니다.

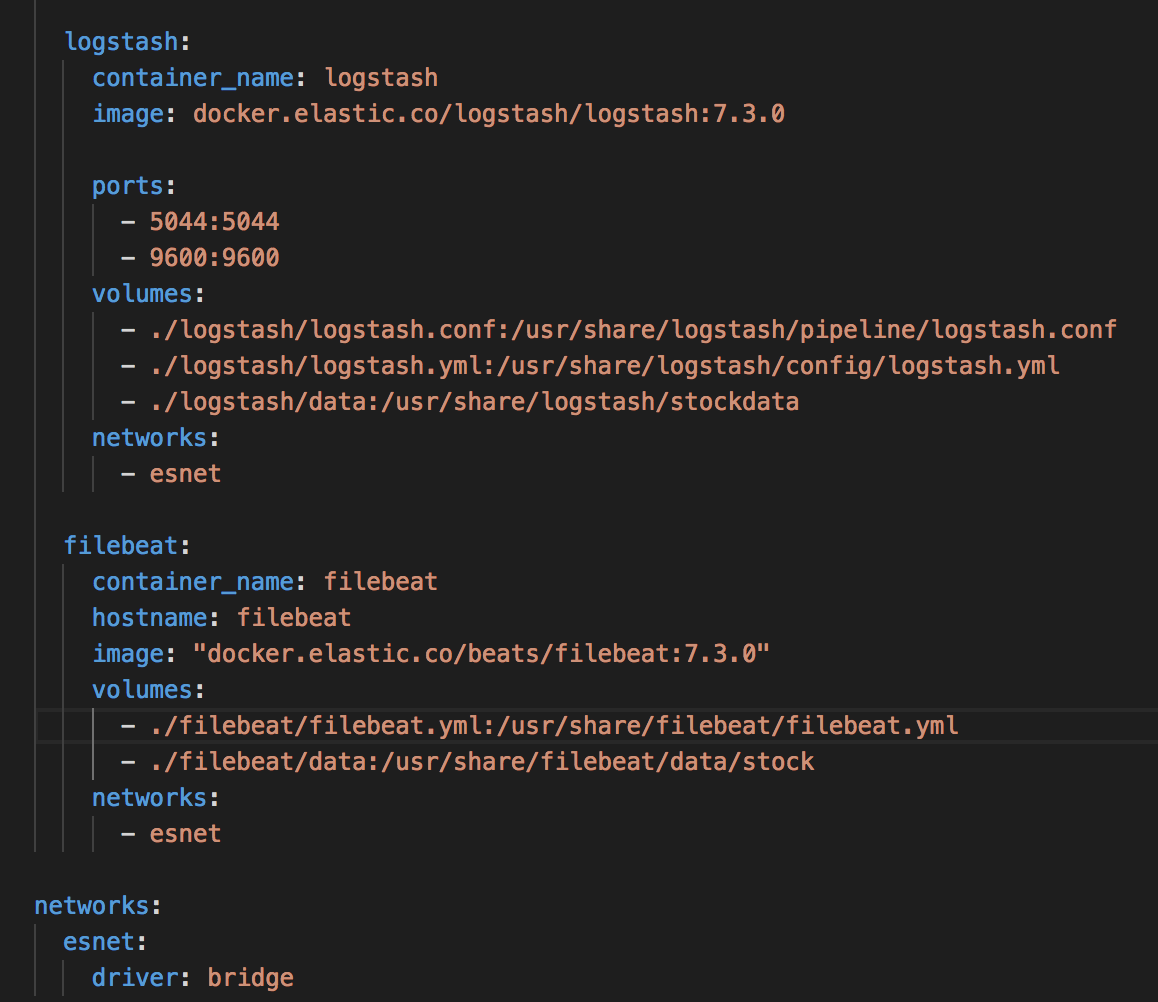
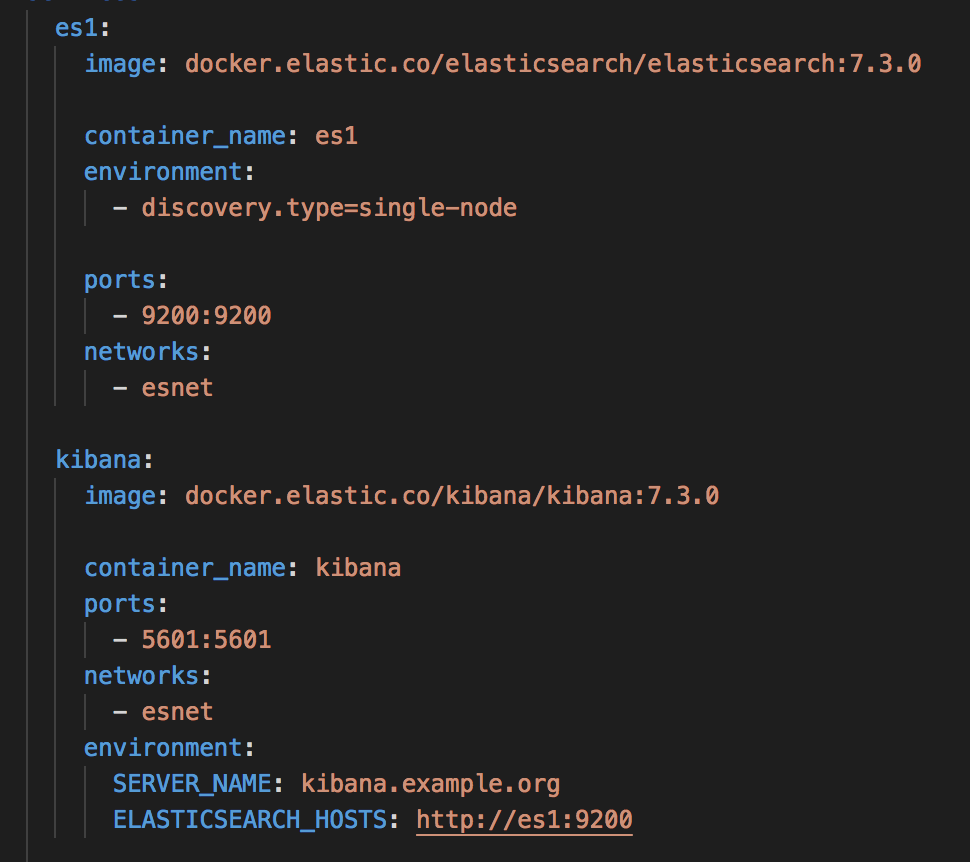
**5. 개발 내용**

**5-1) 서비스 흐름도**

****

**5-2) 사용기술**

|  |  |  |  |
| --- | --- | --- | --- |
| **구분** | **이름** | **버전** | **목적** |
| **사용툴** | **Docker** | **19.03.1** | **가상 Container 역할** |
| **Python** | **3.7.2** | **웹 크롤링(crawling)** |
| **서버통신** | **Kibana** | **7.3.0** | **Database 시각화 tool** |
| **ElasticSearch** | **7.3.0** | **분산형 Datbase** |
| **Logstash** | **7.3.0** | **Filebeat에서 보내온 Log 처리** |
| **Filebeat** | **7.3.0** | **크롤링 결과로 만들어지는 로그수집** |

****

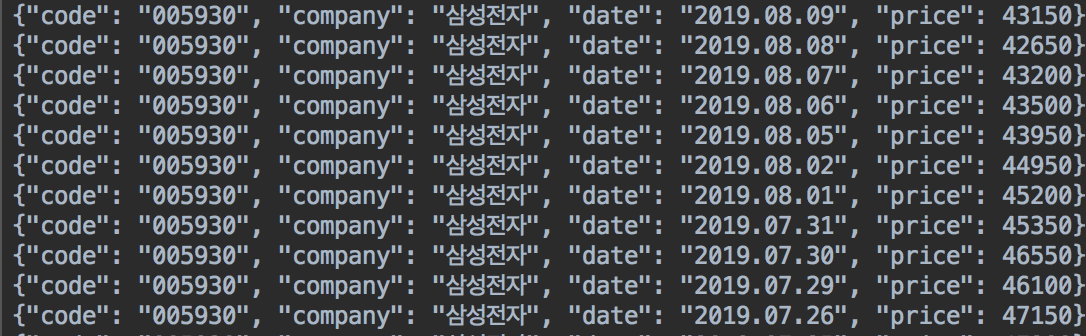
**(\* Docker-compose 을 이용하여 명령어 한줄로 환경이 자동으로 구축되게 하였음. Elasticsearch는 단일노드로 구성)**

**(\* Kibana, ElasticSearch, Logstash, Filebeat 각각 다른 Container에 구축함으로써 최대한 실무환경과 비슷하도록 구성.)**

**(\* Docker container 에 있는 설정파일들은 docker-compose volume으로 호스트와 연결함으로써 수정용이하도록 구성.)**

**5-3) 데이터 구조**

**주식데이터 : 종목코드, 종목, 날짜, 가격 (JSON 형식)**

****

**5-4) 산출물**

**5-4-1) KIBANA**

|  |
| --- |
| **C:\Users\Jiyong\AppData\Local\Microsoft\Windows\INetCache\Content.Word\dasdsad.jpg**  첫화면으로 해당기간의 최고가, 최저가와 구현한 알고리즘에 따라 매수, 매도를 추천해줍니다. |
| 총 조회할 수 있는 종목은 코스피 기준 시가총액 상위 20위 종목입니다.C:\Users\Jiyong\AppData\Local\Microsoft\Windows\INetCache\Content.Word\KakaoTalk_20190815_225832670_01.png |

|  |
| --- |
| kibana의 filter을 이용하여 company : “삼성바이오로직스" 처럼 조회를 하면 화면이  자동으로 해당 회사에 관한 정보를 불러옵니다. |
| 이동평균선과 볼린저밴드에 대한 설명과 이를 시각적으로 나타내어 이해에 도움을 줍니다. |

|  |
| --- |
| **C:\Users\Jiyong\AppData\Local\Microsoft\Windows\INetCache\Content.Word\KakaoTalk_20190815_225832670_03.png**  골든크로스, 이격도 과열 에 대한 설명과 엔벨로프를 시각적으로 나타내어 이해에 도움을 줍니다. |

**5-4-2) R STUDIO**

**2018년 5월 28일 ~ 2019 8월 14일까지 데이터 기반으로 15개의 주식회사 포트폴리오 분석**

**#평균 주가 - 선정 종목별 시황의 추세를 비교하기 위해 분석함**

|  |  |  |
| --- | --- | --- |
|  | | |
| **##** | 스톡 | 평균값 |
| ## 1 | LG생활건강 | 1259410.00 |
| ## 2 | LG화학 | 352183.33 |
| ## 3 | NAVER | 130379.67 |
| ## 4 | POSCO | 272426.67 |
| ## 5 | SK텔레콤 | 257850.00 |
| ## 6 | SK하이닉스 | 74423.67 |
| ## 7 | 기아차 | 35611.67 |
| ## 8 | 삼성SDI | 227900.00 |
| ## 9 | 삼성바이오로직스 | 374661.67 |
| ## 10 | 삼성전자 | 44747.67 |
| ## 11 | 삼성전자우 | 36360.17 |
| ## 12 | 셀트리온 | 229668.33 |
| ## 13 | 신한지주 | 43555.33 |
| ## 14 | 현대모비스 | 215165.00 |
| ## 15 | 현대차 | 125651.33 |

**# 주식성장률**

**#회사별 주식성장률 평균값 및 성장률 표준편차**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **##** | 스톡 | 평균\_성장률 | 표준편차 |
| ## 1 | LG생활건강 | -1.645572e-04 | 0.02166697 |
| ## 2 | LG화학 | -8.878137e-05 | 0.02002071 |
| ## 3 | NAVER | 2.863622e-04 | 0.02015376 |
| ## 4 | POSCO | -1.638650e-03 | 0.01804450 |
| ## 5 | SK텔레콤 | 2.523094e-04 | 0.01375776 |
| ## 6 | SK하이닉스 | -4.223717e-04 | 0.02315440 |
| ## 7 | 기아차 | 1.199280e-03 | 0.01879749 |
| ## 8 | 삼성SDI | 9.378279e-04 | 0.02368723 |
| ## 9 | 삼성바이오로직스 | -7.992787e-04 | 0.03175219 |
| ## 10 | 삼성전자 | -4.832872e-04 | 0.01534366 |
| ## 11 | 삼성전자우 | -3.438954e-04 | 0.01665676 |
| ## 12 | 셀트리온 | -1.491686e-03 | 0.02727664 |
| ## 13 | 신한지주 | -3.595927e-04 | 0.01383616 |
| ## 14 | 현대모비스 | 3.439917e-04 | 0.02017955 |
| ## 15 | 현대차 | -6.095424e-05 | 0.01815241 |

**Equal Allocation 15개 회사에 같은 비중으로 투자할 경우**

Equal Allocation 란? : 선정한 다수의 종목들의 분산투자시 리스크와 수익률 분석

목적 : 분산투자를 위해 리스크 분석 및 기대 수익률 분석

**#기대 수익률:**

alloreturn = t(x)%\*%assetmean1

**## [1,] -0.0001888855**

**#위험도**

allorisk = sqrt(t(x) %\*% sigma1 %\*% x)

**## [1,] 0.009771843**

|  |  |  |
| --- | --- | --- |
|  | | |
| **##** | 스톡 | 분배율 |
| ## 1 | LG생활건강 | 0.0399407381 |
| ## 2 | LG화학 | -0.0018232432 |
| ## 3 | NAVER | 0.0932603145 |
| ## 4 | POSCO | 0.0353427913 |
| ## 5 | SK텔레콤 | 0.3311528671 |
| ## 6 | SK하이닉스 | 0.0148766859 |
| ## 7 | 기아차 | 0.1308642481 |
| ## 8 | 삼성SDI | -0.0043975268 |
| ## 9 | 삼성바이오로직스 | 0.0006323605 |
| ## 10 | 삼성전자 | -0.0058336492 |
| ## 11 | 삼성전자우 | 0.1636546562 |
| ## 12 | 셀트리온 | 0.0590755023 |
| ## 13 | 신한지주 | 0.1680807557 |
| ## 14 | 현대모비스 | -0.0774667682 |
| ## 15 | 현대차 | 0.0526402677 |

**최소 위험도 분배도 및 수치 찾기**

최소위험도 분배도 및 수치 찾기 란? : 수집된 회사별 데이터를 위험도를 분석하여 수치화,시각화 하는 과정

목적 : 분석하고자 하는 종목들의 위험도 및 수익성 분석, 리스크 최소화 수익률 최대화 목적 달성

sigma = cov(r)

sigmainv = solve(sigma)

ones = matrix(rep(1,15))

weights = (sigmainv1 %\*% ones) / as.numeric(t(ones) %\*% sigmainv1 %\*% ones)

thetable3 = data.frame(스톡 = colnames(wide\_DF[,-1]), 분배율 = weights)

**#최소위험도의 기대 수익률**

최소위험도의 기대 수익률 이란? : 상단의 동분배 그레프에 나타난 위험도가 가장 낮으며 수익률이 높은 종목 선정

선정 종목 : SK 텔레콤

minrisky = t(weights) %\*% assetmean1

**## [1,] -3.991611e-05**

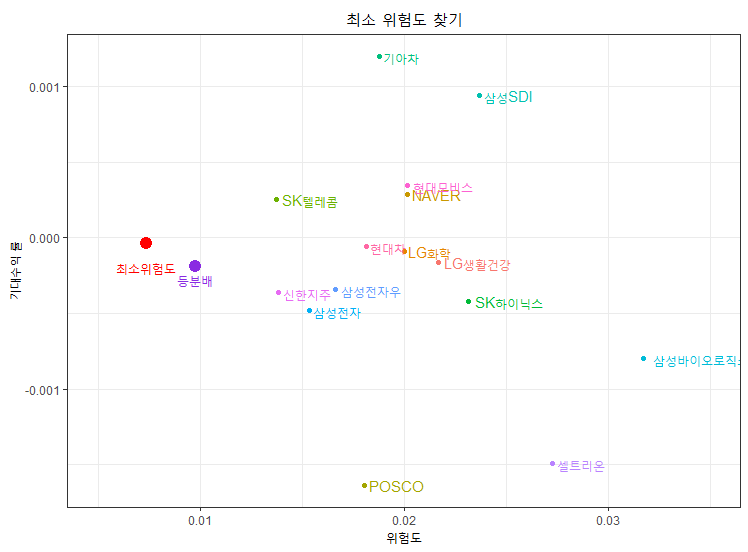
**#최소위험도 수치**

최소위험도 수치란? : 상단의 동분배 그레프에 나타난 X축이 나타내는 위험도가 가장 낮은 종목

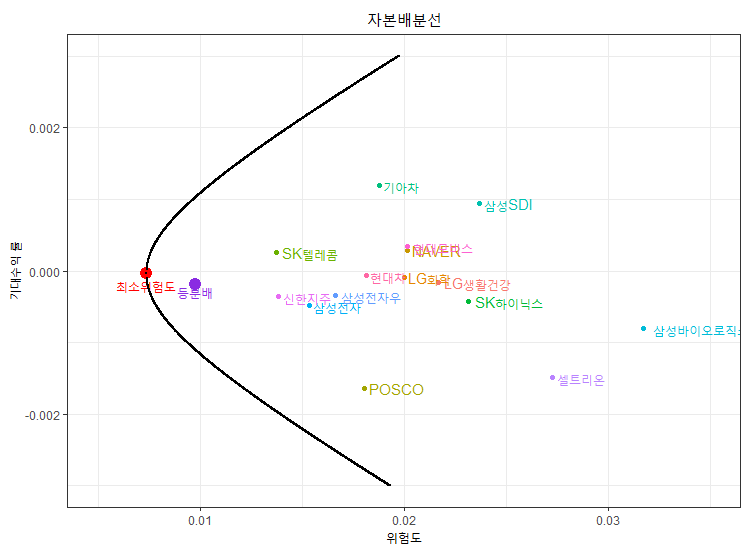
선정 종목 : SK 텔레콤

minriskx = sqrt(t(weights) %\*% sigma1 %\*% weights)

**## [1,] 0.007386769**



**자본배분선 시각화 하기**



**코스피지수 (KOPSI)를 통해 주식 알파랑 베타 분석**

thedata = cbind(df$date , KOSIPI = kospi$KOSPI ,df[,-1])

df = thedata

r <- (df[-1,2:ncol(df)]-df[-nrow(df),2:ncol(df)])/df[-nrow(df),2:ncol(df)]

alpha = rep(0,15)

beta = rep(0,15)

MSE = rep(0,15)

for(i in 2:16){

q <- lm(as.matrix(r[i])~ as.matrix(r[1]))

alpha[i-1] = q$coefficients[1]

beta[i-1] = q$coefficients[2]

MSE[i-1] = summary(q)$sigma^2

}

names(alpha) = colnames(r[2:16])

names(beta) = colnames(r[2:16])

names(MSE) = colnames(r[2:16])

alphadata = data.frame(회사 = colnames(r[,-c(1)]),알파 = alpha, 베타 = beta, MSE = MSE)

rownames(alphadata) = NULL

**alphadata**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **##** | 스톡 | 알파 | 베타 | MSE |
| ## 1 | LG생활건강 | 7.095000e-04 | 1.11463056 | 0.0003758839 |
| ## 2 | LG화학 | 8.586937e-04 | 1.20825569 | 0.0002903670 |
| ## 3 | NAVER | 9.127644e-04 | 0.79881161 | 0.0003586702 |
| ## 4 | POSCO | -7.132277e-04 | 1.18013374 | 0.0002200334 |
| ## 5 | SK텔레콤 | 3.223722e-04 | 0.08934665 | 0.0001893019 |
| ## 6 | SK하이닉스 | 7.557270e-04 | 1.50235557 | 0.0003650640 |
| ## 7 | 기아차 | 1.735008e-03 | 0.68318029 | 0.0003187885 |
| ## 8 | 삼성SDI | 2.030868e-03 | 1.39388525 | 0.0004141674 |
| ## 9 | 삼성바이오로직스 | 6.615592e-05 | 1.10363461 | 0.0009183100 |
| ## 10 | 삼성전자 | 5.337128e-04 | 1.29691645 | 0.0001073980 |
| ## 11 | 삼성전자우 | 6.152628e-04 | 1.22315448 | 0.0001637962 |
| ## 12 | 셀트리온 | -7.087806e-04 | 0.99838981 | 0.0006701773 |
| ## 13 | 신한지주 | 1.369411e-04 | 0.63319842 | 0.0001613761 |
| ## 14 | 현대모비스 | 1.102268e-03 | 0.96698189 | 0.0003369703 |
| ## 15 | 현대차 | 6.189644e-04 | 0.86705778 | 0.0002730406 |

**#Vasicek's 모형을 통해 더 좋은 예측**

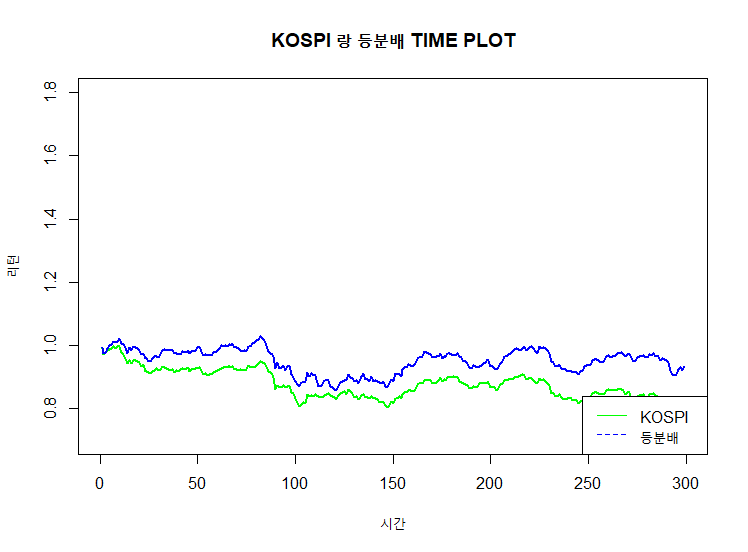
for(i in 1:30){

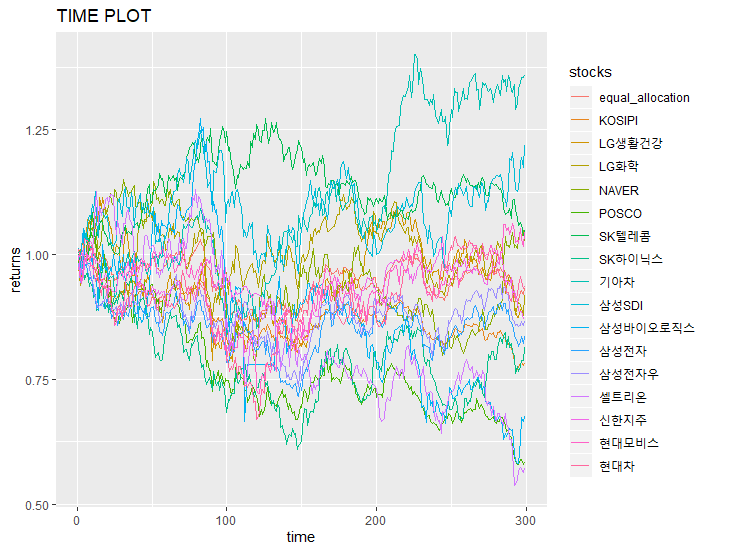
beta\_adj1[i] <- var\_beta1[i]\*mean(beta1)/(var(beta1)+var\_beta1[i]) +

var(beta1)\*beta1[i]/(var(beta1)+var\_beta1[i])}

|  |  |  |  |
| --- | --- | --- | --- |
| **##** | 스톡 | 베타 | 예측 베타 |
| ## 1 | LG생활건강 | 1.11463056 | 1.11463056 |
| ## 2 | LG화학 | 1.20825569 | 1.20825569 |
| ## 3 | NAVER | 0.79881161 | 0.79881161 |
| ## 4 | POSCO | 1.18013374 | 1.18013374 |
| ## 5 | SK텔레콤 | 0.08934665 | 0.08934665 |
| ## 6 | SK하이닉스 | 1.50235557 | 1.50235557 |
| ## 7 | 기아차 | 0.68318029 | 0.68318029 |
| ## 8 | 삼성SDI | 1.39388525 | 1.39388525 |
| ## 9 | 삼성바이오로직스 | 1.10363461 | 1.10363461 |
| ## 10 | 삼성전자 | 1.29691645 | 1.29691645 |
| ## 11 | 삼성전자우 | 1.22315448 | 1.22315448 |
| ## 12 | 셀트리온 | 0.99838981 | 0.99838981 |
| ## 13 | 신한지주 | 0.63319842 | 0.63319842 |
| ## 14 | 현대모비스 | 0.96698189 | 0.96698189 |
| ## 15 | 현대차 | 0.86705778 | 0.86705778 |

**Time Plot (2018 5/28 ~ 2019 8/14)**





**6.향후 개발 계획**

**6-1) 애로사항**

1) 주식 데이터 분석할때 사용되는 주요 알고리즘에 대한 이해가 부족.

2) 확보된 데이터를 가상환경을 구축하여 분석시 성능 저하 및 한글 인식 오류 발생.

3) 팀원들간의 역량 차이로 인한 업무 분담화에 대한 어려움

4) 지리적 제약 이로인한 프로젝트 진행에 어려움

**6-3) 해결 방안**

1) 프로젝트에 사용할 알고리즘에 대한 학습을 통해 프로젝트에 적용.

2) 윈도우환경에서 Docker를 활용하여 성능 저하 및 한글 인식 오류 해결함.

3) 업무 분담이 아닌 획일화된 업무 진행 으로 조직 역량 강화

4) 온라인 기반의 협업 툴을 활용하여 지리적 제약 조건 해결함

**6-2) 추가 필요 사항**

1. 웹크롤링을 통해 Naver Finance에서 해당일자 종가기준 가격만 가져와서 모든 시간의 데이터는 가져오지 못했습니다. 향후 키움 API 등 증권사의 오픈API를 통해서 응용프로그램에서 실시간으로 주식 데이터를 생성하면 Filebeat 혹은 Metircbeat로 실시간으로 가져와 Logstash에서 Parsing을 해서 Elasticsearch로 보내주는 것을 구현하는것을 계획으로 하고있습니다.
2. 수학적인 알고리즘과 분석 기법에 대한 공부가 좀 더 필요할 것 같습니다. 또한 루씬 검색기반의 Elasticsearch에 대한 지식이 더 필요하다 판단 하였습니다.
3. 해당 프로젝트에 활용된 분석 기술을 통해 다른 분야의 분석 프로젝트도 많이 진행해 본다면 좋겠다고 생각하였습니다.