

R USER CONFERENCE IN KOREA 2019 GO TO COMMUNITY!

📅 10월 25일 (금) 📍 한국마이크로소프트(더케이타워 11층)

한국R사용자커뮤니티(R Korea)

R을 이용한 금융 데이터 수집 및 포트폴리오 만들기

이현열 (Henry Lee)

Profile

□ EDUCATION

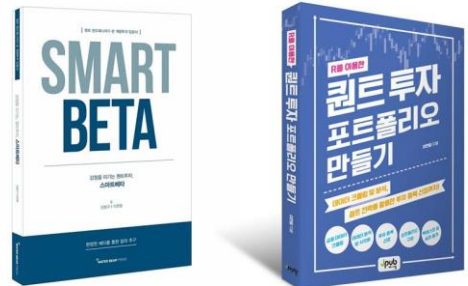
박사과정, 한양대학교 재무금융
석사, KAIST 금융공학
학사, 한양대학교 경영학부

□ Job Experience

메리츠화재, 데이터분석 (2019 ~ 현재)
NH-Amundi 자산운용, 패시브솔루션본부, 퀀트 매니저 (2016 ~ 2019)
한국투자증권, 고객자산운용부, 주식 매니저 (2014~2016)

□ Others

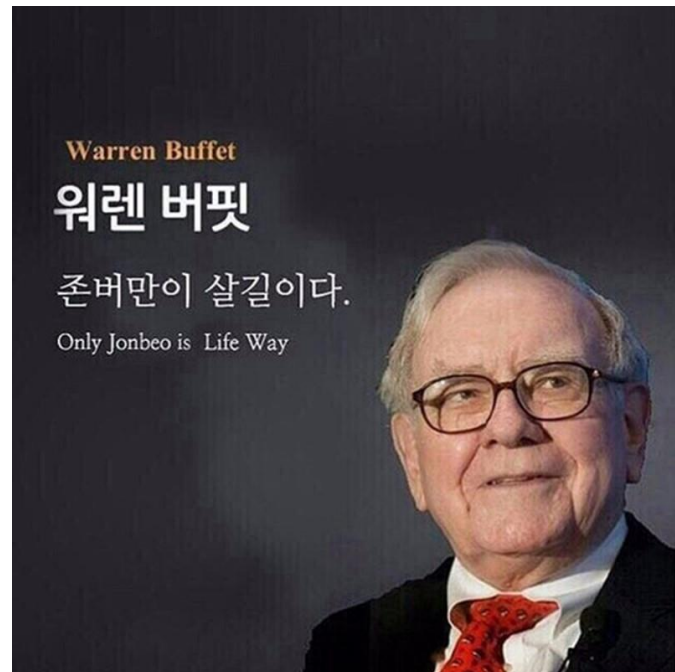
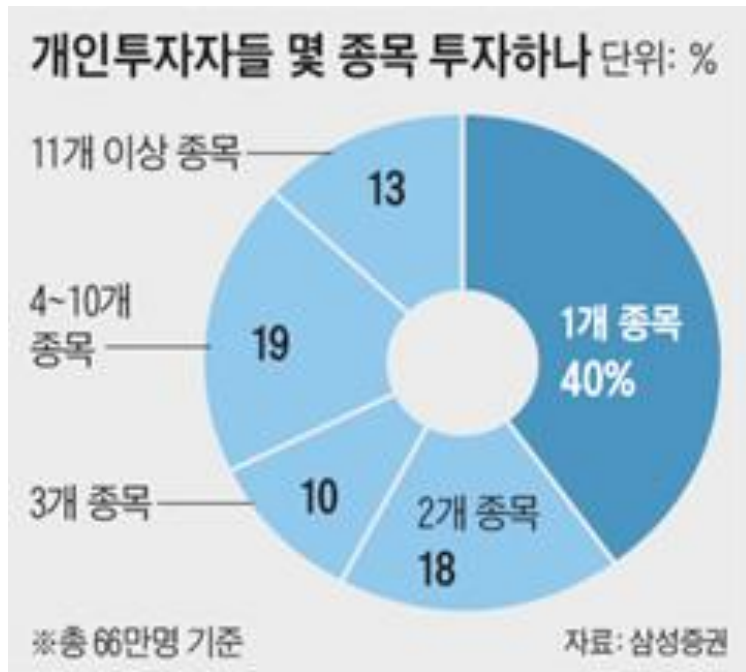
스마트베타 (위터베어프레스, 2017), R을 이용한 퀀트 투자 포트폴리오 만들기 (제이펍, 2019)
패스트캠퍼스, 퀀트투자 과정
Henry's Quantopia (<http://henryquant.blogspot.kr>)
<https://github.com/hyunyulhenry>
leebisu@gmail.com



Wanna be



Reality



Reality



Portfolio



Factor

□ 사전적 정의

요인, 인자

사물이나 사건이 성립되는 까닭. 또는 조건이 되는 요소. ‘원인’, ‘이유’

□ In Finance

주식의 수익률에 영향을 미치는 요소 혹은 지표

팩터의 강도가 양인 종목들로 구성된 포트폴리오는 향후 수익률이 높을 것으로 예상

팩터의 강도가 음인 종목들로 구성된 포트폴리오는 향후 수익률이 낮을 것으로 예상

Factor

THERE ARE FIVE COMMON FACTORS THAT RESEARCH HAS SHOWN TO HAVE THE POTENTIAL TO DELIVER HIGHER RETURNS THAN THE MARKET:



VALUE

Excess returns of stocks that are inexpensive relative to their fundamental value



MOMENTUM

Excess returns of stocks with strong recent performance due to trend following



QUALITY

Excess returns of stocks underpinned by a financially healthy business



LOW VOLATILITY

Excess returns of stocks with lower volatility



SIZE

Excess returns of smaller companies relative to larger counterparts

Factor

Portfolio Cumulative Return

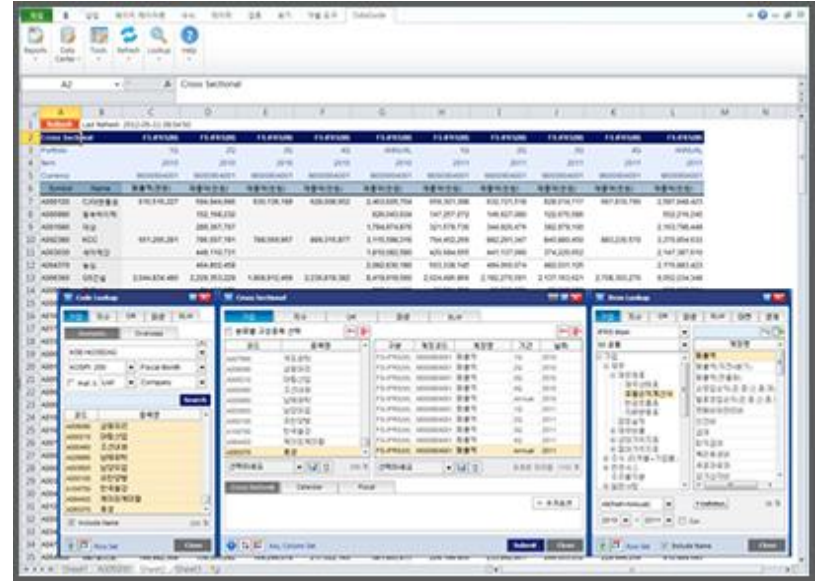


Factor

From 1987.08 ~ 2019.08			
	US Market	고 PBR, 저 ROE	저 PBR, 고 ROE
누적수익률	6.26	2.79	60.74
연율화 수익률	6.38%	4.25%	13.71%
연 변동성	15.14%	28.59%	29.73%
샤프지수	0.42	0.14	0.46

자료 출처: French Library(https://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html)

Data



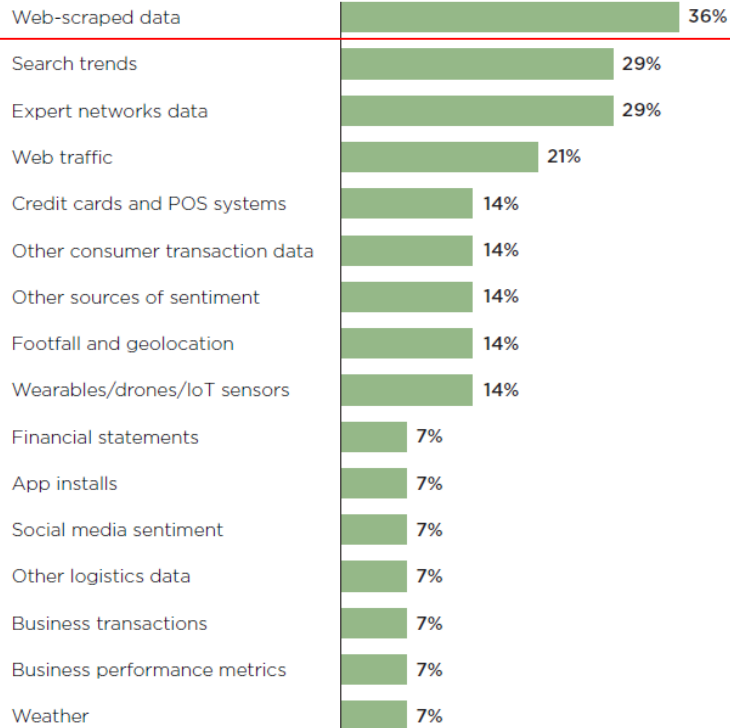
개인투자자에게는 부담되는 사용료

크롤링을 통해 데이터 수집 가능!



Data

ALTERNATIVE DATA SETS CURRENTLY USED



Note: Based on 14 respondents.

Source: Greenwich Associates 2018 Future of Investment Research Study

Packages

httr: <https://cran.r-project.org/web/packages/httr/httr.pdf>

rvest: <https://cran.r-project.org/web/packages/rvest/rvest.pdf>

jsonlite: <https://arxiv.org/pdf/1403.2805.pdf>

Rselenium: <https://cran.r-project.org/web/packages/RSelenium/RSelenium.pdf>

1. Ticker (주식 코드)

- 국내 혹은 해외 상장된 전 종목의 티커 수집
- 수집된 티커들을 토대로 for loop를 통해 전종목의 주가, 재무제표 데이터, 가치 지표 데이터 등을 수집

1. Ticker (주식 코드)

CompanyGuide

기업정보 | ETF/ETN 정보 | 리포트 | 실적속보 | 컨센서스 스크리닝 | 랭킹 분석 | 캘린더 | 부가정보 | 이용안내

Snapshot | 기업개요 | 재무제표 | 재무비율 | 투자지표 | 컨센서스 | 자본분석 | 업종분석 | 경쟁사비교 | 거래소공시 | 금감원공시

삼성전자 005930 | PER 8.28 | 12M PER 12.27 | 업종 PER 6.58 | PBR 1.41 | 배당수익률 2.84%

KSE 코스피 전기,전자 | FICS 휴대폰 및 관련부품 | 12월 결산

3개월 | 1년 | 3년 | SELL BUY | 외국인 보유비중 | 시가총액 | 삼성전자 | 코스피 전기,전자 | KOSPI

주가수익, 내부지거래 **49,900** | 외국인 보유비중, 시가총액 **57.51** | 상대수익률 **13.28**

시세현황 [2019/10/18] 단위: 원, 주, %

종가/전일대비	49,900 / -600	거래량	8,483,106
52주, 최고가/ 최저가	50,700 / 37,450	거래대금 (억원)	4,269
수익률 (1M/ 3M/ 6M/ 1Y)	+4.61 / +8.24 / +9.43 / +13.28	외국인 보유비중	57.51
시가총액 (상장예정포함, 억원)	3,310,545	베타	1.31629
시가총액 (보통주, 억원)	2,978,921	역면가	100
발행주식수 (보통주/ 우선주)	5,969,782,550 / 822,886,700	유동주식수/비율 (보통주)	4,449,224,951 / 74.53

실적이슈 [2019/09 기준, 분기]

경영성과표현	경영성적 (영업이익, 억원)	매상성적대비 (%)	전년도기대비 (%)
2019/10/0R	77,000	+8.32	-56.10

Elements | Console | Sources | Network | Performance | Memory | Application | Security

Filter | Hide data URLs | XHR | JS | CSS | Img | Media | Font | Doc | WS | Manifest | Other

Name | Headers | Preview | Response | Timing

General

Request URL: <http://comp.fnguide.com/XML/Market/CompanyList.txt>

Request Method: GET

Status Code: 200 OK (from disk cache)

Remote Address: 61.111.22.238:80

Referrer Policy: no-referrer-when-downgrade

Response Headers (10)

Request Headers (4)

49 requests | 34.6 KB transferred | 851 KB resources | Finish: 4

Console | What's New

Highlights from the Chrome 77 update

<http://comp.fnguide.com> → 개발자도구 네트워크 → CompanyList.txt → <http://comp.fnguide.com/XML/Market/CompanyList.txt>

1. Ticker (주식 코드)

```
url = 'http://comp.fnguide.com/XML/Market/CompanyList.txt'  
ticker = fromJSON(url) %>%  
  .[[1]]  
  
head(ticker)
```

```
##           cd           nm gb  
## 1 A060310           3S 701  
## 2 A095570 AJ네트웍스 701  
## 3 A068400 AJ렌터카 701  
## 4 A006840 AK홀딩스 701  
## 5 A054620 APS홀딩스 701  
## 6 A265520 AP시스템 701
```

```
tail(ticker)
```

```
##           cd           nm gb  
## 3097 Q500039           신한 코스피 콘도르 4/10% ETN 780  
## 3098 Q500043 신한 코스피 콘도르 4/10% 콜 2204-01 ETN 780  
## 3099 Q500042           신한 코스피 콘도르 6/10% ETN 780  
## 3100 Q500044 신한 코스피 콘도르 6/10% 콜 2204-01 ETN 780  
## 3101 Q500023           신한 콩 선물 ETN(H) 780  
## 3102 Q700001 하나 코스피 변동성추세 추종 양매도 ETN 780
```

1. Ticker (주식 코드)

The image shows a screenshot of the KRX Marketdata website and a browser's network tab. The website displays the '30015 시가총액 상/하위' (Market Capitalization Top/Bottom 30015) page. The page includes a search bar, filters for market section (전체, KOSPI, KOSDAQ, KONEX) and industry (전체), and a search button. Below the search bar, there is a table of stock data with columns for rank, stock code, name, and various price metrics. The table lists 10 stocks, including Samsung Electronics, SK Hynix, Samsung Electronics Co., Ltd., Hyundai, NAVER, SK Hynix, Hyundai Motor, Samsung Biopharmaceuticals, LG Chemical, and Shinhan Financial Group.

순위	종목코드	종목명	시가	고가	저가	종가	대비
1	005930	삼성전자	50,300	50,900	49,650	49,900	▼ 60
2	000660	SK하이닉스	81,500	81,700	77,300	77,400	▼ 360
3	005935	삼성전자우	40,700	41,000	40,200	40,300	▼ 60
4	005380	현대차	123,500	124,500	122,500	123,000	-
5	035420	NAVER	151,500	153,500	150,000	150,500	▼ 250
6	068270	셀트리온	188,000	188,500	181,000	181,500	▼ 650
7	012330	현대모비스	243,500	243,500	238,500	241,000	▲ 50
8	207940	삼성바이오...	345,000	349,500	342,000	344,000	▼ 1,00
9	051910	LG화학	301,500	305,000	295,500	295,500	▼ 850
10	055550	신한지주	42,550	42,700	42,000	42,100	▼ 60

The browser's network tab shows a list of requests. The first request is 'GenerateOTP.jsp?name=fileDown&filetype=...' with a GET method, status 200, and initiator 'jquery.min.js'. The second request is 'download.jsp' with a POST method, status 200, and initiator 'jquery.min.js'.

<http://marketdata.krx.co.kr/mdr#document=040402> (시장정보 → 순위정보 → 시가총액 상/하위) → 개발자도구 네트워크
→ CSV 눌러 통신과정 추출

1. Ticker (주식 코드)

```
gen_otp_url =
    'http://marketdata.krx.co.kr/contents/COM/GenerateOTP.jspx'

gen_otp_data = list(
    name = 'fileDown',
    filetype = 'csv',
    url = 'MKD/04/0404/04040200/mkd04040200_01',
    market_gubun = 'ALL',
    schdate = '20191018',
    pagePath = '/contents/MKD/04/0404/04040200/MKD04040200.jsp')

otp = POST(gen_otp_url, query = gen_otp_data) %>%
  read_html() %>%
  html_text()

print(otp)
```

```
## [1] "kR4BVNFag+szu8hB3hhNymgJsgeQ7X0ZaXc6vyVk/+Qs03cF1vuuv+hdyLRUziuDyRMX/HBpjAkirb2nHJcS8CxUFLKh8tQ5MDXVvJGdp
pCVSeexeUnQLKlztlviiHyrIJYmWPa7u0mENV6pZAcUCXM8JEZiIvSb6rvdvdB1S1s7PVstggWxKbRzLV/8odehpTlxi+YEplKfC8WbSJakwClE0I
gGU6ItqrueuA8KSnKrZxeVnliSxzSkgLQ+d+rHk4tspaOwQJJ0P0AyMQUQSfxc/Lg7cwy89YICseH6sqo="
```

1. Ticker (주식 코드)

```
down_url = 'http://file.krx.co.kr/download.jspx'

KOR_ticker = POST(down_url, query = list(code = otp),
                  add_headers(referer = gen_otp_url)) %>%
  read_html() %>%
  html_text() %>%
  readr::read_csv()

head(KOR_ticker)
```

```
## # A tibble: 6 x 17
##   순위 종목코드 종목명 현재가 대비 등락률 거래량 거래대금 시가 고가
##   <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     1 005930 삼성전자~ 49900 -600 -1.2 8.48e6 4.27e11 50300 50900
## 2     2 000660 SK하이닉~ 77400 -3600 -4.4 1.36e7 1.06e12 81500 81700
## 3     3 005935 삼성전자우~ 40300 -600 -1.5 1.79e6 7.26e10 40700 41000
## 4     4 005380 현대차 123000 0 0 4.18e5 5.16e10 123500 124500
## 5     5 035420 NAVER 150500 -2500 -1.6 2.46e5 3.73e10 151500 153500
## 6     6 068270 셀트리온~ 181500 -6500 -3.5 4.35e5 8.00e10 188000 188500
## # ... with 7 more variables: 저가 <dbl>, 시가총액 <dbl>,
## # `시가총액비중(%)` <dbl>, 상장주식수 <dbl>, `외국인 보유주식수` <lgl>,
## # `외국인 지분율(%)` <lgl>, 총카운트 <dbl>
```

2. Price (주가)

- ❑ quantmod 패키지의 getSymbols() 함수를 이용
- ❑ 크롤링을 통해 주가 수집

2. Price (주가)

```
library(quantmod)
```

```
getSymbols('005930.KS')
```

```
## [1] "005930.KS"
```

```
str(`005930.KS`)
```

```
## An 'xts' object on 2007-01-02/2019-10-18 containing:  
## Data: num [1:3171, 1:6] 12400 12540 12220 12160 11840 ...  
## - attr(*, "dimnames")=List of 2  
## ..$ : NULL  
## ..$ : chr [1:6] "005930.KS.Open" "005930.KS.High" "005930.KS.Low" "005930.KS.Close" ...  
## Indexed by objects of class: [Date] TZ: UTC  
## xts Attributes:  
## List of 2  
## $ src : chr "yahoo"  
## $ updated: POSIXct[1:1], format: "2019-10-20 13:27:18"
```

```
tail(`005930.KS`)
```

```
##           005930.KS.Open 005930.KS.High 005930.KS.Low 005930.KS.Close  
## 2019-10-11           49000           49450           48800           49150  
## 2019-10-14           50000           50300           49850           50000  
## 2019-10-15           49900           50200           49900           50100  
## 2019-10-16           50700           50900           50400           50700  
## 2019-10-17           50500           50600           50100           50500  
## 2019-10-18           50300           50900           49650           49900  
##           005930.KS.Volume 005930.KS.Adjusted  
## 2019-10-11           7783275           49150  
## 2019-10-14          11130635           50000  
## 2019-10-15           6058105           50100  
## 2019-10-16           9136282           50700  
## 2019-10-17           6701380           50500  
## 2019-10-18           8483106           49900
```

2. Price (주가)

getSymbols() 함수

- 야후 파이낸스 API를 이용하여 손쉽게 주가 다운로드

해당 함수의 문제점

- 국내 대형주 데이터는 존재하나 중,소형주 정보는 없는 경우가 존재
- 수정주가 관리가 제대로 되지않음

2. Price (주가)

30040 일자별시세

시장정보 > 주식 > 종목정보

종목검색: A005930/삼성전자

조회날짜: 20191011 ~ 20191018

기간: 1일 1개월 6개월 1년

조회

차트 | Excel | CSV

삼성전자[A005930] 유가증권시장 / 통신 및 방송 장비 제조업

2019/10/20 PM 01:32:42 (20분 지연 정보)

현재가	전일비	등락률(%)	시가	고가	저가	거래량(주)	거래대금(원)
49,900	▼ 600	-1.19	50,300	50,900	49,650	4,660,056	235,766,136,800

년/월/일	종가	대비	거래량(주)	거래대금(원)	시가	고가	저가
2019/10/18	49,900	▼ 600	8,483,106	426,947,644,700	50,300	50,900	49,650
2019/10/17	50,500	▼ 200	6,701,380	337,530,648,575	50,500	50,600	50,100
2019/10/16	50,700	▲ 600	9,136,282	462,745,898,500	50,700	50,900	50,400
2019/10/15	50,100	▲ 100	6,058,105	303,431,646,175	49,900	50,200	49,900
2019/10/14	50,000	▲ 850	11,130,635	557,617,555,504	50,000	50,300	49,850
2019/10/11	49,150	▲ 600	7,783,275	382,773,784,700	49,000	49,450	48,800

Filter: Hide data URLs | XHR JS CSS Img Media Font Doc WS Manifest Other

50 ms 100 ms 150 ms 200 ms 250 ms 300 ms

Name: GenerateOTP.jspx?name=fileDown&filetype=csv&url=MK...F... download.jspx

Headers Preview Response Cookies Timing

Request Method: GET

Status Code: 200 OK

Remote Address: 115.22.33.34:80

Referrer Policy: no-referrer-when-downgrade

Response Headers (3)

Request Headers (9)

Query String Parameters (view source view URL encoded)

name: fileDown
filetype: csv
url: MKD/04/0402/04020100/mkd04020100t3_02
isu_cdnm: A005930/삼성전자
isu_cd: KR7005930003
isu_nm: 삼성전자
isu_srt_cd: A005930
fromdate: 20191011
todate: 20191018
pagePath: /contents/MKD/04/0402/04020100/MKD04020100T3T2.jsp

2 requests | 757 B transferred | 472 B resources

<http://marketdata.krx.co.kr/mdid#document=040204> (시장정보 → 종목정보 → 일자별시세) → 개발자도구 네트워크 → CSV 눌러 통신과정 추출

2. Price (추가)

```
gen_otp_url =
  'http://marketdata.krx.co.kr/contents/COM/GenerateOTP.jspx'

gen_otp_data = list(
  name = 'fileDown',
  filetype = 'csv',
  url = 'MKD/04/0402/04020100/mkd04020100t3_02',
  isu_cdnm = 'A005930/삼성전자',
  isu_cd = 'KR7005930003',
  isu_nm = '삼성전자',
  isu_srt_cd = 'A005930',
  fromdate = '20150102',
  todate = '20191018',
  pagePath = '/contents/MKD/04/0402/04020100/MKD04020100T3T2.jsp'
)

otp = POST(gen_otp_url, query = gen_otp_data) %>%
  read_html() %>%
  html_text()

down_url = 'http://file.krx.co.kr/download.jspx'

price_samsung = POST(down_url, query = list(code = otp),
  add_headers(referer = gen_otp_url)) %>%
  read_html() %>%
  html_text() %>%
  readr::read_csv()

head(price_samsung)
```

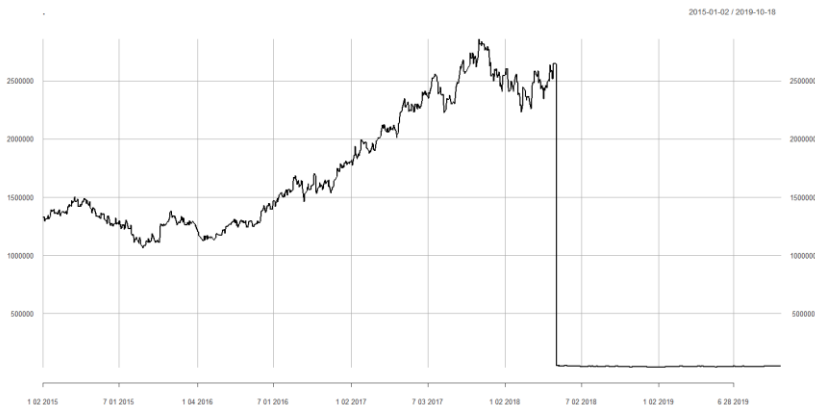
```
## # A tibble: 6 x 10
##   `년/월/일`   `증가 대비 `거래량(주)` `거래대금(원)`   `시가`   `고가`   `저가`
##   <date>       <dbl> <dbl>         <dbl>         <dbl> <dbl> <dbl>
## 1 2019-10-18  49900  -600         8483106       426947644700 50300 50900 49650
## 2 2019-10-17  50500  -200         6701380       337530648575 50500 50600 50100
## 3 2019-10-16  50700   600         9136282       462745898500 50700 50900 50400
## 4 2019-10-15  50100  100         6058105       303431646175 49900 50200 49900
## 5 2019-10-14  50000  850         11130635      557617555504 50000 50300 49850
## 6 2019-10-11  49150  600         7783275       382773784700 49000 49450 48800
## # ... with 2 more variables: `시가총액(백만)` <dbl>,
## #   `상장주식수(주)` <dbl>
```

2. Price (주가)

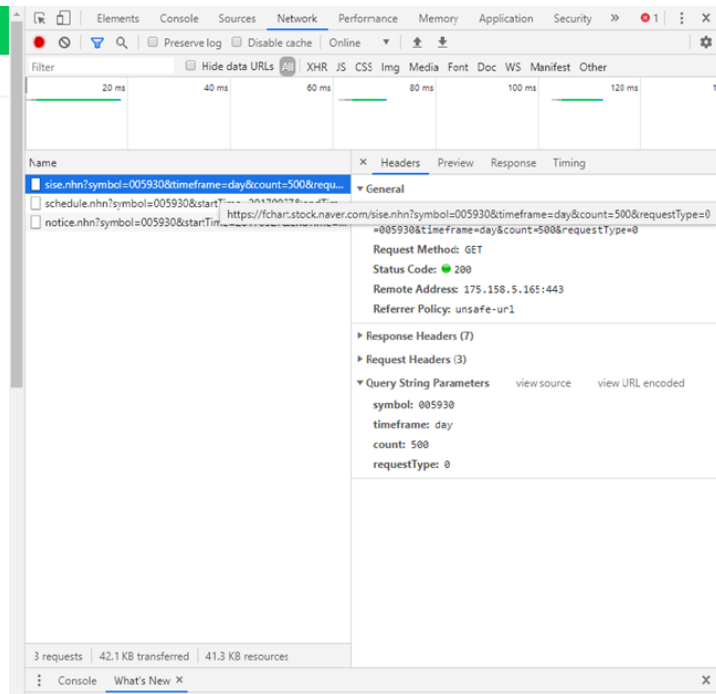
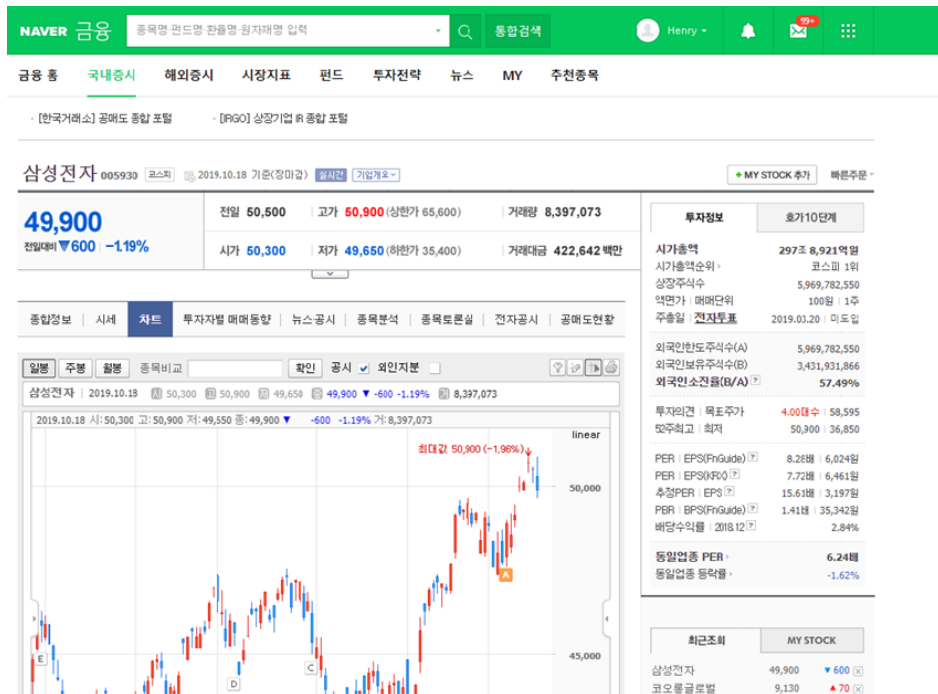
fromdate를 원하는 일자로 입력하면, 상장 이후부터 주가 다운로드 가능

해당 방법의 문제점

- ❑ 쿼리 항목이 많음
- ❑ 전 종목 주가를 받기에는 속도가 느림
- ❑ 수정주가 미제공



2. Price (주가)



<https://finance.naver.com/item/fchart.nhn?code=005930> (네이버 금융) → 차트 → 개발자도구 네트워크 → [일봉]클릭
→ sise.nhn?symbol= → <https://fchart.stock.naver.com/sise.nhn?symbol=005930&timeframe=day&count=500&requestType=0>

2. Price (주가)

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<protocol>
  ▼<chartdata symbol="005930" name="삼성전자" count="500" timeframe="day" precision="0" origintime="19900103">
    <item data="20171012|54840|55160|54099|54800|277814"/>
    <item data="20171013|54540|54840|53780|54000|252033"/>
    <item data="20171016|53980|54860|53760|53920|195399"/>
    <item data="20171017|54019|55380|54000|54800|212156"/>
    <item data="20171018|54820|55240|54039|54760|202215"/>
    <item data="20171019|54699|54699|52980|52980|242174"/>
    <item data="20171020|52800|54100|52800|53840|160541"/>
    <item data="20171023|54600|54640|54000|54300|166221"/>
    <item data="20171024|54699|54780|54040|54040|116121"/>
    <item data="20171025|54040|54420|53699|53900|117657"/>
    <item data="20171026|53720|53900|52400|52400|195531"/>
    <item data="20171027|52400|53320|52140|53080|147850"/>
    <item data="20171030|53780|54320|53699|54040|163923"/>
    <item data="20171031|54060|55440|53500|55080|267631"/>
    <item data="20171101|57500|57500|56179|57220|290205"/>
    <item data="20171102|57500|57519|56760|57060|214421"/>
    <item data="20171103|57059|57140|55860|56380|198716"/>
    <item data="20171106|56380|56500|55340|56380|180019"/>
    <item data="20171107|56380|56920|55860|56100|149291"/>
    <item data="20171108|56100|57060|55860|56760|170404"/>
    <item data="20171109|56920|56920|55899|56340|239881"/>
    <item data="20171110|55800|56540|55780|56400|134976"/>
    <item data="20171113|56400|56799|56100|56380|176133"/>
    <item data="20171114|56380|56740|55920|55920|134500"/>
    <item data="20171115|55920|56319|55320|55340|176339"/>
    <item data="20171116|55440|56000|55400|55780|152814"/>
    <item data="20171117|56399|56880|55820|55820|214663"/>
    <item data="20171120|55900|55980|55200|55200|189455"/>
    <item data="20171121|55400|55839|55280|55280|206494"/>
    <item data="20171122|55980|56200|55620|55960|159059"/>
    <item data="20171123|55980|55980|55980|55980|198459"/>
  </chartdata>
</protocol>
```

2. Price (주가)

```
url = 'https://fchart.stock.naver.com/sise.nhn?symbol=005930&timeframe=day&count=500&requestType=0'

data = GET(url)
data_html = read_html(data, encoding = 'EUC-KR') %>%
  html_nodes('item') %>%
  html_attr('data')

price = readr::read_delim(data_html, delim = '|', col_names = FALSE)
head(price)
```

```
## # A tibble: 6 x 6
##       X1      X2      X3      X4      X5      X6
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 20170927 52000 52200 51500 51680 215940
## 2 20170928 52260 52460 51260 51260 240509
## 3 20170929 51180 51620 50840 51280 258188
## 4 20171010 53360 53640 52799 52800 410396
## 5 20171011 53600 54760 53340 54640 273043
## 6 20171012 54840 55160 54099 54800 277814
```

2. Price (주가)

```
price = price %>%  
  select(1,5) %>%  
  data.frame() %>%  
  magrittr::set_colnames(c('Date', 'Price')) %>%  
  mutate(Date = lubridate::ymd(Date)) %>%  
  timetk::tk_xts(., date_var = Date)  
  
head(price)
```

```
##           Price  
## 2017-09-27 51680  
## 2017-09-28 51260  
## 2017-09-29 51280  
## 2017-10-10 52800  
## 2017-10-11 54640  
## 2017-10-12 54800
```

3. FS (재무제표)



공시서류검색



- 통합검색 >
- 상세검색 >
- 회사별검색 >**
- 정정보고서검색 >
- 펀드공시상세검색 >
- 신용평가공시상세검색 >
- 비교검색 >
- 첨부서류검색 >

검색

로그인 | 마이페이지 | 공시업무 | DART소개 | 오픈API | RSS | 사이트맵

[최근공시](#)
[공시서류검색](#)
[공시정보활용마당](#)
[기업개황](#)
[공모게시판](#)
[최근정정보고서](#)
[최근삭제제도](#)

회사별검색

최종보고서
 기간: 19990101 - 20191020 (1주일 | 1개월 | 6개월 | 1년 | 2년 | 3년 | 전체)

정기공시
 주요사항보고
 발행공시
 지분공시
 기타공시
 외부감사관련
 펀드공시
 자산유동화
 거래소공시
 공정위공시

장기공시 > 전체선택

사업보고서
 반기보고서
 분기보고서
 소액공모법인결산서류
 등록법인결산서류(자본시장법 미전)

조회건수 15

번호	공시대상회사	보고서명	제출인	접수일자	비고
1	삼성전자	사업보고서 (2018.12)	삼성전자	2019.04.01	연
2	삼성전자	사업보고서 (2017.12)	삼성전자	2018.04.02	연
3	삼성전자	사업보고서 (2016.12)	삼성전자	2017.03.31	연
4	삼성전자	사업보고서 (2015.12)	삼성전자	2016.03.30	연
5	삼성전자	사업보고서 (2014.12)	삼성전자	2015.03.31	연
6	삼성전자	사업보고서 (2013.12)	삼성전자	2014.03.31	연
7	삼성전자	사업보고서 (2012.12)	삼성전자	2013.04.01	연
8	삼성전자	사업보고서 (2011.12)	삼성전자	2012.03.30	연
9	삼성전자	사업보고서 (2010.12)	삼성전자	2011.03.31	연

주의요항 | dart.fss.or.kr/dsaF001/main.do?rcpNo=20190401004781
 본문: 2019.04.01 사업보고서
 첨부: 1첨부분석*

문서목록

- 사업보고서
- 【대표이사 등의 확인】
- I. 회사의 개요
 - 1. 회사의 개요
 - 2. 회사의 연혁
 - 3. 자본금 변동사항
 - 4. 주식의 총수 등
 - 5.役員권 현황
 - 6. 배당에 관한 사항 등
- II. 사업의 내용
 - III. 재무에 관한 사항
 - 1. 요약재무정보
 - 2. 연결재무제표
 - 3. 연결재무제표 주석
 - 4. 재무제표
 - 5. 재무제표 주석
 - 6. 기타 재무에 관한 사항
 - IV. 이사의 경영진단 및 분석의
 - V. 감사인의 감사의견 등
 - VI. 이사회 등 회사의 기관에 관
 - 1. 이사회에 관한 사항
 - 2. 감사제도에 관한 사항
 - 3. 주주의 役員권 행사에 관
 - VII. 주주에 관한 사항
 - VIII. 임원 및 役員 등에 관한 사
 - 1. 임원 및 役員의 현황
 - 2. 임원의 보수 등
 - IX. 계열회사 등에 관한 사항
 - X. 이해관계자와의 거래내용
 - 지. 그 밖에 투자자 보호를 위하
- 【전문가의 확인】
 - 1. 전문가의 확인
 - 2. 전문가와의 이해관계

2. 연결재무제표

연결 재무상태표

제 50 기	2018.12.31 현재
제 49 기	2017.12.31 현재
제 48 기	2016.12.31 현재

(단위 : 백만원)

	제 50 기	제 49 기	제 48 기
자산			
유동자산	174,697,424	146,982,464	141,429,704
현금및현금성자산	30,340,505	30,545,130	32,111,442
단기금융상품	65,693,797	49,447,696	52,432,411
단기매도가능금융자산		3,191,375	3,638,460
단기상각후원가금융자산	2,703,693		
단기담기손익-공정가치금융자산	2,001,948		
매출채권	33,867,733	27,695,995	24,279,211
미수금	3,080,733	4,108,961	3,521,197
선급금	1,361,807	1,753,673	1,439,938
선급비용	4,136,167	3,835,219	3,502,083
채고자산	28,984,704	24,983,355	18,353,503
기타유동자산	2,326,337	1,421,060	1,315,653
매각예정무류자산			835,806
비유동자산	164,659,820	154,769,626	120,744,620
장기매도가능금융자산		7,752,180	6,804,276
만기보유금융자산		106,751	

3. FS (재무제표)

검색 API

기업개황 API

개발가이드 다운로드

요청 주소

http://dart.fss.or.kr/api/search.xml?auth=xxx xml 응답
 http://dart.fss.or.kr/api/search.json?auth=xxx json 응답

요청 변수

요청변수	설명
auth	발급받은 인증키(40자리)(필수)
crp_cd	공시대상회사의 종목코드(상장사:숫자 6자리) 또는 고유번호(기타법인:숫자 8자리)
end_dt	검색종료 접수일자(YYYYMMDD) : 없으면 당일
start_dt	검색시작 접수일자(YYYYMMDD) : 없으면 end_dt crp_cd가 없는 경우 검색기간은 3개월로 제한
fin_rpt	최종보고서만 검색여부(Y or N) 기본값 : N (정정이 있는 경우 최종정정만 검색)
dsp_tp	정기공시(A), 주요사항보고(B), 발행공시(C), 지분공시(D), 기타공시(E), 외부감사관련(F), 펀드공시(G), 자산유통화(H), 거래소공시(I), 공정위공시(J)
bsn_tp	정기공시(5), 주요사항보고(3), 발행공시(11), 지분공시(4), 기타공시(9), 외부감사관련(5), 펀드공시(3), 자산유통화(6), 거래소공시(6), 공정위공시(5) (상세 유형 참조)
sort	접수일자(date), 회사명(crp), 보고서명(rpt) 기본값 : date
series	오름차순(asc), 내림차순(desc) 기본값 : desc
page_no	페이지 번호(1~n) 기본값 : 1
page_set	페이지당 건수(1~100) 기본값 : 10, 최대값 : 100
callback	콜백함수명(JSONP용)

3. FS (재무제표)

```
library(jsonlite)

url =
  paste0("http://dart.fss.or.kr/api/search.json?",
        "auth=", API_key, "&rcrp_cd=005930&start_dt=19990101&bsn_tp=A001")

dart_data = fromJSON(url)

dart_rcrp_no = dart_data$list$rcrp_no

report_url =
  paste0("http://dart.fss.or.kr/dsaf001/main.do?rcrpNo=", dart_rcrp_no[1])

report_data = GET(report_url) %>%
  read_html() %>%
  html_node(xpath = '//*[id="north"]/div[2]/ul/li[1]/a')

dcm_no = report_data %>%
  html_attr('onclick') %>%
  stringr::str_match_all('[0-9]+') %>%
  unlist() %>%
  tail(1)

excel_data =
  POST("http://dart.fss.or.kr/pdf/download/excel.do",
       query = list(
         rcrp_no = '20190401004781',
         dcm_no = dcm_no,
         lang = 'ko'
       ))

writeBin(content(excel_data, 'raw'),
        'sample.xls')

readxl::read_excel('sample.xls', sheet = 2)
```

```
## # A tibble: 67 x 4
##   `연결 재무상태표`      ...2      ...3      ...4
##   <chr>                <chr>    <chr>    <chr>
## 1 제 50 기 2018.12.31 현재 <NA>    <NA>    <NA>
## 2 제 49 기 2017.12.31 현재 <NA>    <NA>    <NA>
## 3 제 48 기 2016.12.31 현재 <NA>    <NA>    <NA>
## 4 (단위 : 백만원)      <NA>    <NA>    <NA>
## 5 <NA>                  제 50 기 제 49 기 제 48 기
## 6 자산                  <NA>    <NA>    <NA>
## 7 유동자산              174697424 146982464 141429704
## 8 현금및현금성자산      30340505 30545130 32111442
## 9 단기금융상품          65893797 49447696 52432411
## 10 단기매도가능금융자산 <NA>    3191375 3638460
## # ... with 57 more rows
```

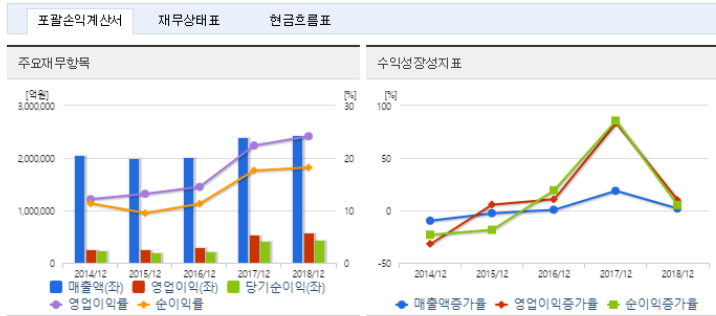
3. FS (재무제표)

<http://henryquant.blogspot.com/2019/02/r-dart-api.html>

문제점

- ❑ 전종목의 다년간 재무제표 크롤링에 시간이 오래 소모됨
- ❑ 클렌징이 어려움
- ❑ 엑셀이 아닌 PDF로 재무제표를 제공하는 기업 존재

3. FS (재무제표)



* 단위 : 억원, %, 배, 천주 * 분기 : 순액기준

항목	2014/12 (IFRS연결)	2015/12 (IFRS연결)	2016/12 (IFRS연결)	2017/12 (IFRS연결)	2018/12 (IFRS연결)	전년대비 (YoY)
▶ 매출액(수익)	2,062,059.9	2,006,534.8	2,018,667.5	2,395,753.8	2,437,714.2	1.8
*내수	207,289.2	208,278.2				
*수출	1,854,770.7	1,798,256.6				
▶ 매출원가	1,282,788.0	1,234,821.2	1,202,777.2	1,292,906.6	1,323,944.1	2.4
매출총이익	779,271.9	771,713.6	815,890.3	1,102,847.2	1,113,770.0	1.0
▶ 판매비외관리비	529,021.2	507,579.2	523,483.6	566,396.8	524,903.4	-7.3
영업이익	250,250.7	264,134.4	292,406.7	536,450.4	588,866.7	9.8
▶ 기타영업손익						
영업이익(발표기준)				536,450.4	588,866.7	9.8
* [K-IFRS]영업이익						
▶ 금융수익	82,598.3	105,148.8	113,856.5	97,373.9	99,993.2	2.7
▶ 금융원가	72,940.0	100,317.7	107,066.1	89,789.1	86,089.0	-4.1
▶ 기타영업외손익	15,416.2	-20,374.9	7,744.5	15,910.1	3,430.2	-78.4
▶ 종속기업, 공동지배기업및관...	3,425.2	11,019.3	195.0	2,014.4	5,398.5	168.0
법인세비용차감전 계속사업...	278,750.3	259,610.0	307,136.5	561,959.7	611,599.6	8.8

□ 네이버 금융의 재무제표 항목

(<https://finance.naver.com/item/coinfo.nhn?code=005930>)

□ 동적 웹페이지로 구성되어 있어 셀레니움 필요 → 번거로움

3. FS (재무제표)

IFRS(연결)	2016/12	2017/12	2018/12	2019/06	전년 동기	전년 동기(%)
매출액	2,018,667	2,395,754	2,437,714	1,085,127	1,190,464	-8.4
매출세금	1,202,777	1,292,907	1,323,944	686,911	631,841	8.7
매출회차	815,890	1,102,847	1,113,770	398,215	558,623	-28.3
매출회차(조정)	523,484	566,397	524,903	269,912	253,511	6.4
영업이익	292,407	536,450	588,867	128,303	305,112	-57.5
영업비용	113,856	97,374	99,992	50,681	54,022	-6.4
영업비용(조정)	107,066	99,799	86,089	40,602	47,732	-14.5
기타비용	32,383	30,107	14,850	6,819	7,386	-7.7
기타비용(조정)	24,638	14,196	11,420	5,700	4,672	-21.5
영업이익(조정)	195	2,014	5,398	1,422	1,797	-18.6
영업비용(조정)이익	307,137	561,960	611,600	140,923	315,827	-55.4
영업비용(조정)이익	79,676	140,092	168,151	38,682	88,508	-56.3
영업비용(조정)이익(조정)	227,261	421,867	443,449	102,242	227,320	-55.4
영업비용(조정)이익(조정)	227,261	421,867	443,449	102,242	227,320	-55.4
비지속이익(손실)	3,104	8,422	4,540	521	1,386	-62.4
정주성이익(손실)						
주요 정주성이익						
연결 순이익						
자산	2,621,743	3,017,521	3,017,521	3,393,572	3,429,401	
유동자산	1,414,297	1,469,825	1,469,825	1,746,974	1,734,331	
비유동자산	1,207,446	1,547,696	1,547,696	1,646,598	1,695,069	
부채	692,113	872,607	872,607	916,041	851,931	
유동부채	547,041	671,751	671,751	690,815	593,095	
비유동부채	145,072	200,855	200,855	225,226	258,836	
자본	1,929,630	2,144,914	2,144,914	2,477,532	2,577,471	
자본(조정)이익	1,864,243	2,072,134	2,072,134	2,400,690	2,498,927	
비지속이익(손실)	65,387	72,780	72,780	76,842	78,544	
연결 자본						
Invested Capital						
Free Cash Flow						
IFRS(연결)	2016/12	2017/12	2018/12	2019/06		
영업활동으로 인한 현금흐름		473,856	621,620	670,319	118,395	
투자활동		227,351	421,867	443,449	102,242	
금융활동		367,817	413,191	484,356	206,841	
영업활동으로 인한 현금흐름(조정)		60,272	51,079	48,210	26,872	
영업활동으로 인한 현금흐름(조정)		-11,810	-106,205	-99,244	-77,672	
영업활동으로 인한 현금흐름(조정)		522,996	677,774	780,251	204,539	
기타 영업활동으로 인한 현금흐름		-69,140	-56,154	-109,932	-86,144	
영업활동으로 인한 현금흐름(조정)		-296,587	-493,852	-522,405	-85,284	

http://comp.fnguide.com/SVO2/ASP/SVD_Finance.asp?pGB=1&qicode=A005930&cID=&MenuYn=Y&ReportGB=&NewMenuID=103&stkGb=701 → 불필요한 url 제거 → http://comp.fnguide.com/SVO2/ASP/SVD_Finance.asp?pGB=1&qicode=A005930

3. FS (재무제표)

```
url = paste0('http://comp.fnguide.com/SVO2/ASP/SVD_Finance.asp?pGB=1&gicod=A005930')

Sys.setlocale("LC_ALL", "English")

data = GET(url)
data = data %>%
  read_html() %>%
  html_table()

Sys.setlocale("LC_ALL", "Korean")
```

```
lapply(data, function(x) {
  head(x, 1)})
```

```
## [[1]]
##   IFRS(연결)   2016/12   2017/12   2018/12   2019/06   전년동기   전년동기(%)
## 1   매출액  2,018,667  2,395,754  2,437,714  1,085,127  1,190,464      -8.8
##
## [[2]]
##   IFRS(연결) 2018/09 2018/12 2019/03 2019/06   전년동기   전년동기(%)
## 1   매출액  654,600  592,651  523,855  561,271   584,827      -4
##
## [[3]]
##   IFRS(연결)   2016/12   2017/12   2018/12   2019/06
## 1   자산  2,621,743  3,017,521  3,393,572  3,429,401
##
## [[4]]
##   IFRS(연결)   2018/09   2018/12   2019/03   2019/06
## 1   자산  3,371,958  3,393,572  3,450,679  3,429,401
##
## [[5]]
##           IFRS(연결) 2016/12 2017/12 2018/12 2019/06
## 1 영업활동으로인한현금흐름 473,856 621,620 670,319 118,392
##
## [[6]]
##           IFRS(연결) 2018/09 2018/12 2019/03 2019/06
## 1 영업활동으로인한현금흐름 155,497 224,281 52,443 65,949
```

3. FS (재무제표)

```
data_IS = data[[1]]
data_BS = data[[3]]
data_CF = data[[5]]

data_IS = data_IS[, 1:(ncol(data_IS)-2)]
data_fs = rbind(data_IS, data_BS, data_CF)
data_fs[, 1] = gsub('계산에 참여한 계정 펼치기',
                  '', data_fs[, 1])
data_fs = data_fs[!duplicated(data_fs[, 1]), ]

rownames(data_fs) = NULL
rownames(data_fs) = data_fs[, 1]
data_fs[, 1] = NULL

data_fs = data_fs[, substr(colnames(data_fs), 6,7) == '12']

print(head(data_fs))
```

##	2016/12	2017/12	2018/12
## 매출액	2,018,667	2,395,754	2,437,714
## 매출원가	1,202,777	1,292,907	1,323,944
## 매출총이익	815,890	1,102,847	1,113,770
## 판매비와관리비	523,484	566,397	524,903
## 인건비	59,763	67,972	64,514
## 유무형자산상각비	10,018	13,366	14,477

3. FS (재무제표)

```
data_fs = sapply(data_fs, function(x) {  
  stringr::str_replace_all(x, ',', '') %>%  
  as.numeric()  
}) %>%  
  data.frame(., row.names = rownames(data_fs))  
  
print(head(data_fs))
```

```
##           X2016.12 X2017.12 X2018.12  
## 매출액          2018667  2395754  2437714  
## 매출원가          1202777  1292907  1323944  
## 매출총이익          815890  1102847  1113770  
## 판매비와관리비      523484   566397   524903  
## 인건비              59763    67972    64514  
## 유무형자산상각비    10018    13366    14477
```


4. Value (가치지표)

기업가치 지표		단위 : 억원, 주, %, 배				
IFRS 연결		2015/12	2016/12	2017/12	2018/12	2019/06
Per Share						
EPS	(원)	2,198	2,735	5,421	6,024	1,498
EBITDAPS	(원)	5,566	6,095	9,934	11,717	3,948
CFPS	(원)	4,658	5,263	8,321	9,659	3,557
SPS	(원)	23,588	24,632	31,414	33,458	15,975
BPS	(원)	21,903	24,340	28,971	35,342	36,789
Dividends						
DPS(보통주,현금)		420	570	850	1,416	708
DPS(1우선주,현금)		421	571	851	1,417	708
배당성향(현금)		16.42	17.81	14.09	21.92	
Multiples						
PER		11.47	13.18	9.40	6.42	
PCR		5.41	6.85	6.12	4.01	
PSR		1.07	1.46	1.62	1.16	
PBR		1.15	1.48	1.76	1.10	1.28
EV/Sales		1.08	1.43	1.56	1.09	
EV/EBITDA		4.58	5.79	4.94	3.10	
FCF						
총현금흐름		403,232	423,491	623,891	691,786	232,948
세후영업이익		193,923	216,362	402,717	426,966	93,086
유무형자산상각비		209,309	207,130	221,174	264,820	139,862
총투자		286,664	208,997	506,359	368,741	301,706
FCFF		116,568	214,494	117,533	323,045	-68,758

Company Guide에서 제공하는 가치지표 크롤링 가능
 주가가 현재 기준 주가가 아닌 재무데이터 발표 기준 주가 → 즉시성 반영이 안됨

4. Value (가치지표)

삼성전자 005930

KSE 코스피 전기,전자 | FICS 휴대폰 및 관련부품 | 12월 결산

PER 8.28 12M PER 12.27 업종 PER 6.58 PBR 1.41 배당수익률 2.84%



시세현황 [2019/10/18]		단위 : 원, 주, %	
종가/ 전일대비	49,900/ -600	거래량	8,483,106
52주.최고가/ 최저가	50,700/ 37,450	거래대금 (억원)	4,269
수익률 (1M/ 3M/ 6M/ 1Y)	+4.61/ +8.24/ +9.43/ +13.28	외국인 보유비중	57.51
시가총액 (상장예정포함,억원)	3,310,545	베타 (1년)	1.31629
시가총액 (보통주,억원)	2,978,921	액면가	100
발행주식수 (보통주/ 우선주)	5,969,782,550 / 822,886,700	유동주식수/비율 (보통주)	4,449,224,951 / 74.53

앞에서 구한 재무제표 데이터와 현재주가, 발행주식수를 이용하여 금일 기준 가치지표 계산 가능

4. Value (가치지표)

□ 가치지표 종류

종류	분모
PER	Earnings (순이익)
PBR	Book Value (순자산)
PCR	Cashflow (영업활동현금흐름)
PSR	Sales (매출액)

□ 가치지표 계산 (예시)

$$\text{PER} = \text{Price} / \text{EPS} = \text{주가} / (\text{순이익} / \text{주식수})$$

4. Value (가치지표)

```
value_type = c('지배주주순이익',  
              '자본',  
              '영업활동으로인한현금흐름',  
              '매출액')  
  
value_index = data_fs[match(value_type, rownames(data_fs)),  
                      ncol(data_fs)]  
  
print(value_index)
```

```
## [1] 438909 2477532 670319 2437714
```

```
url =  
  paste0('http://comp.fnguide.com/SVO2/ASP/SVD_main.asp?pGB=1&gicode=A005930')  
data = GET(url)  
  
price = read_html(data) %>%  
  html_node(xpath = '//*[@id="svdMainChartTxt11"]') %>%  
  html_text() %>%  
  readr::parse_number()  
  
print(price)
```

```
## [1] 49900
```

4. Value (가치지표)

```
share = read_html(data) %>%  
  html_node(  
    xpath =  
      '//*[@id="svdMainGrid1"]/table/tbody/tr[7]/td[1]' ) %>%  
  html_text()  
  
print(share)
```

```
## [1] "5,969,782,550/ 822,886,700"
```

```
share = share %>%  
  strsplit('/') %>%  
  unlist() %>%  
  .[1] %>%  
  readr::parse_number()  
  
print(share)
```

```
## [1] 5969782550
```

4. Value (가치지표)

```
data_value = price / (value_index * 100000000 / share)
names(data_value) = c('PER', 'PBR', 'PCR', 'PSR')
data_value[data_value < 0] = NA

print(data_value)
```

```
##          PER          PBR          PCR          PSR
## 6.787105 1.202375 4.444036 1.222014
```

5. Data Wrangling

```
head(KOR_price[, 1:7])
```

```
##           X005930 X000660 X005380 X068270 X051910 X012330 X005490
## 2017-06-21   47480   64800   160500   112857   280000   254500   267000
## 2017-06-22   47960   65000   161500   110896   287000   258000   269000
## 2017-06-23   47620   65000   164000   111092   283500   257000   270500
## 2017-06-26   48280   67500   164000   111778   282500   255500   272500
## 2017-06-27   48300   69200   160500   112857   283000   253500   275000
## 2017-06-28   47700   67200   160000   111778   280500   254500   284000
```

5. Data Wrangling

```
head(names(KOR_fs))
```

```
## [1] "매출액"      "매출원가"    "매출총이익"  
## [4] "판매비와관리비" "인건비"     "유무형자산상각비"
```

```
head(KOR_fs$`매출액`)
```

```
##           X2015.12 X2016.12 X2017.12 X2018.12  
## 005930         NA  2018667  2395754  2437714  
## 000660         NA   171980   301094   404451  
## 005380         NA   936490   963761   968126  
## 068270         NA    6706    9491    9821  
## 051910         NA   206593   256980   281830  
## 012330         NA   382617   351446   351492
```


5. Data Wrangling

```
head(KOR_value)
```

```
##           PER           PBR           PCR           PSR
## 1  6.209045  1.0999679  4.065536  1.1179350
## 2  3.204314  1.0628157  2.240289  1.2311840
## 3 20.047774  0.4091432  8.031785  0.3122946
## 4 101.957397 10.1384255 69.857227 27.1789498
## 5  16.993743  1.4446854 11.776464  0.8879461
## 6  11.782144  0.7248094 13.821572  0.6331328
```

6. Modeling

- ❑ 앞서 받은 티커에 for loop를 적용하면 모든 종목의 주가, 재무제표, 가치지표 다운로드 가능
- ❑ 다운로드 받은 데이터들을 이용하여 퀀트 모델링 적용
- ❑ 예제: 고 GPA + 저 PBR

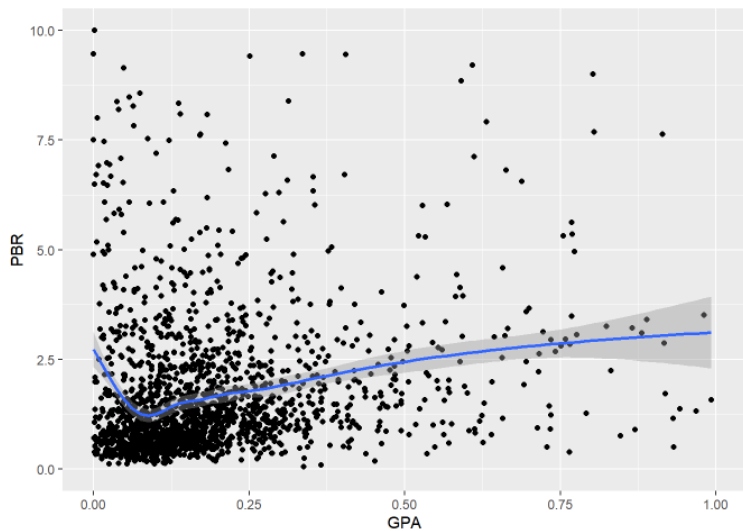
Novy-Marx, Robert. "The quality dimension of value investing." Rnm. simon. rochester. edu (2013): 1-54.

- ❑ 데이터: https://github.com/hyunyulhenry/quant_cookbook/tree/master/data

6. Modeling

```
num_col = ncol(KOR_fs[[1]])  
gpa = (KOR_fs$`매출총이익` / KOR_fs$`자산`)[num_col]  
pbr = KOR_value$PBR  
df = cbind(gpa, pbr) %>%  
  magrittr::set_colnames(c(`GPA`, `PBR`))
```

```
library(ggplot2)  
  
df %>%  
  ggplot(aes(x = GPA, y = PBR)) +  
  geom_point() +  
  geom_smooth() +  
  xlim(c(0, 1)) +  
  ylim(c(0, 10))
```



6. Modeling

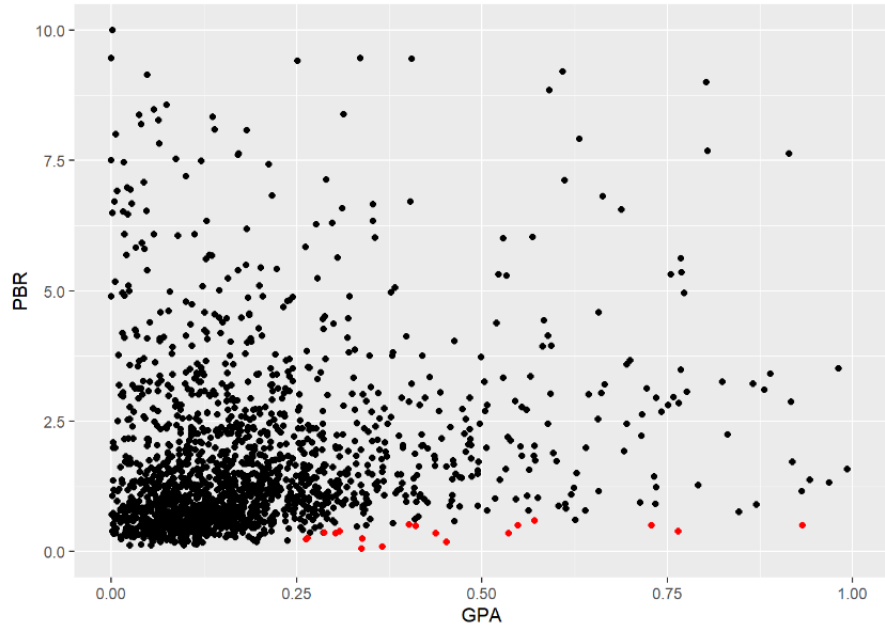
```
invest = df %>%
  tibble::rownames_to_column('ticker') %>%
  mutate(GPA_rank = min_rank(desc(GPA)),
         GPA_pbr = min_rank(PBR),
         sum = GPA_rank + GPA_pbr,
         rank = min_rank(sum)) %>%
  filter(rank <= 20) %>%
  select(ticker, GPA, PBR)

print(invest)
```

##	ticker	GPA	PBR
## 1	030200	0.7288280	0.50338907
## 2	001040	0.2632629	0.23300962
## 3	093050	0.5707713	0.59300321
## 4	006840	0.3077195	0.39322939
## 5	037560	0.5483438	0.50373728
## 6	084690	0.3389109	0.24476871
## 7	095570	0.4103530	0.48821201
## 8	005990	0.5358699	0.34820729
## 9	005740	0.2869024	0.36088346
## 10	001390	0.2642848	0.25665558
## 11	007540	0.4013711	0.51591605
## 12	005800	0.3023533	0.34884852
## 13	058850	2.9006566	0.59130736
## 14	058860	1.9668412	0.51594725
## 15	069640	0.9323930	0.50767973
## 16	084870	0.4380165	0.34514965
## 17	031980	0.4524703	0.17974574
## 18	002070	0.7646680	0.39359485
## 19	017680	0.3654135	0.08848606
## 20	149940	0.3378261	0.05385654

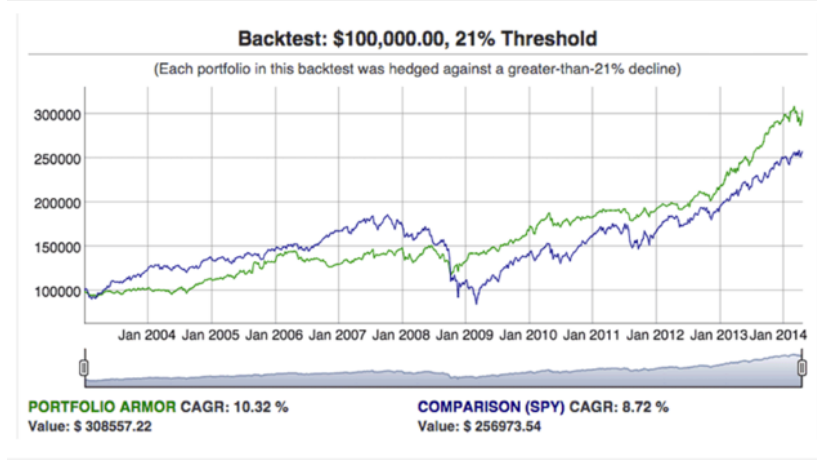
6. Modeling

```
df %>%  
  ggplot(aes(x = GPA, y = PBR)) +  
  geom_point() +  
  xlim(c(0, 1)) +  
  ylim(c(0, 10)) +  
  geom_point(  
    data = invest, aes(x = GPA, y = PBR), color = 'red'  
  )  
)
```



7. Backtest

- 현재 생각하는 전략을 과거부터 실행했을 때 어떠한 성과가 발생하는지 테스트해보는 과정
- 해당 전략의 손익뿐만 아니라 각종 위험을 대략적으로 판단



7. Backtest

- ‘리밸런싱 기간 = 수익률 기간’ 이면 문제가 없음

(예: 월간 리밸런싱, 월간 수익률 사용)

$$r_p = \sum_{i=1}^n w_i r_i$$

비중과 수익률의 행렬 곱으로 포트폴리오 수익률 계산 가능

- ‘리밸런싱 기간 != 수익률 기간’ : **Too complex**

(예: 분기 리밸런싱, 일간 수익률 사용)

7. Backtest

- ❑ Solution: **PerformanceAnalytics** 패키지의 **Return.portfolio()** 함수
- ❑ 각 자산의 수익률과 리밸런싱 시점의 비중만 필요
- ❑ 두 테이블의 기간이 달라도, 매칭시켜 수익률 계산해줌
(분기 리밸런싱, 일간 수익률 경우에도 이에 맞게 수익률이 계산됨)

7. Backtest

- ❑ <https://www.rdocumentation.org/packages/PerformanceAnalytics/versions/1.5.3/topics/Return.portfolio> (Document)
- ❑ <https://github.com/R-Finance/PerformanceAnalytics/blob/master/R/Return.portfolio.R> (R code)
- ❑ <https://github.com/hyunyulhenry/pyquant/blob/master/ReturnPortfolio.py>
(Python version by Me)

7. Backtest

1. 시계열 형태로 인식할 수 있도록 행 이름 혹은 인덱스가 날짜 형태로 입력되어야 함 (xts 형식 추천)
2. 수익률 데이터와 비중 데이터의 열 개수는 동일해야 하며, 각 열에 해당하는 자산은 동일해야 함.
3. 각 시점의 비중의 합은 1이 되어야 함. 그렇지 않을 경우 제대로 된 수익률이 계산되지 않음.

7. Backtest

	시작금액		시작합계	시작비중		수익률		종료금액		종료합계	종료비중		최종수익률
	1.주식	2.채권	3.1+2	4.주식	5.채권	6.주식	7.채권	8.주식	9.채권	10.8+9	11.주식	12.채권	13.최종
2017-12-26	1.603	0.940	2.543	0.630	0.370	-0.001	0.003	1.601	0.943	2.544	0.629	0.371	0.000
2017-12-27	1.601	0.943	2.544	0.629	0.371	0.000	0.013	1.602	0.956	2.557	0.628	0.374	0.005
2017-12-28	1.602	0.956	2.557	0.628	0.374	0.002	-0.001	1.605	0.955	2.560	0.627	0.373	0.001
2017-12-29	1.605	0.955	2.560	0.627	0.373	-0.004	0.002	1.599	0.956	2.555	0.628	0.374	-0.002
2018-01-02	1.533	1.022	2.555	0.600	0.400	0.007	-0.011	1.544	1.011	2.555	0.604	0.396	0.000
2018-01-03	1.544	1.011	2.555	0.604	0.396	0.006	0.005	1.554	1.016	2.570	0.605	0.395	0.006
2018-01-04	1.554	1.016	2.570	0.605	0.395	0.004	0.000	1.560	1.016	2.576	0.606	0.394	0.002

7. Backtest

- 글로벌 10개 자산 중 과거 12개월 수익률이 높은 5개 자산을 선택
- 포트폴리오를 구성하며, 개별 투자비중은 최소 10%, 최대 30% 제약조건을 설정
- 매월 리밸런싱을 실시

7. Backtest

```
symbols = c('SPY', # 미국 주식
            'IEV', # 유럽 주식
            'EIJ', # 일본 주식
            'EEM', # 이머징 주식
            'TLT', # 미국 장기채
            'IEF', # 미국 중기채
            'IYR', # 미국 리츠
            'RWX', # 글로벌 리츠
            'GLD', # 금
            'DBC' # 상품
            )
getSymbols(symbols, src = 'yahoo')
```

```
prices = do.call(cbind,
                lapply(symbols, function(x) Ad(get(x)))) %>%
  setNames(symbols)

rets = Return.calculate(prices) %>% na.omit()

head(rets)
```

```
##           SPY           IEV           EIJ           EEM
## 2007-01-04  0.0021220284 -0.005681697  0.010563372 -0.013809016
## 2007-01-05 -0.0079761695 -0.014761830 -0.025784116 -0.029238396
## 2007-01-08  0.0046253042  0.001643313  0.005722569  0.007257429
## 2007-01-09 -0.0008505326 -0.003763758  0.006401210 -0.022336310
## 2007-01-10  0.0033321560 -0.005618737 -0.014841092 -0.002303261
## 2007-01-11  0.0043802383  0.010813485 -0.005021513  0.012650133
##           TLT           IEF           IYR           RWX
## 2007-01-04  0.006062792  0.0036289043 -0.0002383499 -0.007436671
## 2007-01-05 -0.004351920 -0.0032537307 -0.0155038329 -0.012433769
## 2007-01-08  0.001793291  0.0007250350 -0.0002424869 -0.003390199
## 2007-01-09  0.000000000 -0.0002419919  0.0117533255  0.001619692
## 2007-01-10 -0.004475144 -0.0016917576  0.0159283329 -0.005659486
## 2007-01-11 -0.005844996 -0.0026626269  0.0114343574  0.003252343
##           GLD           DBC
## 2007-01-04 -0.0101155589 -0.026050332
## 2007-01-05 -0.0240065523 -0.003451290
## 2007-01-08  0.0051521025  0.005194705
## 2007-01-09  0.0061176918 -0.008613160
## 2007-01-10 -0.0042727692 -0.014769831
## 2007-01-11  0.0006601915 -0.001322726
```

7. Backtest

```
ep = endpoints(rets, on = 'months')
print(ep)
```

```
## [1] 0 19 38 60 80 102 123 144 167 186 209 230 250 271
## [15] 291 311 333 354 375 397 418 439 462 481 503 523 542 564
## [29] 585 605 627 649 670 691 713 733 755 774 793 816 837 857
## [43] 879 900 922 943 964 985 1007 1027 1046 1069 1089 1110 1132 1152
## [57] 1175 1196 1217 1238 1259 1279 1299 1321 1341 1363 1384 1405 1428 1447
## [71] 1468 1489 1509 1530 1549 1569 1591 1613 1633 1655 1677 1697 1720 1740
## [85] 1761 1782 1801 1822 1843 1864 1885 1907 1928 1949 1972 1991 2013 2033
## [99] 2052 2074 2095 2115 2137 2159 2180 2201 2223 2243 2265 2284 2304 2326
## [113] 2347 2368 2390 2410 2433 2454 2475 2496 2517 2537 2556 2579 2598 2620
## [127] 2642 2662 2685 2705 2727 2748 2768 2789 2808 2829 2850 2872 2893 2914
## [141] 2937 2956 2979 3000 3019 3040 3059 3080 3101 3123 3143 3165 3187 3207
## [155] 3224
```

```
head(index(rets)[ep], 10)
```

```
## [1] "2007-01-31 UTC" "2007-02-28 UTC" "2007-03-30 UTC" "2007-04-30 UTC"
## [5] "2007-05-31 UTC" "2007-06-29 UTC" "2007-07-31 UTC" "2007-08-31 UTC"
## [9] "2007-09-28 UTC" "2007-10-31 UTC"
```

7. Backtest

```
wts = list()
lookback = 12
wt_zero = rep(0, 10) %>% setNames(colnames(rets))
print(wt_zero)
```

```
## SPY IEV EWJ EEM TLT IEF IYR RWX GLD DBC
##  0  0  0  0  0  0  0  0  0  0
```

7. Backtest

```
for (i in (lookback+1) : length(ep)) {
  sub_ret = rets[ep[i-lookback] : ep[i] , ]
  cum = Return.cumulative(sub_ret)

  K = rank(-cum) <= 5
  covmat = cov(sub_ret[, K])

  wt = wt_zero
  wt[K] = optimalPortfolio(covmat,
                          control = list(type = 'minvol',
                                          constraint = 'user',
                                          LB = rep(0.10, 5),
                                          UB = rep(0.30, 5)))

  wts[[i]] = xts(t(wt), order.by = index(rets[ep[i]]))
}

wts = do.call(rbind, wts)
head(wts)
```

##	SPY	IEV EWJ	EEM TLT	IEF	IYR	RWX	GLD	DBC
## 2007-12-31	0 0.12658	0 0.1000000	0.0 0.3	0	0	0 0.1734200	0.3000000	
## 2008-01-31	0 0.00000	0 0.1000000	0.3 0.3	0	0	0 0.1000000	0.2000000	
## 2008-02-29	0 0.00000	0 0.1000000	0.3 0.3	0	0	0 0.1000000	0.2000000	
## 2008-03-31	0 0.00000	0 0.1060958	0.3 0.3	0	0	0 0.1148597	0.1790445	
## 2008-04-30	0 0.00000	0 0.1141941	0.3 0.3	0	0	0 0.1000000	0.1858059	
## 2008-05-30	0 0.00000	0 0.1247180	0.3 0.3	0	0	0 0.1000000	0.1752820	

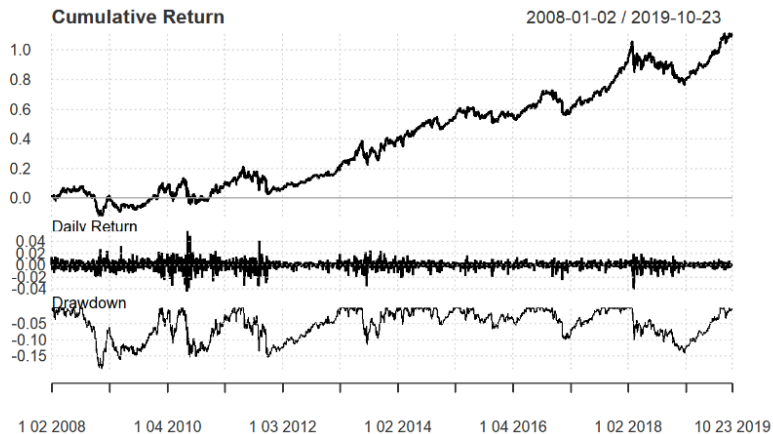
7. Backtest

```
GDA = Return.portfolio(rets, wts, verbose = TRUE)
summary(GDA)
```

```
##           Length Class Mode
## returns      2974  xts  numeric
## contribution 29740  xts  numeric
## BOP.Weight    29740  xts  numeric
## EOP.Weight    29740  xts  numeric
## BOP.Value     29740  xts  numeric
## EOP.Value     29740  xts  numeric
```

```
charts.PerformanceSummary(GDA$returns, main = '동적자산배분')
```

동적자산배분



7. Backtest

	시작금액		시작합계	시작비중		수익률		종료금액		종료합계	종료비중		최종수익률
	1.주식	2.채권	3.1+2	4.주식	5.채권	6.주식	7.채권	8.주식	9.채권	10.8+9	11.주식	12.채권	13.최종
2017-12-26	1.603	0.940	2.543	0.630	0.370	-0.001	0.003	1.601	0.943	2.544	0.629	0.371	0.000
2017-12-27	1.601	0.943	2.544	0.629	0.371	0.000	0.013	1.602	0.956	2.557	0.628	0.374	0.005
2017-12-28	1.602	0.956	2.557	0.628	0.374	0.002	-0.001	1.605	0.955	2.560	0.627	0.373	0.001
2017-12-29	1.605	0.955	2.560	0.627	0.373	-0.004	0.002	1.599	0.956	2.555	0.626	0.374	-0.002
2018-01-02	1.533	1.022	2.555	0.600	0.400	0.007	-0.011	1.544	1.011	2.555	0.604	0.396	0.000
2018-01-03	1.544	1.011	2.555	0.604	0.396	0.006	0.005	1.554	1.016	2.570	0.605	0.395	0.006
2018-01-04	1.554	1.016	2.570	0.605	0.395	0.004	0.000	1.560	1.016	2.576	0.606	0.394	0.002

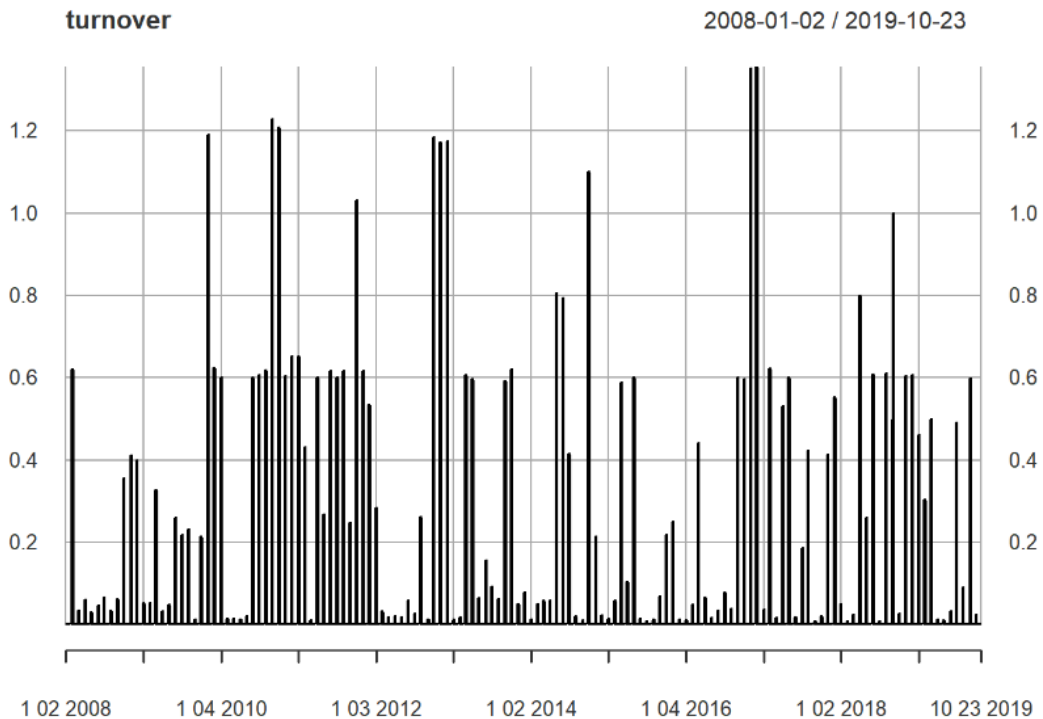
7. Backtest

```
turnover = xts(  
  rowSums(abs(GDAASBOP.Weight -  
            timeSeries::lag(GDAASEOP.Weight)),  
          na.rm = TRUE),  
  order.by = index(GDAASBOP.Weight))  
  
head(turnover, 25)
```

```
##           [,1]  
## 2008-01-02 0.000000  
## 2008-01-03 0.000000  
## 2008-01-04 0.000000  
## 2008-01-07 0.000000  
## 2008-01-08 0.000000  
## 2008-01-09 0.000000  
## 2008-01-10 0.000000  
## 2008-01-11 0.000000  
## 2008-01-14 0.000000  
## 2008-01-15 0.000000  
## 2008-01-16 0.000000  
## 2008-01-17 0.000000  
## 2008-01-18 0.000000  
## 2008-01-22 0.000000  
## 2008-01-23 0.000000  
## 2008-01-24 0.000000  
## 2008-01-25 0.000000  
## 2008-01-28 0.000000  
## 2008-01-29 0.000000  
## 2008-01-30 0.000000  
## 2008-01-31 0.000000  
## 2008-02-01 0.620987  
## 2008-02-04 0.000000  
## 2008-02-05 0.000000  
## 2008-02-06 0.000000
```

7. Backtest

```
plot(turnover)
```



7. Backtest

```
wts_mod = list()

for (i in (lookback+1) : length(ep)) {

  ret_z = lapply(3:12, function(x) {
    scale(rank(Return.cumulative( rets[c(ep[i-x] : ep[i]) , ] )))
  })

  K = (rank(- apply(do.call(cbind, ret_z), 1, sum), ties.method = "first") <= 5 )

  sub_ret = rets[ep[i-lookback] : ep[i] , ]
  covmat = cov(sub_ret[, K])

  wt = wt_zero
  wt[K] = optimalPortfolio(covmat,
                          control = list(type = 'minvol',
                                          constraint = 'user',
                                          LB = rep(0.10, 5),
                                          UB = rep(0.30, 5)))

  wts_mod[[i]] = xts(t(wt), order.by = index(rets[ep[i]]))

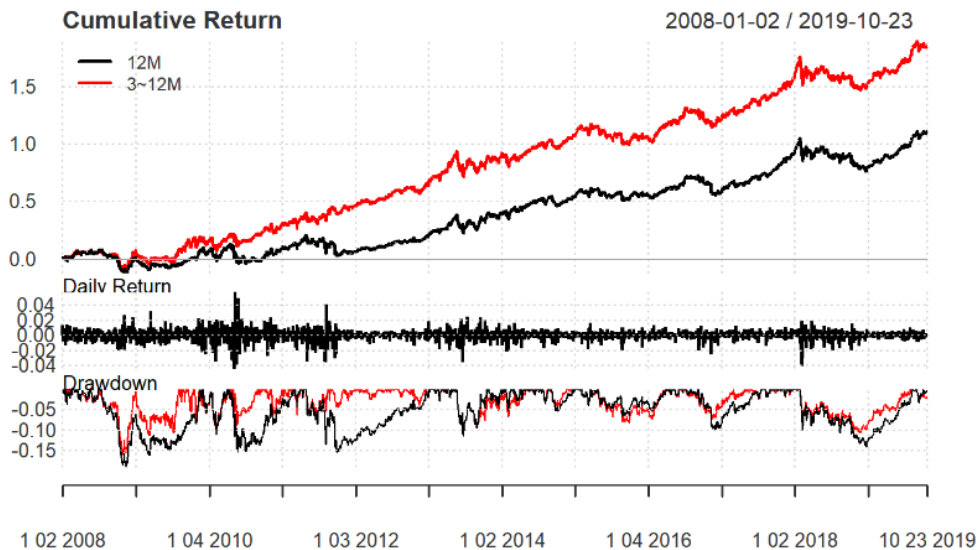
}

wts_mod = do.call(rbind, wts_mod)
```

7. Backtest

```
GDA_mod = Return.portfolio(rets, wts_mod, verbose = TRUE)
df = cbind(GDAA$return, GDA_mod$return) %>%
  set_colnames(c('12M', '3~12M'))
charts.PerformanceSummary(df, main = '동적자산배분')
```

동적자산배분



7. Backtest

```
ls("package:PerformanceAnalytics") %>% head(30)
```

```
## [1] "ActivePremium"      "ActiveReturn"      "AdjustedSharpeRatio"  
## [4] "allsymbols"        "apply.fromstart"   "apply.rolling"  
## [7] "AppraisalRatio"    "AverageDrawdown"   "AverageLength"  
## [10] "AverageRecovery"    "BernardoLedoitRatio" "BetaCoKurtosis"  
## [13] "BetaCoSkewness"    "BetaCoVariance"    "bluefocus"  
## [16] "bluemono"          "bond.dates"        "bond.labels"  
## [19] "BurkeRatio"        "CalculateReturns"   "CalmarRatio"  
## [22] "CAPM.alpha"        "CAPM.beta"         "CAPM.beta.bear"  
## [25] "CAPM.beta.bull"    "CAPM.CML"          "CAPM.CML.slope"  
## [28] "CAPM.dynamic"      "CAPM.epsilon"      "CAPM.jensenAlpha"
```

```
ls("package:PerformanceAnalytics") %>% tail(30)
```

```
## [1] "to.quarterly.contributions" "to.weekly.contributions"  
## [3] "to.yearly.contributions"    "tol10qualitative"  
## [5] "tol11qualitative"          "tol12qualitative"  
## [7] "tol14rainbow"              "tol15rainbow"  
## [9] "tol18rainbow"              "tol1qualitative"  
## [11] "tol21rainbow"              "tol2qualitative"  
## [13] "tol3qualitative"           "tol4qualitative"  
## [15] "tol5qualitative"           "tol6qualitative"  
## [17] "tol7qualitative"           "tol8qualitative"  
## [19] "tol9qualitative"           "TotalRisk"  
## [21] "TrackingError"             "TreynorRatio"  
## [23] "UlcerIndex"                "UpDownRatios"  
## [25] "UpsideFrequency"           "UpsidePotentialRatio"  
## [27] "UpsideRisk"                "VaR"  
## [29] "VolatilitySkewness"        "zerofill"
```

7. Backtest

```
list(  
  'Cumulative Return' = Return.cumulative(df),  
  'Annual Return' = Return.annualized(df),  
  'Annual Vol' = StdDev.annualized(df),  
  'Sharpe Ratio' = SharpeRatio.annualized(df),  
  'MDD' = maxDrawdown(df)  
)
```

```
## $`Cumulative Return`  
##           12M      3~12M  
## Cumulative Return 1.108257 1.852295  
##  
## $`Annual Return`  
##           12M      3~12M  
## Annualized Return 0.06523997 0.0928753  
##  
## $`Annual Vol`  
##           12M      3~12M  
## Annualized Standard Deviation 0.1096576 0.1043323  
##  
## $`Sharpe Ratio`  
##           12M      3~12M  
## Annualized Sharpe Ratio (Rf=0%) 0.5949425 0.8901874  
##  
## $MDD  
##           12M      3~12M  
## Worst Drawdown 0.1883655 0.1588437
```


7. Backtest

```
apply.quarterly(df, Return.cumulative) %>% tail(10)
```

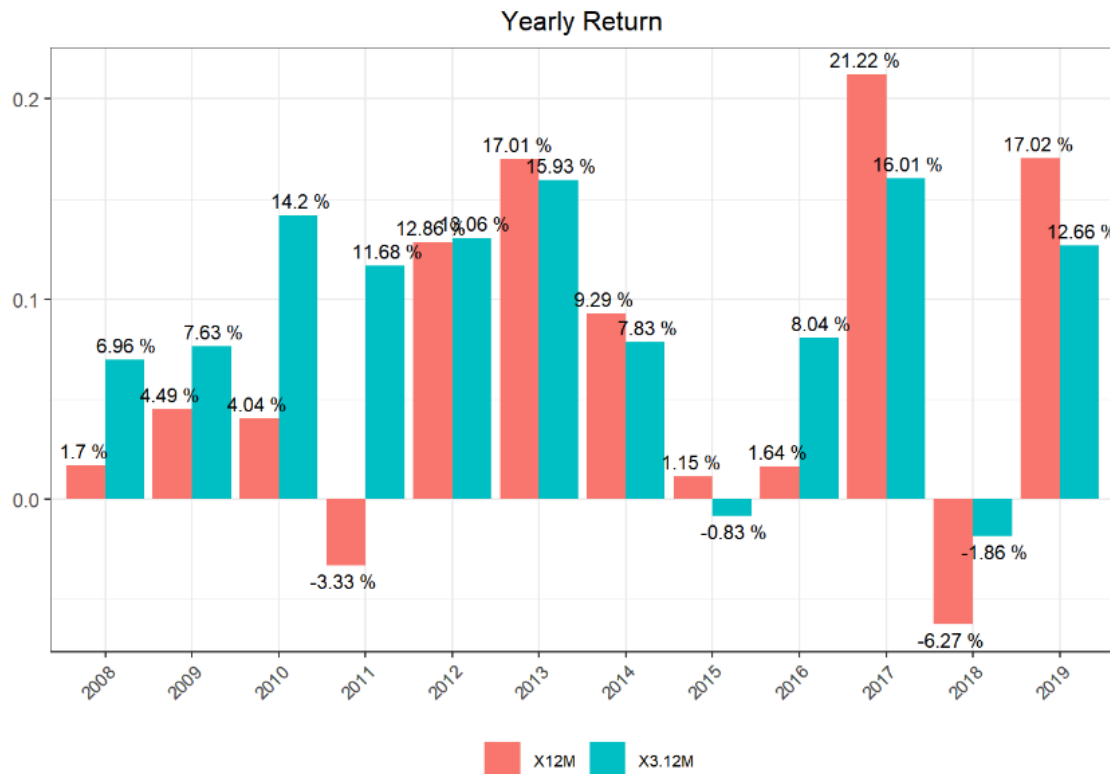
```
##           12M      3~12M
## 2017-09-29  0.0459728206  0.028067285
## 2017-12-29  0.0753977050  0.061614078
## 2018-03-29 -0.0008400019  0.009612223
## 2018-06-29 -0.0093952870 -0.007436998
## 2018-09-28  0.0044968537  0.002583517
## 2018-12-31 -0.0572513625 -0.023177711
## 2019-03-29  0.0616367604  0.051183799
## 2019-06-28  0.0341861458  0.021958856
## 2019-09-30  0.0567192570  0.049854629
## 2019-10-23  0.0086128828 -0.001052169
```

```
apply.yearly(df, Return.cumulative)
```

```
##           12M      3~12M
## 2008-12-31  0.01697197  0.069636100
## 2009-12-31  0.04492538  0.076314247
## 2010-12-31  0.04037717  0.142013691
## 2011-12-30 -0.03332547  0.116826702
## 2012-12-31  0.12858724  0.130616511
## 2013-12-31  0.17005346  0.159339673
## 2014-12-31  0.09289859  0.078280001
## 2015-12-31  0.01148835 -0.008340307
## 2016-12-30  0.01642562  0.080439201
## 2017-12-29  0.21216642  0.160135648
## 2018-12-31 -0.06269718 -0.018593786
## 2019-10-23  0.17019650  0.126637094
```

7. Backtest

```
HenryQuant::plot_yearly(df)
```



8. Report

한경컨센서스 2019-09-20 ~ 2019-10-20 전체 분류선택 검색 Q 등록

전체 기업 산업 시장 파생 경제 상황 하향 기업정보 LIST 20 50 80

작성일	분류	제목	작성자	제공출처	첨부파일
2019-10-18	산업	에너지-SK Energy Daily	손지우,박한샘	SK증권	
2019-10-18	시장	다음주에 불안한 차트	한대훈	SK증권	
2019-10-18	채권	Daily BOND MORNING BRIEF	리서치센터	KTB투자증권	
2019-10-18	기업	LG디스플레이(034220)3 Q Preview: 시련의 계절	이승우	유진투자증권	
2019-10-18	산업	화환-IBKS Petrochem Weekly	할철도	IBK투자증권	
2019-10-18	기타	경제지표 및 이벤트 캘린더	정원일	유안타증권	
2019-10-18	산업	인터넷-IBKS Daily [인터넷/게임]	이승훈	IBK투자증권	
2019-10-18	기타	경제지표 및 이벤트 캘린더	박준홍	유안타증권	
2019-10-18	기타	주식시장 지표	박준홍	유안타증권	
2019-10-18	시장	김예은의 견학e아기	김예은	IBK투자증권	
2019-10-18	시장	Start with IBKS	투자전략팀	IBK투자증권	
2019-10-18	외환	Daily Forex Live	민경원	우리은행	
2019-10-18	퀀트	Bonjour Quant Daily	강봉주	메리츠증권증권	
2019-10-18	기업	키움증권(039490)	김고은	메리츠증권증권	
2019-10-18	기업	NH투자증권(005940)	김고은	메리츠증권증권	
2019-10-18	기업	미래에셋대우(006800)	김고은	메리츠증권증권	
2019-10-18	기업	삼성증권(016360)	김고은	메리츠증권증권	
2019-10-18	기업	한국금융지주(071050)	김고은	메리츠증권증권	
2019-10-18	시장	미국 소매판매 해부: 연말 소비시즌, 눈높이를 조심스럽게 낮춰...	김재현	현대차투자증권	
2019-10-18	기타	Morning Meeting Notes	백주현	현대차투자증권	

PDF 리더기 다운로드

한경컨센서스(<http://hkconsensus.hankyung.com/apps.analysis/analysis.list>)에서 애널리스트 리포트 확인가능
 → 테이블 다운 및 PDF 부분의 주소만 선택하기

8. Report

```
url = 'http://hkconsensus.hankyung.com/apps.analysis/analysis.list'
```

```
Sys.setlocale("LC_ALL", "English")
```

```
down = GET(url) %>%  
  read_html(encoding = "EUC-KR")
```

```
down_table = down %>%  
  html_table() %>%  
  .[[1]]
```

```
Sys.setlocale("LC_ALL", "Korean")
```

```
head(down_table, 5)
```

```
## 작성일 분류  
## 1 2019-10-18 산업  
## 2 2019-10-18 시장  
## 3 2019-10-18 채권  
## 4 2019-10-18 기업  
## 5 2019-10-18 산업  
##  
## 제목  
## 1 에너지-SK Energy Daily  
#r#n #r#n #r#n 에너지-SK Energy Daily  
## 2 다음주에 불안한 차트#r#n  
#r#n 다음주에 불안한 차트#r#n  
태평양을 건너 대서양으로 시선 이동  
## 3 Daily BOND MORNING BRIEF#r#n  
#r#n Daily BOND MORNING BRIEF#r#n  
Daily BOND MORNING BRIEF  
## 4 LG디스플레이(034220)3 Q Preview: 시련의 계절#r#n #r#n  
#r#n LG디스플레이(034220)3 Q Preview: 시련의 계절#r#n  
<J+F06E> 3 분기 영업 적자 4, 0.00 억원 대 전망<J+F06E> 4 분기 및 2020년 전망  
## 5 화학-IBKS Petrochem Weekly#r#n  
#r#n 화학-IBKS Petrochem Weekly#r#n  
화학-IBKS Petrochem Weekly  
## 작성자 제공출처 첨부파일  
## 1 손지우,박한샘 SK증권 NA  
## 2 한대훈 SK증권 NA  
## 3 리서치센터 KTB투자증권 NA  
## 4 이승우 유진투자증권 NA  
## 5 함형도 IBK투자증권 NA
```

8. Report

```
down_table$`제목` = lapply(down_table$`제목`, function(x) {  
  stringr::str_split(x, pattern = '\\r\\n', simplify = TRUE)[1]  
}) %>% unlist()
```

```
down_table$`제목`
```

```
## [1] "에너지-SK Energy Daily"  
## [2] "다음주에 불안한 차트"  
## [3] "Daily BOND MORNING BRIEF"  
## [4] "LG디스플레이(034220)3 Q Preview: 시련의 계절"  
## [5] "화학-IBKS Petrochem Weekly"  
## [6] "경제지표 및 이벤트 캘린더"  
## [7] "인터넷-IBKS Daily [인터넷/게임]"  
## [8] "경제지표 및 이벤트 캘린더"  
## [9] "주식시장 지표"  
## [10] "김예은의 전략e야기"  
## [11] "Start with IBKS"  
## [12] "Daily Forex Live"  
## [13] "Bonjour Quant Daily"  
## [14] "키움증권(039490)"  
## [15] "NH투자증권(005940)"  
## [16] "미래에셋대우(006800)"  
## [17] "삼성증권(016360)"  
## [18] "한국금융지주(071050)"  
## [19] "미국 소매판매 해부: 연말 소비시즌, 눈높이를 조심스럽게 낮춰..."  
## [20] "Morning Meeting Notes"
```

8. Report

```
report_name =  
  paste(down_table$`작성일`, down_table$`작성자`,  
        down_table$`제목`, down_table$`제공출처`, sep = '_')  
  
report_name = stringr::str_replace_all(report_name, '[:punct:]]+', '_')  
head(report_name)
```

```
## [1] "2019_10_18_손지우_박한샘_에너지_SK Energy Daily_SK증권"  
## [2] "2019_10_18_한대훈_다음주에 불안한 차트_SK증권"  
## [3] "2019_10_18_리서치센터_Daily BOND MORNING BRIEF_KTB투자증권"  
## [4] "2019_10_18_이승우_LG디스플레이_034220_3 Q Preview_ 시련의 계절_유진투자증권"  
## [5] "2019_10_18_함형도_화학_IBKS Petrochem Weekly_IBK투자증권"  
## [6] "2019_10_18_정원일_경제지표 및 이벤트 캘린더_유안타증권"
```

8. Report

```
report_url = down %>%  
  html_nodes('.dv_input') %>% html_nodes('a') %>% html_attr('href')  
head(report_url)
```

```
## [1] "/apps.analysis/analysis.downpdf?report_idx=541020"  
## [2] "/apps.analysis/analysis.downpdf?report_idx=541019"  
## [3] "/apps.analysis/analysis.downpdf?report_idx=541018"  
## [4] "/apps.analysis/analysis.downpdf?report_idx=541017"  
## [5] "/apps.analysis/analysis.downpdf?report_idx=541016"  
## [6] "/apps.analysis/analysis.downpdf?report_idx=541015"
```

```
report_url = paste0('http://hkconsensus.hankyung.com',  
  report_url)  
head(report_url)
```

```
## [1] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541020"  
## [2] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541019"  
## [3] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541018"  
## [4] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541017"  
## [5] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541016"  
## [6] "http://hkconsensus.hankyung.com/apps.analysis/analysis.downpdf?report_idx=541015"
```

9. AWS를 이용한 자동화

AWS 내에 R, R studio를 설치하여 자동화 가능

매월 말 데이터 다운로드, 레포트 및 공시를 텔레그램으로 전송 등

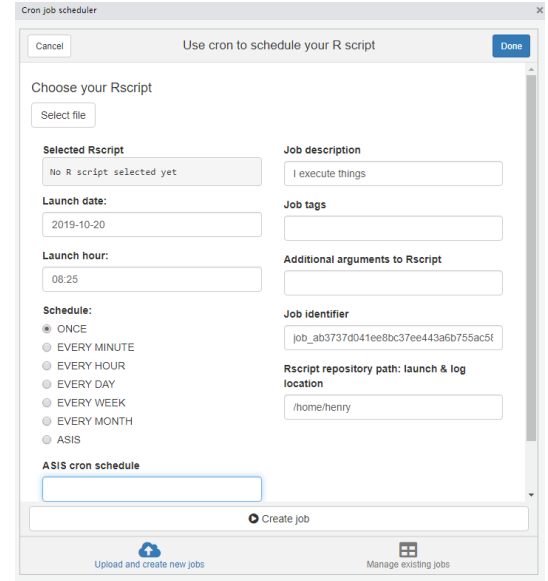
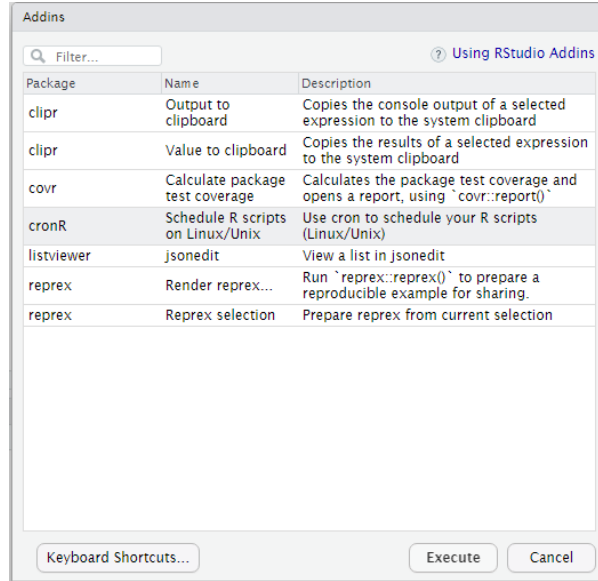
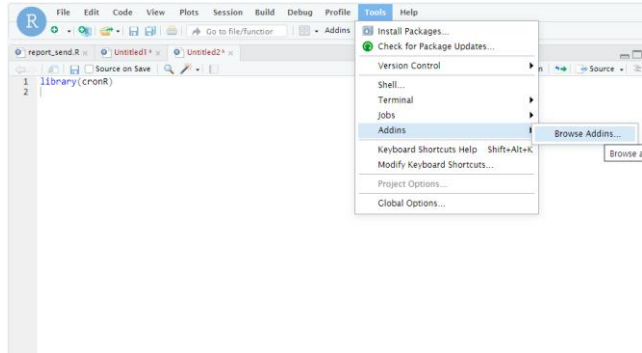
1. AWS EC2 인스턴스 만들고 접속하기: http://henryquant.blogspot.com/2019/09/aws-1-aws-ec2_6.html
2. R, R studio 설치 및 패키지 설치하기: http://henryquant.blogspot.com/2019/09/aws-2-r-r-studio_6.html
3. 스케줄러를 이용한 텔레그램 자동 전송: http://henryquant.blogspot.com/2019/09/aws-3_6.html
4. 샤이니 서버 설치 및 app 실행하기: http://henryquant.blogspot.com/2019/09/aws-4-app_6.html

9. AWS를 이용한 자동화

- ❑ cronR 패키지를 이용하여 스케줄링 지정
- ❑ Addins 기능 이용하여 손쉽게 이용 가능
- ❑ cron expression 이용하여 원하는 스케줄링 선택 가능

(<https://www.freeformatter.com/cron-expression-generator-quartz.html>)

9. AWS를 이용한 자동화



9. AWS를 이용한 자동화

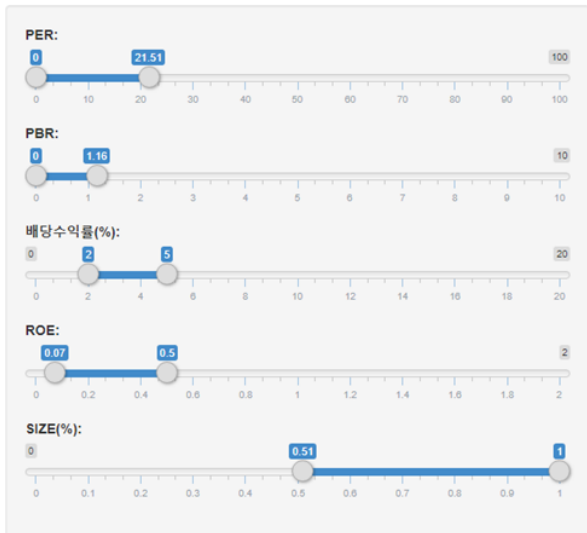
- ❑ Shiny를 이용하여 포트폴리오 실시간 수익률 및 기타 사항 관리 가능
- ❑ <https://mastering-shiny.org/>
- ❑ R Shiny 프로그래밍 가이드 (고석범 저)

9. AWS를 이용한 자동화



9. AWS를 이용한 자동화

Stock Screener



Copy CSV Excel PDF Search:

일자	종목코드	종목명	종가	PER	PBR	배당수익률	ROE	시총순위
2019-10-16	034300	신세계건설	25150	2.34	0.51	2.98	0.22	0.52
2019-10-16	092070	디엔에프	9450	6.93	1.13	2.12	0.16	0.52
2019-10-16	004450	삼화왕관	46000	8.64	0.71	2.93	0.08	0.53
2019-10-16	053690	한미글로벌	9000	12.91	1	2.22	0.08	0.53
2019-10-16	120240	대정화금	13750	9.16	0.85	2.18	0.09	0.53
2019-10-16	134380	미원화학	42350	11.52	1.03	2.36	0.09	0.55
2019-10-16	011560	세보엠이씨	8490	3.91	0.56	3.53	0.14	0.57
2019-10-16	016740	두울	3280	7.54	0.68	2.44	0.09	0.57
2019-10-16	058730	다스코	5770	4.89	0.7	3.47	0.14	0.57
2019-10-16	083930	아바코	5550	3.23	0.74	3.6	0.23	0.57
2019-10-16	131290	티에스이	8000	6.5	0.6	2.5	0.09	0.57
2019-10-16	094850	참좋은여행	5990	8.05	1.01	2.84	0.13	0.6
2019-10-16	155650	와이엠씨	4035	4.96	0.94	4.96	0.19	0.62
2019-10-16	031310	아이즈비전	4635	6.33	0.81	2.16	0.13	0.65
2019-10-16	037350	성도이엔지	4720	4.61	0.38	2.12	0.08	0.65
2019-10-16	039010	현대통신	8170	4.13	0.94	3.67	0.23	0.66
2019-10-16	008260	NI스틸	2340	6.54	0.63	2.14	0.1	0.68
2019-10-16	115310	인포바인	20950	4.72	0.85	4.53	0.18	0.68
2019-10-16	024120	KB오토시스	5700	5.14	0.69	4.91	0.13	0.69
2019-10-16	004550	케이엔지	5940	8.84	0.88	3.45	0.08	0.7