## [ library ]

library("tidyverse")

library("data.table")

library("gridExtra")

library("caret")

library("MLmetrics")

library("lubridate")

library("factoextra")

library("corrplot")

library("Hmisc")

## [ data loading ]

setwd("C:/Users/10sop/Desktop/통계분석학회PSAT/주제분석/데이터/주제분석\_데이터")

raw\_BTC\_price <- "BTC\_KRW coinmarketcap.csv" %>% fread # 단위: 원

raw\_exchange\_rate <- "USD\_KRW\_exchange\_rate.csv" %>% fread # 단위: 달러vs원

raw\_gold\_price <- "Gold\_price.csv" %>% fread # 단위: 달러/oz

raw\_USA\_bond <- "USA\_10year\_bond\_.csv" %>% fread # 단위: %

raw\_kospi <- "KOSPI.csv" %>% fread

raw\_consumer\_price <- "consumer\_price\_index.csv" %>% fread(head = FALSE)

raw\_bitcoin\_circulation <- "bitcoin\_circulation.csv" %>% fread # 단위: 달러

raw\_VIX <- "변동성지수\_VIX.csv" %>% fread

raw\_nasdaq <- "nasdaq.csv" %>% fread # 단위 : 달러

raw\_VKOSPI <- "VKOSPI.csv" %>% fread

raw\_BTC\_price

## [ data 가공 ]

# 1. 날짜 데이터 형식 맞추기

raw\_BTC\_price[, "날짜"] <- raw\_BTC\_price[, "날짜"] %>% as.matrix %>% dmy

raw\_BTC\_price[, "날짜"]

raw\_exchange\_rate[, 'Date'] <- raw\_exchange\_rate[,'Date'] %>% as.matrix %>% dmy

raw\_exchange\_rate[, 'Date']

raw\_gold\_price[, "Date"] <- raw\_gold\_price[, "Date"] %>% as.matrix %>% mdy()

raw\_gold\_price[, "Date"]

raw\_USA\_bond[, "날짜"] <- raw\_USA\_bond[, "날짜"] %>% as.matrix %>% ymd()

raw\_USA\_bond[,'날짜']

raw\_kospi[, "날짜"] <- raw\_kospi[, "날짜"] %>% as.matrix %>% ymd()

raw\_kospi[, "날짜"]

consumer\_price <- (raw\_consumer\_price[c(1,2),] %>% t() %>% as.data.frame)[-c(1,2), ]

names(consumer\_price) <- c("연-월 날짜", "총\_지수")

consumer\_price[, '연-월 날짜'] %>% as.matrix %>% ym()

consumer\_price[, '연-월 날짜'] <- consumer\_price[, '연-월 날짜'] %>% as.matrix %>% ym()

consumer\_price

raw\_bitcoin\_circulation[,'DateTime'] <- raw\_bitcoin\_circulation[, "DateTime"] %>% as.matrix %>% as.Date

raw\_VIX[, "날짜"] <- raw\_VIX[, "날짜"] %>% as.matrix %>% ymd()

raw\_nasdaq[, 'Date'] <- raw\_nasdaq[, "Date"] %>% as.matrix %>% ymd()

raw\_VKOSPI[, '일자'] <- raw\_VKOSPI[, "일자"] %>% as.matrix %>% ymd

## [ 필요한 데이터만 추출하기]

# 종속변수: 비트코인 일일 수익률

a <- gsub("\\\\",'', raw\_BTC\_price[, '시가\*'] %>% as.matrix) # 시가 안의 \\ 삭제.

b <- gsub(",", "", a) # 시가 안의 ',' 삭제.

raw\_BTC\_price[, "시가\*"] <- b

BTC\_price <- raw\_BTC\_price[, c("날짜", "시가\*")]

BTC\_price[, '시가\*'] <- BTC\_price[,'시가\*'] %>% as.matrix %>% as.numeric

names(BTC\_price) <- c("날짜", "BTC\_가격")

BTC\_price[, 'BTC\_가격'] %>% str

# 변수 (1) USD 환율

exchange\_rate <- raw\_exchange\_rate[, c("Date", "Price")]

names(exchange\_rate) <- c("날짜", "환율")

exchange\_rate[,"환율"] <- gsub(",", "",exchange\_rate[, "환율"] %>% as.matrix)

exchange\_rate[, "환율"] <- exchange\_rate[, "환율"] %>% as.matrix %>% as.numeric

exchange\_rate %>% str

# 변수 (2) gold 가격

gold\_price <- raw\_gold\_price[, c('Date', 'Close/Last')]

names(gold\_price) <- c("날짜", "금\_가격")

gold\_price %>% str

# 변수 (3) 미국 채권 수익률

USA\_bond <- raw\_USA\_bond[, c("날짜", "종가")]

names(USA\_bond) <- c("날짜", "채권\_수익률")

USA\_bond %>%str

# 변수 (4) KOSPI 수익률

kospi <- raw\_kospi[, c("날짜", "종가")]

kospi[,"종가"] <- gsub(",", "",kospi[, "종가"] %>% as.matrix)

kospi[, "종가"] <- kospi[, "종가"] %>% as.matrix %>% as.numeric

names(kospi) <- c("날짜", "KOSPI")

kospi %>% str

# 변수 (5) 소비자물가지수

names(consumer\_price) <- c("연-월 날짜", "소비자물가지수")

consumer\_price[,"소비자물가지수"] <- consumer\_price[, "소비자물가지수"] %>% as.matrix %>% as.numeric

consumer\_price %>% str

# 변수 (6) 비트코인 공급량 (단위: 1개)

bitcoin\_circulation <- raw\_bitcoin\_circulation[, c("DateTime", "Total Supply")]

names(bitcoin\_circulation) <- c("날짜", "BTC\_공급량")

bitcoin\_circulation %>% str

# 변수 (7) 변동성지수 VIX

VIX <- raw\_VIX[, c("날짜", "종가")]

names(VIX) <- c("날짜", "VIX\_value")

# 변수 (8) NASDAQ

nasdaq <- raw\_nasdaq[, c("Date", "Close")]

names(nasdaq) <- c("날짜", "Nasdaq\_종가")

# 변수 (9) VKOSPI

VKOSPI <- raw\_VKOSPI[, c("일자", "종가")]

names(VKOSPI) <- c("날짜", "VKOSPI\_종가")

## [ 모든 변수 데이터 합치기. ]

# 종속변수;BTC\_profit\_rate: (13.04.29 ~ 21.04.13)

# exchange\_rate: (11.01.03 ~ 21.04.01)

# gold\_price: (11.04.14 ~ 21.04.13)

# USA\_bond: (11.01.03 ~ 21.04.15)

# kospi: (11.01.03 ~ 21.04.14)

# consumer\_price: (10.01.01 ~ 21.02.01) -> 월별 데이터.

# BTC circulation

# VIX

# NASDAQ

# VKOSPI

temperary\_frame <- inner\_join(exchange\_rate, gold\_price, by = "날짜") %>%

inner\_join(USA\_bond, by = "날짜") %>%

inner\_join(kospi, by = "날짜") %>%

inner\_join(BTC\_price, by = '날짜') %>%

inner\_join(bitcoin\_circulation, by = '날짜') %>%

inner\_join(VIX, by = '날짜') %>%

inner\_join(nasdaq, by = "날짜") %>%

inner\_join(VKOSPI, by = "날짜") # 일별 데이터는 모두 합쳤음.

temperary\_frame2 <- temperary\_frame %>% mutate(temperary\_frame[, "날짜"] %>% format("%Y-%m")) # 일별 데이터에 월별 데이터를 합치기 위해 작업.

names(temperary\_frame2)[(temperary\_frame2 %>% names %>% length)] <- "연-월 날짜"

temperary\_frame2[, "연-월 날짜"] <- temperary\_frame2[, "연-월 날짜"] %>% as.matrix %>% ym

temperary\_frame3 <- inner\_join(temperary\_frame2, consumer\_price, by = "연-월 날짜") # 일별 데이터의 각 월마다 월별 데이터 값 inner\_join.

final\_frame<- temperary\_frame3[, -"연-월 날짜"]

final\_frame # 최종 data frame

## [ EDA ]

# 1. 모든 설명변수와 종속변수의 그래프를 한 공간에 그리기 - 추세 확인.

USD\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = 환율), color = "lightblue") +

geom\_line(aes(y = BTC\_가격/6000), color = 'orange') +

scale\_y\_continuous(

name = "USD axis",

sec.axis = sec\_axis(~.\*6000 , name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs USD/KRW 환율",

x = "날짜")+

theme(plot.title = element\_text(hjust = 0.5))

Gold\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = 금\_가격), color = "#CC3399") +

geom\_line(aes(y = BTC\_가격/6000), color = 'orange') +

scale\_y\_continuous(

name = "Gold axis",

sec.axis = sec\_axis(~. \*6000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs 금 가격",

x = "날짜")+

theme(plot.title = element\_text(hjust = 0.5))

Bond\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = 채권\_수익률), color = "green") +

geom\_line(aes(y = BTC\_가격/4000000), color = 'orange') +

scale\_y\_continuous(

name = "Bond axis",

sec.axis = sec\_axis(~. \*4000000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs 미국 10년 채권 가격",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

Kospi\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = KOSPI), color = "pink") +

geom\_line(aes(y = BTC\_가격/4000), color = 'orange') +

scale\_y\_continuous(

name = "KOSPI axis",

sec.axis = sec\_axis(~.\*4000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs KOSPI",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

Price\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = 소비자물가지수),

color = "black",

size = 1.3) +

geom\_line(aes(y = BTC\_가격/80000),

color = 'orange') +

scale\_y\_continuous(

name = "Price index axis",

sec.axis = sec\_axis(~. \*80000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs 소비자물가지수",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

circulation\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = BTC\_공급량),

color = "gray") +

geom\_line(aes(y = BTC\_가격),

color = 'orange') +

scale\_y\_continuous(

name = "BTC circulation axis",

sec.axis = sec\_axis(~., name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs BTC 공급량",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

VIX\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = VIX\_value),

color = "gray") +

geom\_line(aes(y = BTC\_가격/500000),

color = 'orange') +

scale\_y\_continuous(

name = "VIX axis",

sec.axis = sec\_axis(~.\*500000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs VIX 값",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

Nasdaq\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = Nasdaq\_종가),

color = "blue") +

geom\_line(aes(y = BTC\_가격/2000),

color = 'orange') +

scale\_y\_continuous(

name = "NASDAQ axis",

sec.axis = sec\_axis(~.\*2000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs NASDAQ",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

Vkospi\_plot <- ggplot(data = final\_frame,

aes(x = 날짜)) +

geom\_line(aes(y = VKOSPI\_종가),

color = "red") +

geom\_line(aes(y = BTC\_가격/800000),

color = 'orange') +

scale\_y\_continuous(

name = "VKOSPI axis",

sec.axis = sec\_axis(~.\*800000, name="BTC axis")

) +

theme\_classic() +

labs(title = "BTC 가격 vs VKOSPI",

x = "날짜") +

theme(plot.title = element\_text(hjust = 0.5))

Vkospi\_plot

grid.arrange(USD\_plot, Gold\_plot, Bond\_plot,

Kospi\_plot, Price\_plot, circulation\_plot,

VIX\_plot, Nasdaq\_plot, Vkospi\_plot, nrow= 3)

grid.arrange(USD\_plot, Gold\_plot, nrow= 1)

grid.arrange( Bond\_plot, Kospi\_plot, nrow= 1)

grid.arrange(Price\_plot, circulation\_plot, nrow= 1)

grid.arrange(VIX\_plot, Nasdaq\_plot, nrow= 3)

## 2. 설명변수간 상관관계 확인 --> 상관관계가 크다면 다중회귀분석을 사용하기 곤란하다. 만약 작다면 사용해도 무방하므로 사용해보겠다는 flow

# 피어슨 상관분석 --> 설명변수 간 선형상관관계도 작지만, 설명변수와 종속변수 간 선형상관관계도 너무 작음. 이거 한 후에 비선형 상관관계도 고려해보자.

data\_cor <- cor(final\_frame[,-1])

data\_cor[,-1]

corrplot(data\_cor, method = 'number', type = "upper")

# 스피어만 상관분석 --> 설명변수와 종속변수 간 비선형상관관계도 너무 작음. 설명변수 추가해야하나.

spearman\_cor <- cor(final\_frame[, -1], method = 'spearman')

corrplot(spearman\_cor, method = 'number', type = 'upper')

# 상관계수 검정. 영가설: 상관계수가 없다.

cor\_pairwise\_test <- rcorr(data\_cor,

type = 'spearman')

cor\_pairwise\_test[[3]]

corrplot(cor\_pairwise\_test[[3]], method = 'number', type = 'upper')