A Study on Korea Box office Prediction through Data Crawling and Regression Analysis of NAVER MOVIE site

BuBigata

2018년 8월 8일

Project title :

네이버 영화 사이트 주제별 크롤링 및 회귀분석을 통한 영화 흥행도 예측

```
## Warning: package 'car' was built under R version 3.4.4
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 3.4.4
```

Contents

1. 연구 목적(Purpose of Study)

본 연구는 2010년부터 2018년 상반기까지의 개봉된 한국 영화의 데이터 분석을 바탕으로
영화의 흥행 정도 를 예측해보고자 한다. 연구 대상은 위키피디아에서 2010년 1월 1일부터 2015년 12월 31일까지 개봉한 한국 영화 제목 목록으로 선정하였다. 획득한 총 849개의 영화 제목 데이터를 바탕으로 국내 최대 포털 사이트인 네이버(Naver)의 메인 홈페이지(https://www.naver.com)에서의 검색과 네이버 영화 사이트(https://movie.naver.com)에서 각 영화의 기본 정보를 R에서의 for문을 통해 스크래핑(Scraping)하였다. 이들의 과거 8년 6개월치의 누적된 데이터를 분석하여 미래에 개봉할 영화들의 흥행 정도를 예측해보았다. 본 연구의 목적은 실제 영화의 흥행 성적과 예측한 값과의 차이를 최소화하는 데에 있다.

본 연구는 예측의 정확도 높이기 위해 *Naver Movie* 에서 7가지의 변수를 찾아내었다. 스크래핑(Scraping)한 4개의 변수와 새로이 만들어낸 3가지의 변수를 사용하여 실제 관객 수(performance)의 상관도를 분석하였다.

본 연구에서 사용한 방법은 **회귀분석(Regression Analysis)**이며 이를 통해 결정계수(R^값)를 높이고자 하였다.

2. EDA (Exploratory Data Analysis)

(1) 영화의 기본 변수 정의 및 데이터 추출

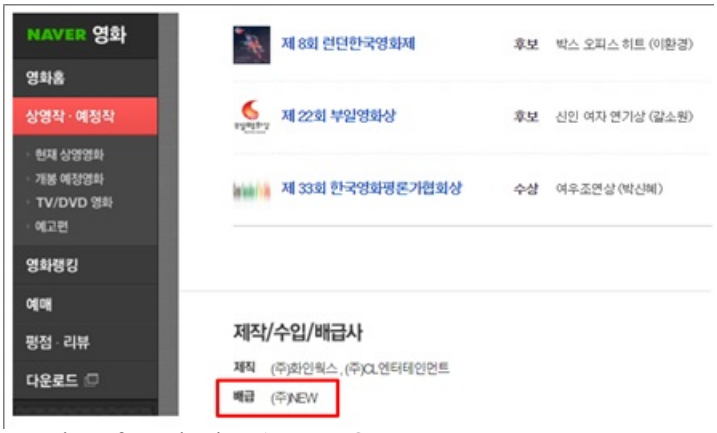
i. 변수 정의

* title : 영화 제목 * distributor : 영화 배급사 * performance : 실제 관객수(흥행 정도) * genre : 장르(SF, 공포/스릴러, 다큐멘터리, 드라마, 멜로/로맨스, 뮤지컬, 범죄, 애니메이션, 액션, 코미디) * release_time : 개봉일 * time : 상영시간 * rat_mpaa : 상영등급(12세 관람가, 15세 관람가, 전체 관람가, 청소년 관람불가) * director : 감독 * contents : 내용

ii. 데이터 전처리(Preprocess of Data)

데이터 획득 화면

<Movies Information in NAVER



<Movies Information in NAVER MOVIE

csv 형태로 정리한 파일은 다음과 같다.

	A	B	C	D	E	F	G	H	I
1		title	distributor	performanc	genre	release_time	rat_mpaa	director	contents
2	1	카페 서울	나이너스언	353	드라마	2010-01-02	12세 관람	타케 마사히	본인 프리랜서 르포라이터 '준(사이토 타쿠미 분
3	2	용서는 없다	시네마서비	1125154	느와르	2010-01-07	청소년 관람	김형준	과학수사대 최고의 실력파 부검의 강민호 교수(
4	3	소규모 야	이모션 픽처	791	다큐멘터리	2010-01-14	12세 관람	민환기	친근한 노랫말과 서정적이고 포근한 멜로디의 음
5	4	아빠가 여기	(주)쇼박스	174333	멜로/로맨스	2010-01-14	12세 관람	이광재	29년 핸섬했던(?) 과거 싹~ 고친 미모의 포토그
6	5	웨딩드레스	싸이더스	143458	드라마	2010-01-14	전체 관람	권형진	미안해, 우리 아가.. 엄마가 먼저 가서, 너무 미안
7	6	페어 러브	CJ 엔터테인	27083	드라마	2010-01-14	12세 관람	신연식	오십이 넘도록 연애 한번 못해본 형만은 친구에게
8	7	회복	스토리셋	155281	다큐멘터리	2010-01-14	12세 관람	김종철	과격파 유대교 청년단체가 보낸 폭탄소포 사건으
9	8	주유소 습격	시네마서비	729834	코미디	2010-01-21	15세 관람	김상진	노마크에게 무참히 주유소를 털린 지 언 10년, 그
10	9	사사건건	KT&G 상상	1143	드라마	2010-01-21	청소년 관람	이정욱	보이지 않아도 아름다운 세상 &lt산책가&gt. 시각
11	10	하모니	CJ 엔터테인	3018154	드라마	2010-01-28	12세 관람	강대규	{형형법상 여성수용자가 교정시설에서 출산할 경
12	11	의형제	(주)쇼박스	5416829	액션	2010-02-04	15세 관람	장훈	6년 전. 서울 한복판에서 일어난 의문의 총격전.
13	12	식객: 김치	롯데엔터테	462714	드라마	2010-01-28	전체 관람	백동훈	대령숙수의 칼을 얻은 후 스포트라이트를 뒤로 한
14	13	이웃집 좀비	인디스토리	2807	공포	2010-02-18	15세 관람	류훈	좀비 바이러스를 소재로한 여섯가지 내용의 옴니
15	14	채식주의자	스폰지	3536	드라마	2010-02-18	청소년 관람	임우성	꽃이, 나무가 되고 싶었던 그녀... 채식주의자 영
16	15	평행 이론	CJ 엔터테인	921111	공포	2010-02-18	15세 관람	권호영	최연소 부장판사로 출세가도를 달리던 석현(지진
17	16	행복한 울릉	마운틴 픽처	1068	다큐멘터리	2010-02-25	전체 관람	황석호	울릉도에서 태어나고 자라서 이제는 할아버지가
18	17	회오리 바람	모쿠슈라	2275	드라마	2010-02-25	15세 관람	장건재	고2 겨울방학, 태훈은 미정과 함께 연애 100일 기
19	18	경계도시 2	시네마 달	9448	다큐멘터리	2010-03-18	15세 관람	홍형숙	2003년, 재독철학자 송두율 교수는 체포영장이
20	19	무법자	(주)NEW	160734	느와르	2010-03-18	청소년 관람	김철한	아무 이유 없이 잔인하게 죽어 간 시체들을 마주
21	20	육혈포 강도	전망좋은영	1214237	느와르	2010-03-18	15세 관람	강효진	8년간 힘들게 모은 하와이 여행자금을 은행 강도
22	21	이웃집 남자	(주)루믹스	1188	드라마	2010-03-18	청소년 관람	장동홍	이웃집 남자 상수. 나의 꿈은 땅과 벤츠다. 올빽
23	22	불타는 내	CJ CGV	9089	멜로/로맨스	2010-03-25	15세 관람	최원섭	커피 전문점에서 만난 운명 같은 그녀. 병열(최도
24	23	비밀애	시너지	164252	멜로/로맨스	2010-03-25	청소년 관람	류훈	결혼 2개월만에 불의의 사고로 혼수상태에 빠진
25	24	아마존의 눈	마운틴픽처	104367	다큐멘터리	2010-03-25	15세 관람	김진만	9개월의 사전 조사와 250일의 제작 기간, 제작비
26	25	소명 2 - 도	에스피엠	29668	다큐멘터리	2010-04-01	전체 관람	신현원	축구 신동 강성민 선교사, 어렸을 때부터 축구에

iii. 데이터 추출시 사용 코드

```
for(i in 1:length(movieslist)){
  remdir$navigate("https://www.naver.com/")
  sbox <- remdir$findElement(using = 'xpath','//*[@id="query"]')

  sbox$sendKeysToElement(list(movielist[i], key = "enter"))
  sbox<-NA

  try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")}, silent = T)
  if(is.na(sbox)){
    sbox <- remdir$findElement(using = 'css selector',"strong.info_title a")
    sbox$clickElement()
  }
  if(is.na(sbox)){next}

  url<-remdir$getCurrentUrl()[[1]]
  webpage<-read_html(url)

  newperformance<-webpage %>%
    html_node(xpath = '//*[@id="_au_movie_info"]/div[2]/dl[2]/dd[4]/span') %>%
    html_text(trim = T)

  newperformance <- ifelse(is.na(newperformance),
                           webpage %>%
                             html_node(xpath ='//*[@id="_au_movie_info"]/div[2]/dl[2]/dd[3]/span') %>%
                             html_text(trim = T),
                           newperformance)
  newperformance <- ifelse(is.na(newperformance),
                           webpage %>%
                             html_node(xpath = '//*[@id="_au_movie_info"]/div[2]/dl/dd[4]/span') %>%
                             html_text(trim = T),
                           newperformance)
  newperformance <- ifelse(is.na(newperformance),
                           webpage %>%
                             html_node(xpath = '//*[@id="_au_movie_info"]/div[2]/dl/dd[3]/span') %>%
                             html_text(trim = T),
                           newperformance)

  newrelease_date <- webpage %>%
    html_node(xpath = '//*[@id="dss_h_movie_info_opendate_content"]') %>%
    html_text(trim = T)

  newdirector <- webpage %>%
    html_node(xpath = '//*[@id="dss_h_movie_info_director_content"]/a') %>%
```

```
    html_text(trim = T)

  newmpaa_rating <- webpage %>%
    html_node(xpath = '//*[@id="dss_h_movie_info_grade_content"]') %>%
    html_text(trim = T)
  newmpaa_rating <- ifelse(is.na(newmpaa_rating),"청소년 관람불가",newmpaa_rating)

  sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")
  remdir$executeScript("arguments[0].setAttribute('target', arguments[1]);", list(sbox, ""))
  sbox$clickElement()

  url <- remdir$getCurrentUrl()[[1]]
  webpage <- read_html(url)

  newgenre <- webpage %>%
    html_node(xpath = '//*[@id="content"]/div[1]/div[2]/div[1]/dl/dd[1]/p/span[1]/a') %>%
    html_text(trim = T)

  newtime <- webpage %>%
    html_node(xpath = '//*[@id="content"]/div[1]/div[2]/div[1]/dl/dd[1]/p/span[3]') %>%
    html_text(trim = T)


  newcontents <- (webpage %>% html_node(".con_tx") %>% html_text())[[1]][1]

  newtitle <- movieslist[i]
  newlabel <- i

  sbox <- remdir$findElement(using = 'css selector',".tab05_off")
  sbox$clickElement()

  sbox <- remdir$findElement(using = 'id',"beforePointTab")
  sbox$clickElement()

  url <- remdir$getCurrentUrl()[[1]]
  webpage <- read_html(url)
  newinterest_before <- (webpage %>%
                           html_nodes(xpath = '//*[@id="beforePointArea"]/div[2]/span/em') %>%
                           html_text())[1]

  newratings_before <- webpage %>%
    html_node(xpath = '//*[@id="beforePointArea"]/div[2]/div') %>%
    html_text(trim = T)

  newratings_before <- strsplit(x=newratings_before,split = "중")[[1]][2]

  sbox <- remdir$findElement(using = 'css selector',".tab02_off")
  sbox$clickElement()

  url <- remdir$getCurrentUrl()[[1]]
  webpage <- read_html(url)
  newdistributor <- (webpage %>% html_nodes('.agency_name dd') %>% html_text(trim=T))[2]
  newdistributor <- ifelse(is.na(newdistributor), (webpage %>% html_nodes('.agency_name dd') %>% html_text(t
rim = T)), newdistributor)

  title <- c(title, newtitle)
  distributor <- c(distributor, newdistributor)
  performance <- c(performance, newperformance)
  int_bef <- c(interest_before, newinterest_before)
  rat_bef <- c(ratings_before, newratings_before)
  genre <- c(genre, newgenre)
  release_date <- c(release_date, newrelease_date)

  time <- c(time, newtime)
  mpaa_rating <- c(mpaa_rating, newmpaa_rating)
  director <- c(director, newdirector)
  contents <- c(contents, newcontents)
  label <- c(label, newlabel)
  print(paste(i, newtitle, newdistributor, newratings_before, newperformance, newrelease_date, newdirector,
newmpaa_rating, newinterest_before))
}
```

(2) 영화 사이트 내에서의 필요 요소 변수 정의 및 데이터 추출

i. 변수 정의

최초로 scraping한 변수

- int_bef : 해당 영화에 작성된 개봉 전 댓글 개수
- rat_bef : 해당 영화의 개봉 전 평점
- num_bef : 감독이 해당 영화를 제작하기 전에 제작한 영화의 개수

함수 사용하여 얻어낸 변수

- rel_int : 해당 영화의 관객 수(performance)를 월별 총 관객 수로 나눈 값(변수변환)
- per_dis : 2005-2009년 해당 배급사의 관객수의 평균
- per_month : 2005-2009 해당 월의 관객수의 평균

for문을 이용하여 scraping한 변수

- po_dir : 감독의 역량(해당 영화의 감독의 전작들의 총 관객수를 전작들의 수로 나눈, 평균을 구한 값)
- num_staff : 영화에 참여한 제작진의 수
- num_vid : 해당 영화 사이트에 업로드된 동영상 수(예고편과 메이킹 영상 수. 마케팅 투자 정도를 확인)
- num_mactor : 해당 영화의 주연 수(영화 페이지에서 주연으로 쓰여 있는 개수)

아래 두 개의 변수는 위의 per_dis와 performance와의 상관도가 유의한 결과가 나와서 새로이 추가한 변수들이다.

- sd_dis : 각 배급사에 해당하는 관객수의 표준편차(standard deviation)
- skew_dis : 각 배급사에 해당하는 관객수의 왜도(skewness)

ii. 데이터 전처리(Preprocess of Data)

데이터 획득 화면

csv 형태로 정리한 파일은 다음과 같다.

int_bef	rat_bef	rel_int	per_dis	num_bef	per_month	po_dir	num_staff	num_vid	num_mactor
35	9.74	0.038546	0	0	1173951	0	41	1	2
143	8.69	0.072259	0	2	1173951	300529	304	9	3
4	7.75	0.001515	0	0	1173951	0	18	1	7
149	8.93	0.056439	1	4	1173951	342700.3	275	6	3
196	9.12	0.074242	0	1	1173951	462623	261	8	2
100	9.35	0.037879	1	0	1173951	0	106	4	2
22	8.32	0.008333	0	0	1173951	0	54	1	1
258	7.5	0.097727	0	4	1173951	3907785	402	9	4
1		0.000379	0	0	1173951	0	148	0	8
801	9.56	0.275163	1	3	1173951	4206611	419	7	7
533	9.19	0.185585	1	2	749421.4	691342	408	10	2
270	8.59	0.092752	1	0	1173951	0	372	15	3
108	8.74	0.05042	0	0	749421.4	0	40	7	10
23	9.09	0.010738	0	0	749421.4	0	153	4	5
175	8.49	0.081699	1	1	749421.4	31738	380	2	1
10	9	0.007704	0	0	749421.4	0	3	4	2
94	9.06	0.072419	0	0	749421.4	0	156	2	2
15	7.6	0.01251	0	0	296376.2	0	11	9	2
76	7.72	0.063386	0	0	296376.2	0	250	6	3
131	8.04	0.109258	0	1	296376.2	37518	318	4	3
5	8.2	0.00417	0	0	296376.2	0	117	1	5
8	10	0.006098	1	0	296376.2	0	95	1	2
53	9	0.040396	0	1	296376.2	1799447	333	5	2
203	9.67	0.154726	0	0	296376.2	0	76	7	1
21	9.57	0.012743	0	0	324912.6	0	15	3	6

iii. 데이터 추출 시 사용 화면 및 코드 2

① po_dir(감독의 역량)

해당 영화를 만든 감독이 이 영화 전에 제작한 영화가 평균적으로 몇 명의 관객수를 동원 했는지를 파악하기 위해 추출하였다.

데이터 획득 화면


```
# po_dir
beforeperformance<-c()
beforenum<-c()
index<-c()
for(i in 1:length(title))
{

  remdir$navigate("https://www.naver.com/")

  sbox <- remdir$findElement(using = 'css selector',".input_text")
  sbox$sendKeysToElement(list(title[i], key="enter"))

  sbox<-NA
  try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")})

  if(is.na(sbox))
  {
    sbox<-remdir$findElement(using="css selector","strong.info_title a")
    sbox$clickElement()
  }

  url<-remdir$getCurrentUrl()[[1]]
  webpage<-read_html(url)
  date1<-gsub(webpage %>% html_nodes("#dss_h_movie_info_opendate_content") %>% html_text(),pattern = " 개봉"
,replacement = "")
  date1<-gsub(date1,pattern = "\\.",replacement = "/")
  date1<-as.Date(date1)

  # 이건 영화의 제목 링크를 저장
  sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")
  remdir$executeScript("arguments[0].setAttribute('target', arguments[1]);", list(sbox, ""))
  sbox$clickElement()

  sbox<-remdir$findElement(using="xpath",'//*[@id="content"]/div[1]/div[2]/div[1]/dl/dd[2]/p/a')
```

```
  sbox$clickElement() ##sbox라고 지정한 영화이름의 링크를 클릭



  sbox<-remdir$findElement(using="css selector",'.more')
  sbox$clickElement()

  library(stringr)
  url<-remdir$getCurrentUrl()[[1]]
  direc.num <- as.numeric(str_extract(url, "\\-*\\d+\\.*\\d*"))


  url <- paste0("https://movie.naver.com/movie/bi/pi/filmoMission.nhn?peopleCode=", direc.num, "&year=0&tota
lCount=20")

  webpage <- read_html(url)
  a <- webpage %>% html_nodes('.pilmo_tit a') %>% html_text()

  ind<-which(  gsub(a,pattern = " ",replacement = "") == gsub(title[i],pattern = " ",replacement = "") )



  last<-length(a)
  a <- a[ind+1:last]
  a <- na.omit(a)
  a<-as.vector(a)
  b<- webpage %>% html_nodes('.pilmo_genre a') %>% html_text()
  b <- as.numeric(str_extract(b, "\\-*\\d+\\.*\\d*"))
  b <-na.omit(b)
  b <- b[ind+1:length(b)]
  b<-na.omit(b)
  b<-as.vector(b)
  if(length(b)!=length(a)) a<-a[-length(a)]

  if(length(a)==0)
  {
    newbeforeperformance<-0
    newbeforenum<-0
  }

  else{
    getper<-c()
    for(j in 1:length(a))
    {
      remdir$navigate("https://www.naver.com/")
      sbox <- remdir$findElement(using = 'css selector',".input_text")
      sbox$sendKeysToElement(list(a[j], key="enter"))

       try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")},silent=T)

        if(is.na(sbox)){
                            sbox <- remdir$findElement(using = 'css selector',"strong.info_title a")
                             sbox$clickElement()
                        }

      url<-remdir$getCurrentUrl()[[1]]
      webpage<-read_html(url)
      newgetper<-webpage %>%
           html_node(xpath='//*[@id="_au_movie_info"]/div[2]/dl[2]/dd[4]/span') %>%
           html_text(trim=T)

      newgetper<-ifelse(is.na(newgetper),
                webpage %>%
                       html_node(xpath='//*[@id="_au_movie_info"]/div[2]/dl[2]/dd[3]/span') %>%
                       html_text(trim=T),
                       newgetper)

      newgetper<-ifelse(is.na(newgetper),
                         webpage %>%
                           html_node(xpath='//*[@id="_au_movie_info"]/div[2]/dl/dd[4]/span') %>%
                           html_text(trim=T),
                         newgetper)
```

```
      newgetper<-ifelse(is.na(newgetper),
                        webpage %>%
                          html_node(xpath='//*[@id="_au_movie_info"]/div[2]/dl/dd[3]/span') %>%
                          html_text(trim=T),
                        newgetper)
      newgetper<-gsub(pattern = ",",replacement = "",x=newgetper)
      newgetper <- as.numeric(str_extract(newgetper, "\\-*\\d+\\.*\\d*"))



      date2<-NA
      try({date2<-gsub(webpage %>% html_nodes("#dss_h_movie_info_opendate_content") %>% html_text(),pattern
= " 개봉",replacement = "")})

      if((length(date2)==0))
      {
        newgetper<-NA
      }
      else{


        date2<-as.numeric(substr(date2,1,4))

        if(date2!=b[j])
        {
          newgetper<-NA
        }
      }

      getper<-c(getper,newgetper)
    }
    newbeforeperformance<-mean(getper,na.rm=T)
    newbeforenum<-length(na.omit(getper))
  }

  beforenum<-c(beforenum,newbeforenum)
  beforeperformance<-c(beforeperformance,newbeforeperformance)
  index<-c(index,i)

  print(paste(newbeforenum,newbeforeperformance,title[i],i,sep = "       "))
}
```

② num_staff(제작진 수)

데이터 획득 화면

crawling 과정에서 영역을 지정한 후 해당 영역 안의 이름에 있는 하이퍼링크의 개수를 추출하였다.

```
# num_staff(for문)
num_staff <- c()

for(i in 1:length(title)){

  remdir$navigate("https://www.naver.com/")

  sbox <- remdir$findElement(using = 'css selector',".input_text")
  sbox$sendKeysToElement(list(title[i], key="enter"))

  sbox<-NA
  try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")})

  if(is.na(sbox))
  {
    sbox<-remdir$findElement(using="css selector","strong.info_title a")
    sbox$clickElement()
  }

  sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")
  remdir$executeScript("arguments[0].setAttribute('target', arguments[1]);", list(sbox, ""))
  sbox$clickElement()

  sbox <- remdir$findElement(using = 'xpath', '//*[@id="movieEndTabMenu"]/li[2]/a')
  sbox$clickElement()

  sboxmore <- remdir$findElement(using = 'xpath', '//*[@id="staffMore"]')
  sboxmore$clickElement()

  url<-remdir$getCurrentUrl()[[1]]
  webpage<-read_html(url)

  hlink_data <- html_nodes(webpage,'td > span > a') %>% html_attr('href')
  hlink_data <- na.omit(hlink_data)
  a <- length(hlink_data)
  num_staffs<-c(num_staff,a)
  print(paste(i,"번째 스텝 수", a))
}
```

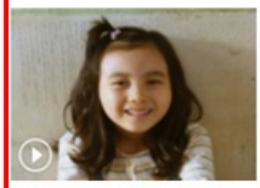
③ num_vid(동영상 수)

데이터 획득 화면

주요정보 | 배우/제작진 | 포토 | 동영상 | 평점 | 리뷰 | 명대사/연관영화

동영상

예고편 (3)



'예승' 버전 특별 예고편
2013.02.13

HD 본 예고편
2013.01.18

티저 예고편
2013.01.07

메이킹 (8)

HD 본편 공개 영상
2013.03.22

HD 천만돌파 감사인사 …
2013.02.26

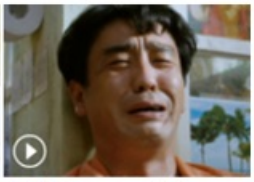
HD 미공개 편집 영상
2013.02.20

무대인사 후기 영상
2013.02.01

MP 추천 영상
2013.01.25

류승룡의 자신만만 추천…
2013.01.24

X-파일 영상
2013.01.11



'흥행킹' 류승룡 영상
2013.01.07

```
# num_vid(for문)

## 이것은 해당 영화의 올라온 예고편 개수를 가리킨다.
numvid <- c()
for(i in 1:length(title)){

  remdir$navigate("https://www.naver.com/")

  sbox <- remdir$findElement(using = 'css selector',".input_text")
  sbox$sendKeysToElement(list(title[i], key="enter"))

  sbox <- NA
  try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")})

  if(is.na(sbox)){
    sbox<-remdir$findElement(using="css selector","strong.info_title a")
    sbox$clickElement()
  }

  sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")
  remdir$executeScript("arguments[0].setAttribute('target', arguments[1]);", list(sbox, ""))
  sbox$clickElement()

  sbox <- remdir$findElement(using = 'xpath', '//*[@id="movieEndTabMenu"]/li[4]/a') # '동영상' 버튼 위치 저장
  sbox$clickElement()


  pre.url <- remdir$getCurrentUrl()[[1]]
  pre <- read_html(pre.url)
  b <- pre %>% html_node(xpath = '//*[@id="content"]/div[1]/div[4]/div/div[2]/div[1]/div/div/span/em') %>% h
tml_text() # 예고편 수
  c <- pre %>% html_node(xpath = '//*[@id="content"]/div[1]/div[4]/div/div[2]/div[2]/div/div/span/em') %>% h
tml_text() # 메이킹 수

  b <- ifelse(is.na(b), 0, b)
  c <- ifelse(is.na(c), 0, c)

  numvid <- c(numvid,as.numeric(b) + as.numeric(c))
  print(paste(as.numeric(b) + as.numeric(c),i,sep="      "))
}
```

④ num_mactor(주연배우 수)

데이터 획득 화면

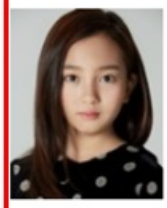
배우

류승룡
주연 | 용구 역

극한직업,2018
7년의 밤,2018

박신혜
주연 | 큰 예승 역

침묵,2017
형,2016



갈소원
주연 | 어린 예승 역

미스터 주(가제),2018
물물교환,2017

오달수
주연 | 소양호 역

이웃사촌,2017
니 부모 얼굴이 보고 싶다,2017

박원상
주연 | 최춘호 역

리틀 포레스트,2018
환절기,2018

김정태
주연 | 강만범 역

역적 : 백성을 훔친 도적,2017
씨클리드,2016

정만식
주연 | 신봉식 역

창궐,2018
조작,2017

김기천
주연 | 서노인 역

두개의 빛: 릴루미노,2017
염력,2017

박길수
조연 | 정교도관 역

당신, 거기 있어줄래요,2016
사랑은 없다,2016

조재윤
조연 | 김교도관 역

중2라도 괜찮아,2017
더 폰션,2017

```
# num_mactor(for문)
numactor <- list()

for(i in 1:length(title)){
  remdir$navigate("https://www.naver.com/")

  sbox <- remdir$findElement(using = 'css selector',".input_text")
  sbox$sendKeysToElement(list(title[i], key="enter"))

  sbox <- NA
  try({sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")})

  if(is.na(sbox)){
    sbox<-remdir$findElement(using="css selector","strong.info_title a")
    sbox$clickElement()
  }

  sbox <- remdir$findElement(using = 'css selector',".sh_movie_link")
  remdir$executeScript("arguments[0].setAttribute('target', arguments[1]);", list(sbox, ""))
  sbox$clickElement()

  sbox <- remdir$findElement(using = 'xpath', '//*[@id="movieEndTabMenu"]/li[2]/a')
  sbox$clickElement()

  a.url <- remdir$getCurrentUrl()[[1]]
  act <- read_html(a.url)

  d <- act %>% html_nodes('.in_prt > em') %>% html_text()
  num_main_actor <- sum(as.numeric(d == "주연"))

  numactor[[length(numactor)+1]] <- num_main_actor
  print(paste(i,"번째 주연배우 수  = ", num_main_actor))
}
```

ii. 표본 개수

위키피디아(Wikipedia)에서 얻은 한국 영화 : 849개

iii. 독립변수(x)와 종속변수(y) 설정

종속변수(Y) = performance
독립변수(X) = rel_int, per_month, per_dis, skew_dis, sd_dis, time, num_vid

3. 첫 번째 크롤링 이후

```
## Warning: package 'corrplot' was built under R version 3.4.4
```

```
## corrplot 0.84 loaded
```

```
## Warning: package 'ggplot2' was built under R version 3.4.4
```

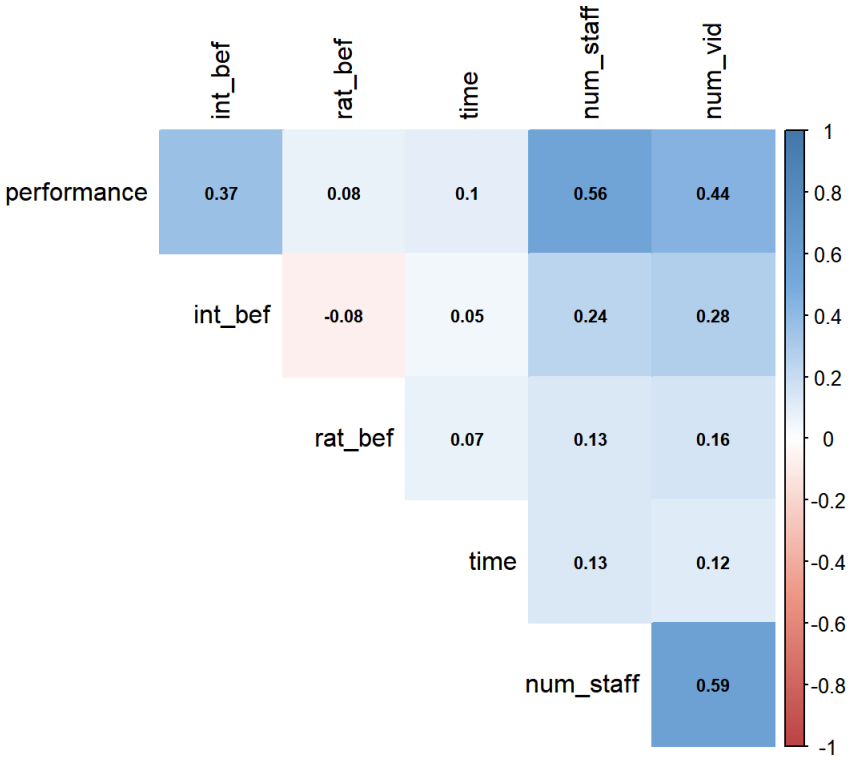
```
## Warning: package 'dplyr' was built under R version 3.4.4
```

```
##
## Attaching package: 'dplyr'
```

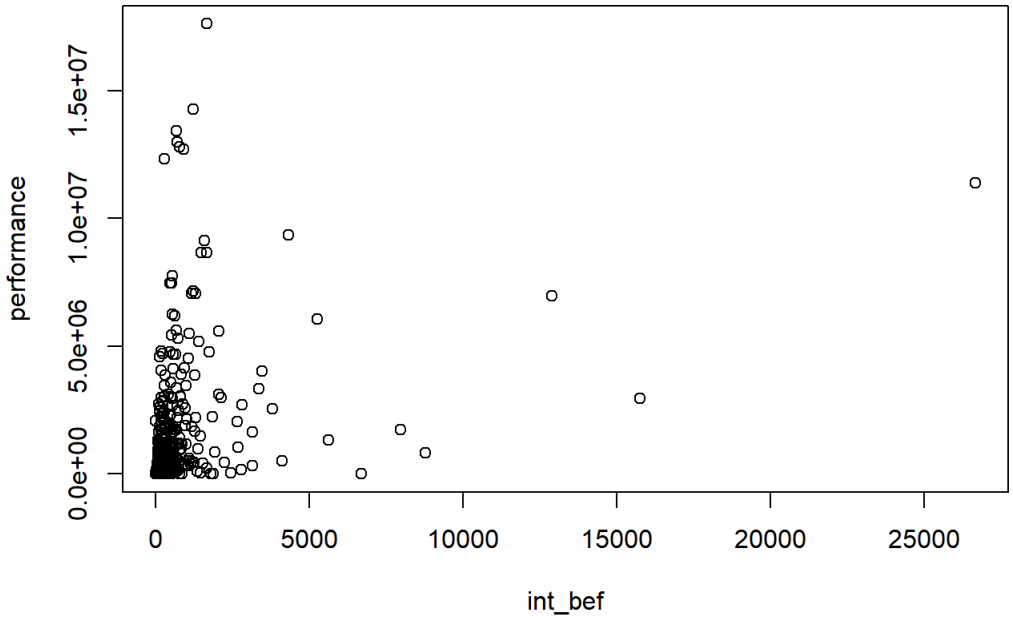
```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

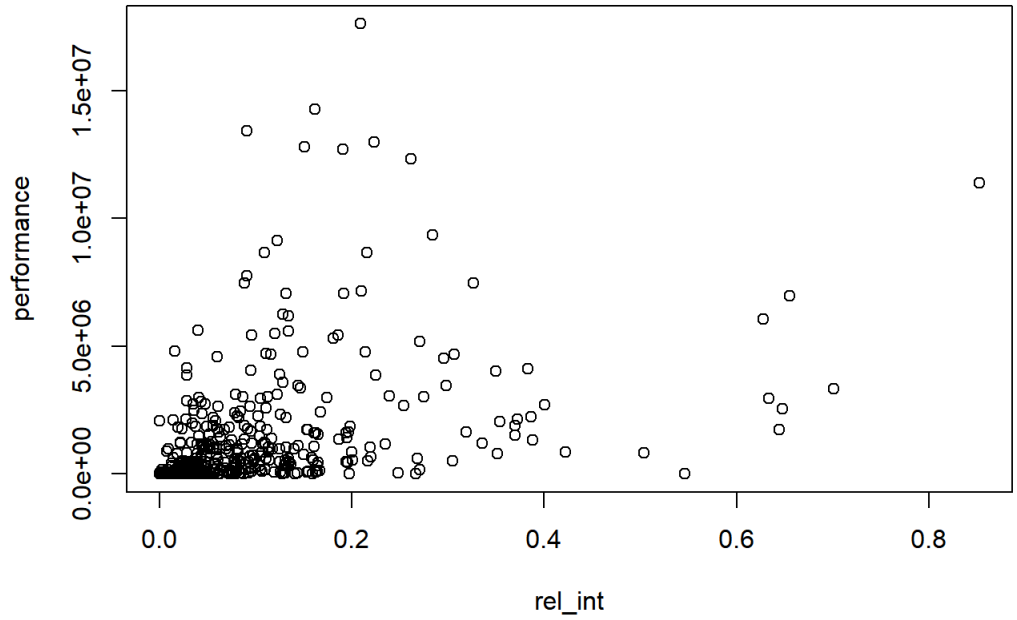


1. int_bef의 변환




```
## [1] 0.368126
```

```
movies$rel_int<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  time<-movies$release_time[i]

  a<-sum(movies[abs(as.numeric(movies$release_time-time))<=25,]$int_bef) ##해당 영화의 개봉일과 25일 차이가 나는 영화의 int_bef를 sum한다.
  movies$rel_int[i]<-movies$int_bef[i]/a   ##이렇게 구한 sum을 int_bef에서 나눔으로써 rel_int란 변수를 생성한다
}
```

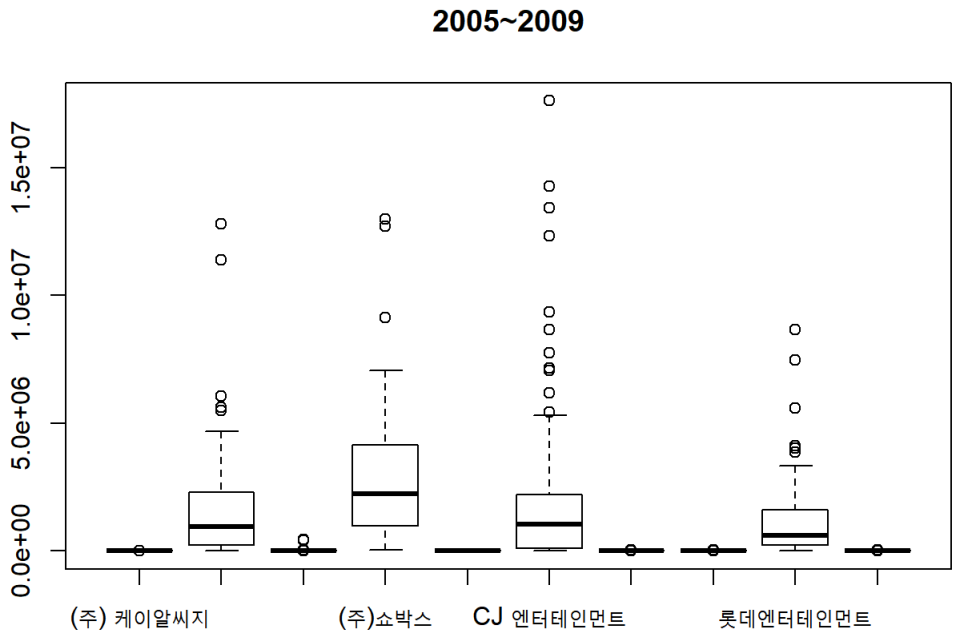



```
## [1] 0.5047767
```

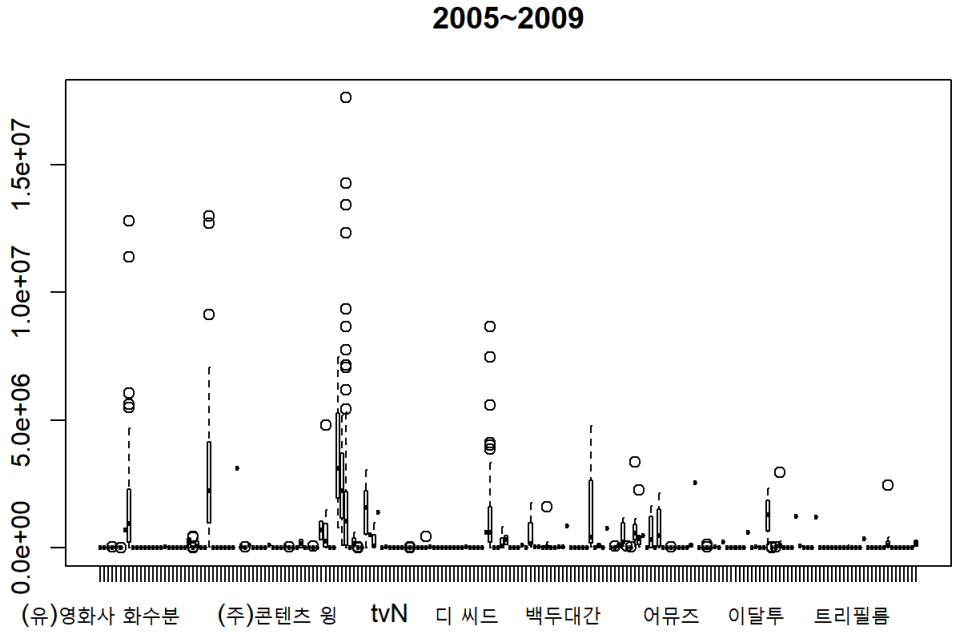
2. distributor

영화의 흥행을 예측하는데 배급사가 큰 도움을 줄 것으로 보인다.

```
t<-names(sort(table(movies1$distributor),decreasing=T))[1:10]
t1<-movies1[movies1$distributor %in% t,]
t1$distributor<-as.character(t1$distributor)
boxplot(t1$performance~t1$distributor,main="2005~2009")
```

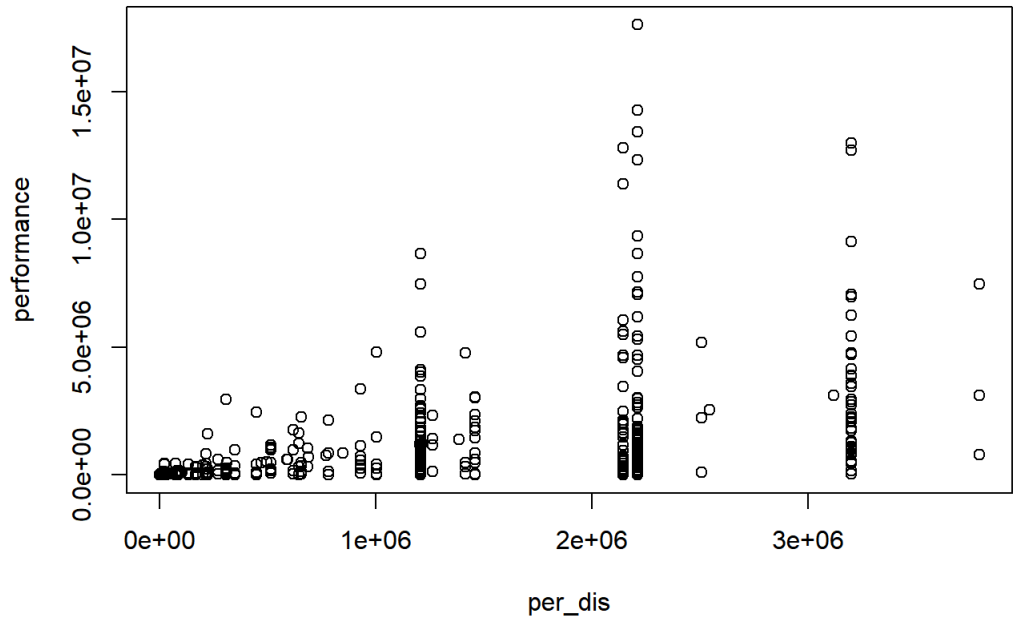



```
boxplot(movies$performance~movies$distributor,main="2005~2009")
```



해당 영화의 배급사가 2005부터 2009년까지 맡은 영화의 평균 관객수를 per_dis에 저장한다.

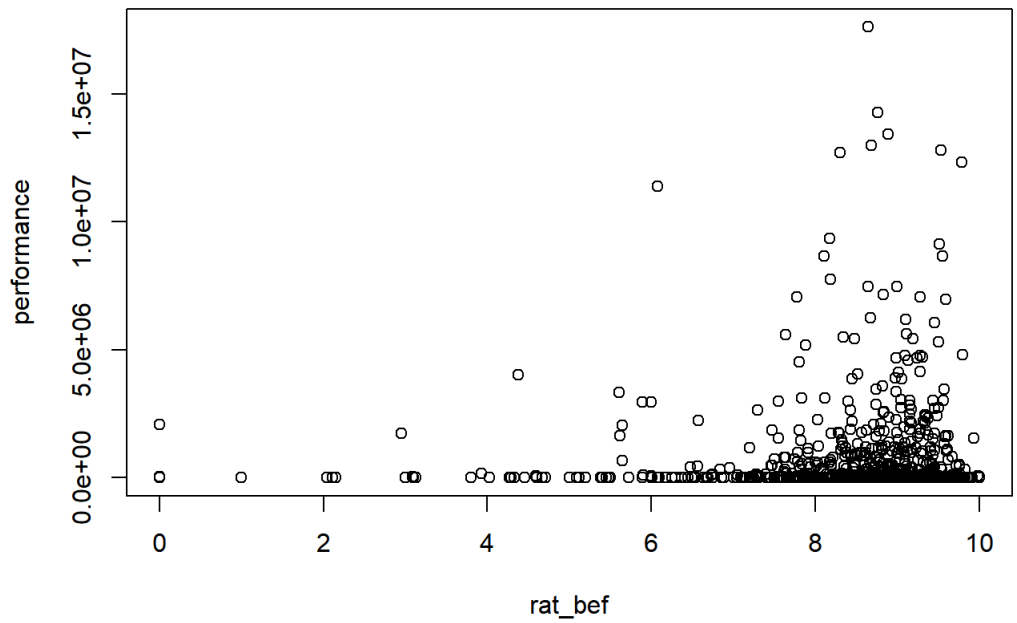
```
movies$per_dis<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  movies$per_dis[i]<-mean((movies1 %>% filter(distributor==movies$distributor[i]))$performance)
}
```




```
## [1] 0.5376806
```

3. rat_bef

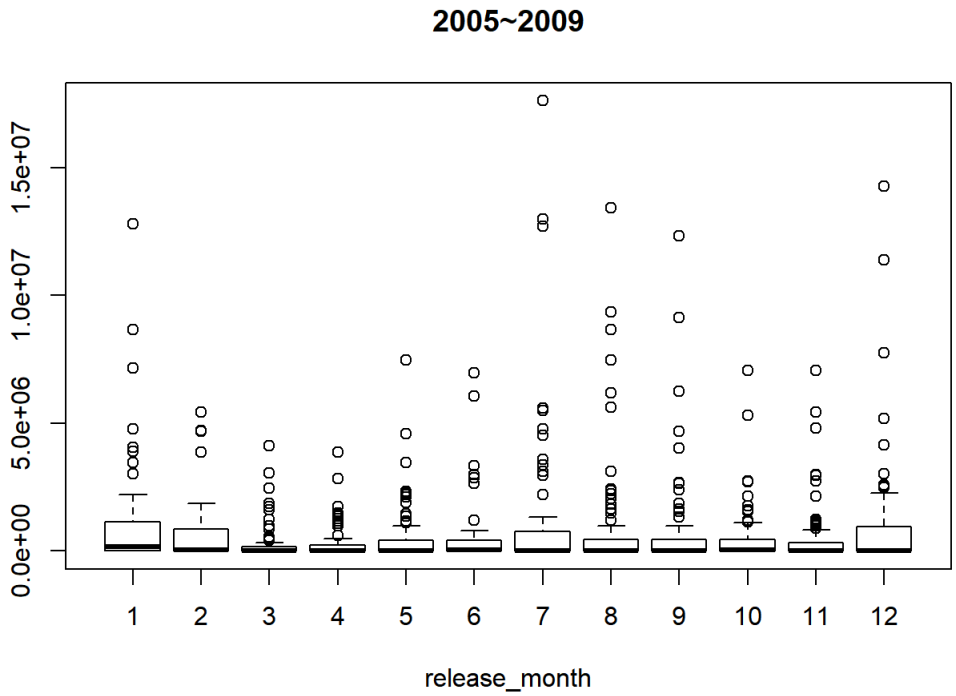
rat_bef와 performance간의 상관관계는 너무 작다 ==> rat_bef는 변수에서 제거하자.




```
## [1] 0.07561405
```

4. release_month의 변환

```
boxplot(movies1$performance~movies1$release_month,main="2005~2009",xlab="release_month")
```



12개의 달에 대해 performance가 차이가 있는지 검정하자.

```
fligner.test(performance ~ release_month, data = movies1)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  performance by release_month
## Fligner-Killeen:med chi-squared = 65.431, df = 11, p-value =
## 8.942e-10
```

등분산성 가정이 깨진다.

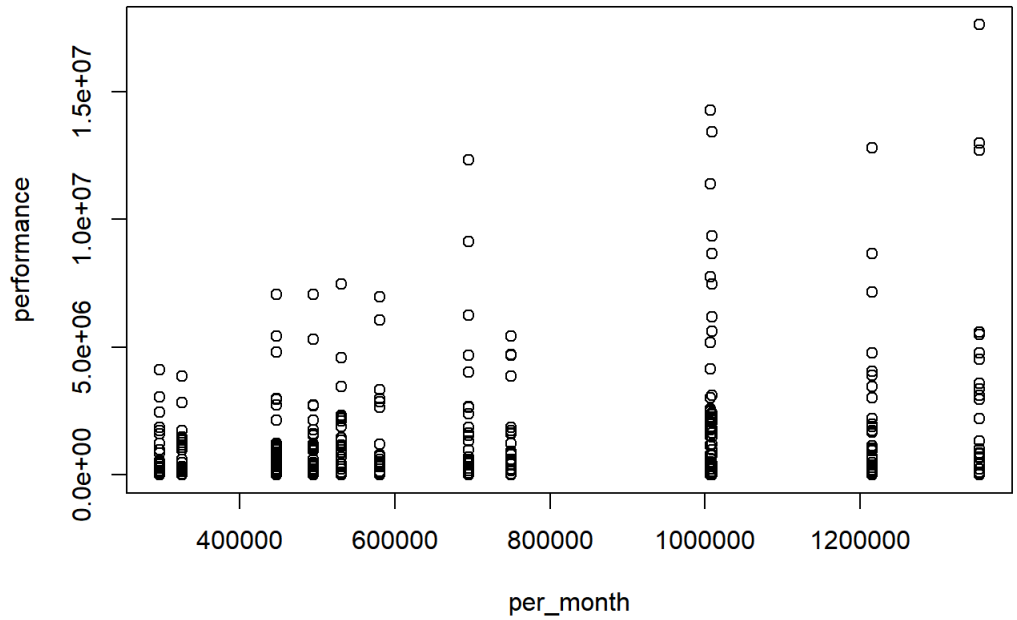
```
analysis <- oneway.test(performance ~ release_month,
                        data      = movies,
                        var.equal = FALSE)
```

p_value가 엄청 작다. ==>월별 관객수의 평균은 통계적으로 유의하게 차이가 있다고 결론

해당 영화가 개봉한 월에 2005년부터 2009년까지 개봉한 영화의 평균 관객수를 per_month에 넣는

```
movies$per_month<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  movies$per_month[i]<-mean((movies1 %>% filter(release_month==movies$release_month[i]))$performance)

}
```

```
plot(movies$per_month,movies$performance,xlab="per_month",ylab="performance")
```




```
cor(movies$per_month,movies$performance)
```

```
## [1] 0.1793425
```

5. 첫 번째 모델 적합

결정계수가 39프로밖에 안 나온다

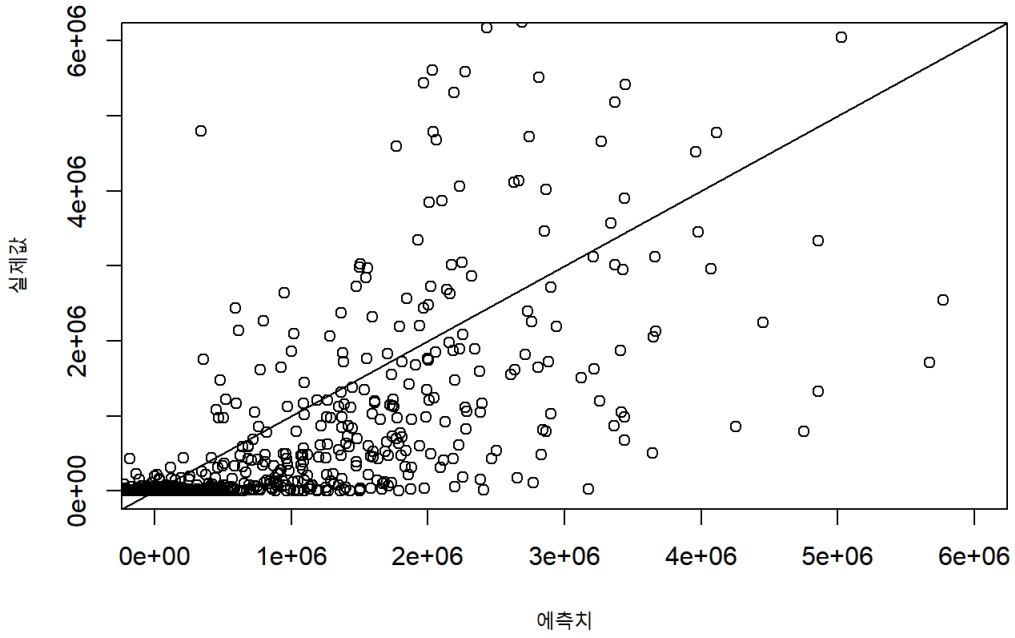
```
model1<-lm(performance~per_month+rel_int+per_dis+rat_mpaa,data=movies)

model1<-step(model1,direction="both")
```

```
## Start:  AIC=24066.14
## performance ~ per_month + rel_int + per_dis + rat_mpaa
##
##             Df  Sum of Sq        RSS   AIC
## <none>                    1.7079e+15 24066
## - rat_mpaa   3 1.5568e+13 1.7234e+15 24068
## - per_month  1 5.9146e+13 1.7670e+15 24093
## - rel_int    1 2.2741e+14 1.9353e+15 24170
## - per_dis    1 2.8173e+14 1.9896e+15 24194
```

```
summary(model1)    ##결정계수가 39프로밖에 안 나온다!! 망했다
```

```
##
## Call:
## lm(formula = performance ~ per_month + rel_int + per_dis + rat_mpaa,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3964292  -474023   -55879   305297 14198884
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -8.351e+05  1.685e+05  -4.957 8.65e-07 ***
## per_month                8.072e-01  1.495e-01   5.400 8.67e-08 ***
## rel_int                  6.295e+06  5.945e+05  10.588  < 2e-16 ***
## per_dis                  6.743e-01  5.721e-02  11.785  < 2e-16 ***
## rat_mpaa15세 관람가      3.548e+05  1.485e+05   2.389   0.0171 *
## rat_mpaa전체 관람가      3.990e+04  1.770e+05   0.225   0.8217
## rat_mpaa청소년 관람불가  1.216e+05  1.487e+05   0.818   0.4137
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1424000 on 842 degrees of freedom
## Multiple R-squared:  0.3995, Adjusted R-squared:  0.3952
## F-statistic: 93.37 on 6 and 842 DF,  p-value: < 2.2e-16
```

```
movies$pred<-predict(model1,movies)
plot(movies$pred,movies$performance,xlim=c(0,6000000),ylim=c(0,6000000),xlab="예측치",ylab="실제값")
abline(0,1)
```



예측이 잘 안 되는 영화를 한번 보자. pred는 100,0000보다 높은데(즉 100,0000보다 높게 예측됐는데) 실제론 10,0000 관객도 안 된 영화의 title을 부르자

```
(movies %>% filter(pred>1000000,performance<100000))$title
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
##  [1] 페어 러브                    폭풍전야
##  [3] 사요나라 이츠카              귀
##  [5] 요술                          된장
##  [7] 회초리                        간증
##  [9] 심도                          짐승의 끝
## [11] 집                            사랑한다, 사랑하지 않는다
## [13] 철가방 우수氏                 복숭아나무
## [15] 미운오리새끼                  I AM
## [17] 밀월도 가는 길               은실이
## [19] 태어나서 미안해               설인
## [21] 이별계약                      48미터
## [23] 뫼비우스                      천안함 프로젝트
## [25] 그 강아지 그 고양이          조난자들
## [27] 하이프 네이션 : 힙합사기꾼 메이크 유어 무브
## [29] 소리굽쇠                      정글히어로
## [31] 막걸스                        명량: 회오리 바다를 향하여
## [33] 지금은맞고그때는틀리다       나쁜 나라
## [35] 미안해 사랑해 고마워
## 843 Levels:                  개를 훔치는 완벽한 방법 ...
```

이 영화들의 공통점은 촬영에 참여한 스탭 수가 적다는 것이었다.

이 사진은 영화 페어 러브의 스텝 목록이다.

그리고 감독의 역량을 표현하는 지표도 만들자. 해당 영화를 만들기 전에 감독이 만든 영화의 평균 관객수를 뽑아내 보자.

그리고 영화 개봉 전 naver 영화에 올라와 있는 에고편 수도 좋은 정보가 될 수 있을 것이다 왜냐하면, 예고편의 수도 어떻게 보면 해당 배급사나 제작사의 홍보능력을 평가할 수 있기 때문이다.

혹시나 주연의 수도 영화의 흥행척도가 될 있지 않을까?

4. 두 번째 크롤링 이후

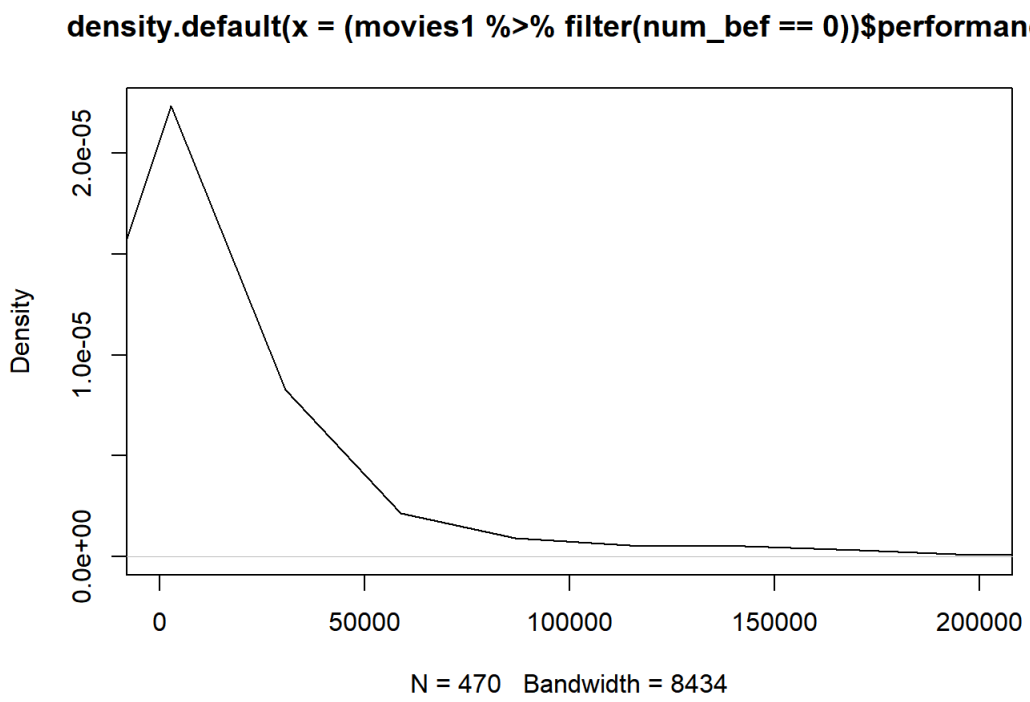
1. po_dir의 결측치 제거

해당 영화를 만들기 전에 아무 영화도 만들지 않은 신인 감독일 경우, 과거 데이터가 없어서 문제가 생긴다. 그래서 2005년부터 2009년까지, 신인감독의 성적을 참고해서 결측치를 제거했다

```
summary((movies1 %>% filter(num_bef==0))$performance)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        1      950     3662   341304    43933 14262766
```

```
plot(density((movies1 %>% filter(num_bef==0))$performance),xlim=c(0,200000))
```



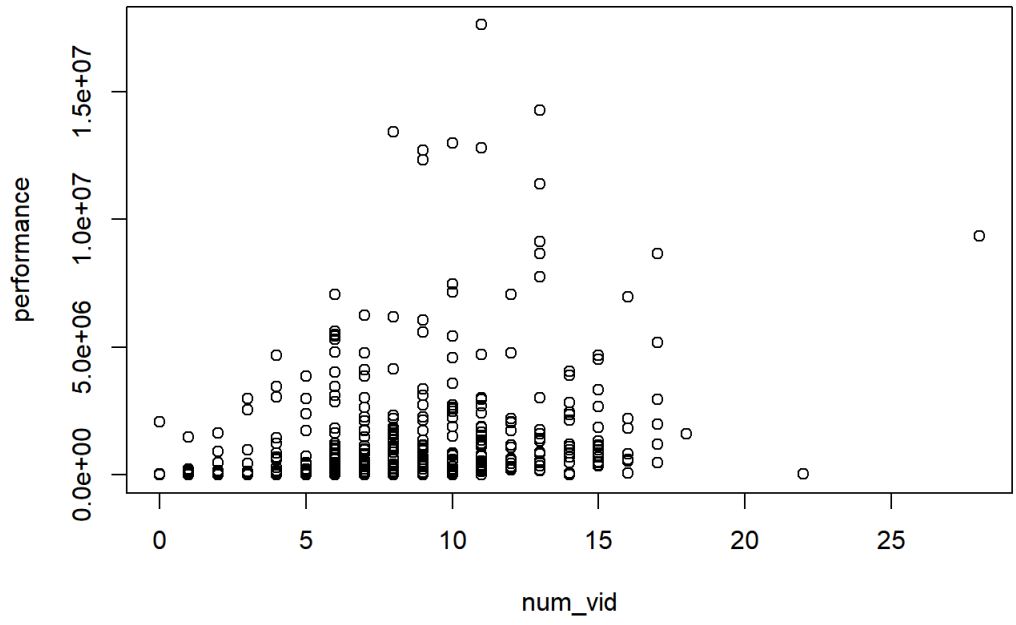
신인감독의 흥행성적의 분포

가 심하게 right skewed되어 있다. 평균보다, 중앙값이 이 분포를 더 잘 대표하겠다고 생각하게 된다.
이전에 아예 영화를 만들지 않은 신인감독은, 2005년부터 2009년까지 신인감독의 영화흥행성적의 중앙값을 사용한다.

```
movies$po_dir<-ifelse(movies$num_bef==0,median((movies1 %>% filter(num_bef==0))$performance),movies$po_dir)
```

2. num_vid와 num_actor

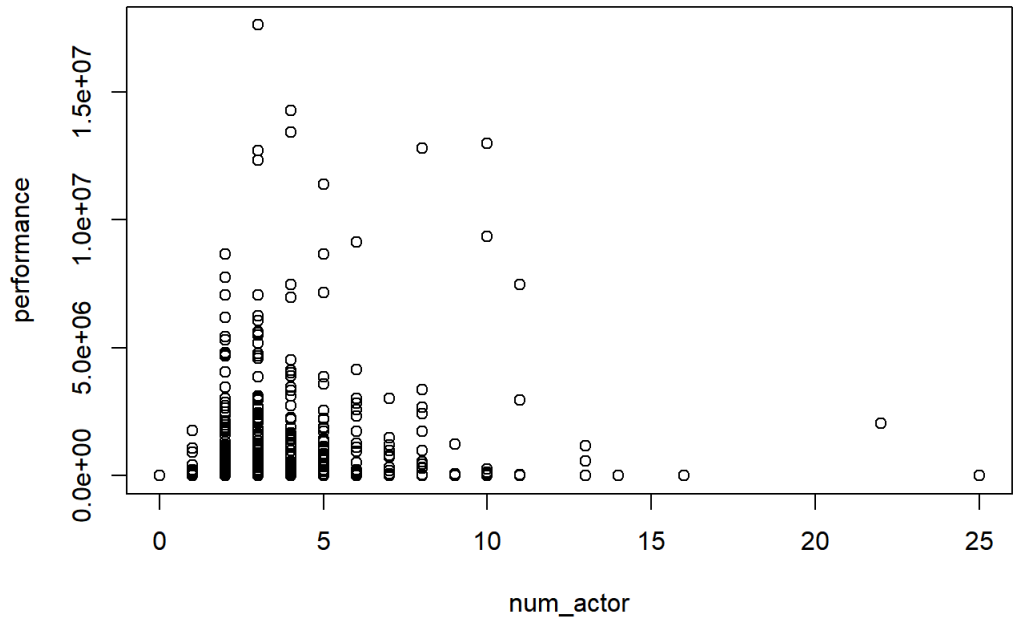
```
plot(movies$num_vid,movies$performance,xlab="num_vid",ylab="performance")
```




```
cor(movies$num_vid,movies$performance)
```

```
## [1] 0.442597
```

```
plot(movies$num_actor,movies$performance,xlab="num_actor",ylab="performance")
```




```
cor(movies$num_actor,movies$performance)
```

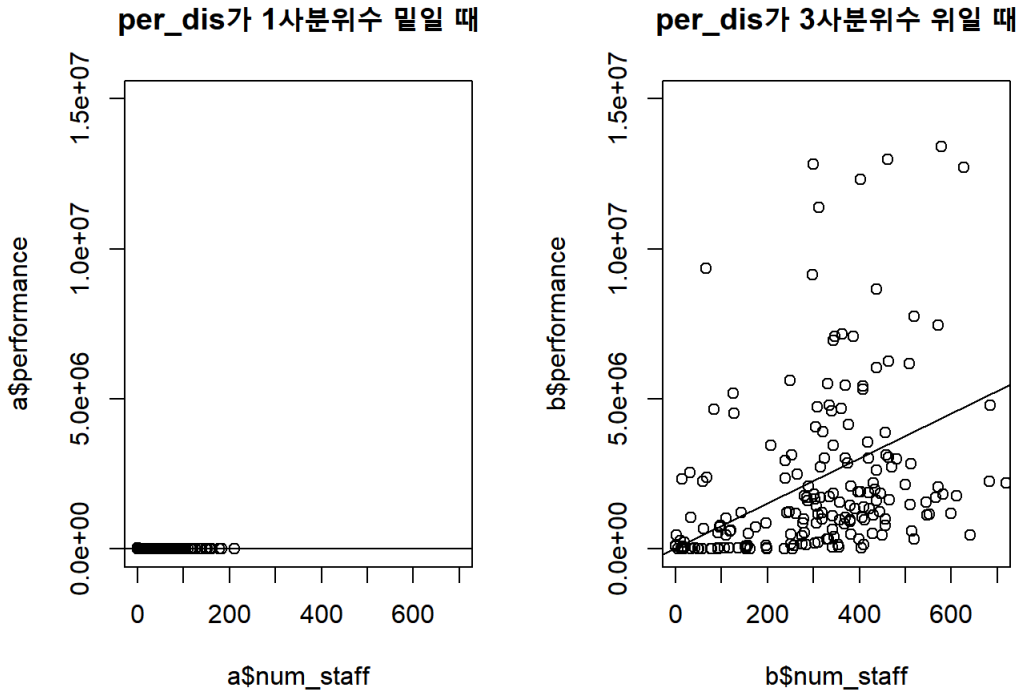
```
## [1] 0.07260086
```

num_vid는 채택하고, num_actor는 버리자.

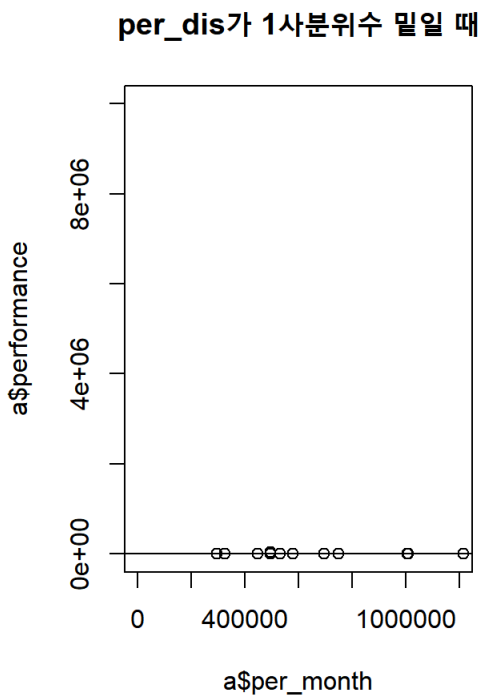
3. 교호작용 고려

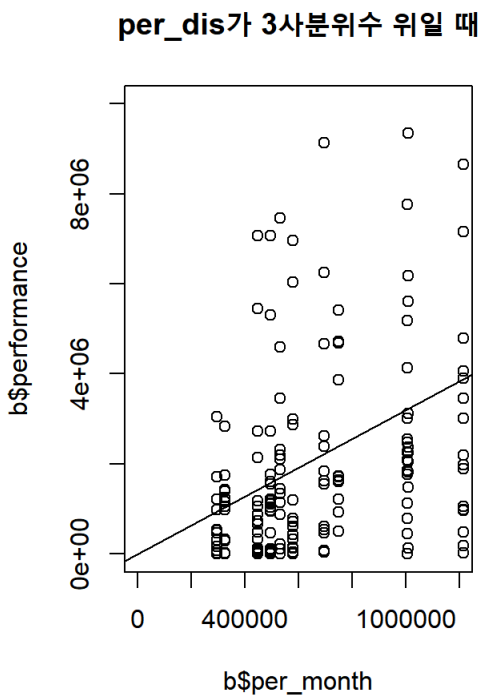
po_dir와 num_vid를 추가해서 선형 회귀모형을 적용해도, 상관계수가 41프로밖에 안 되었다.

i. per_dis와 num_staff의 교호작용

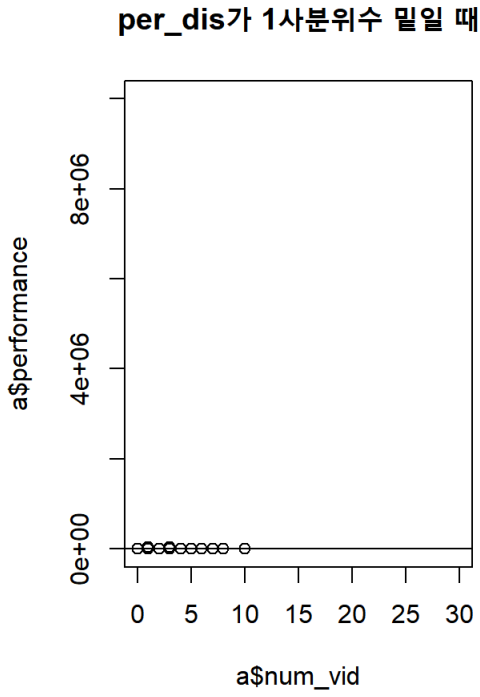


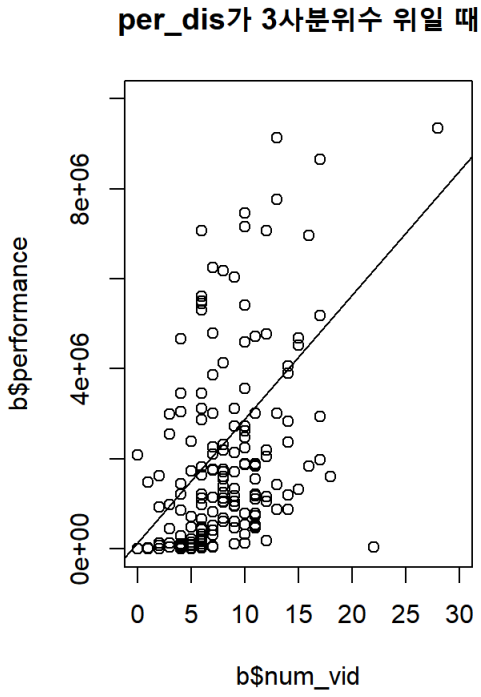
ii. per_dis와 per_month의 교호작용



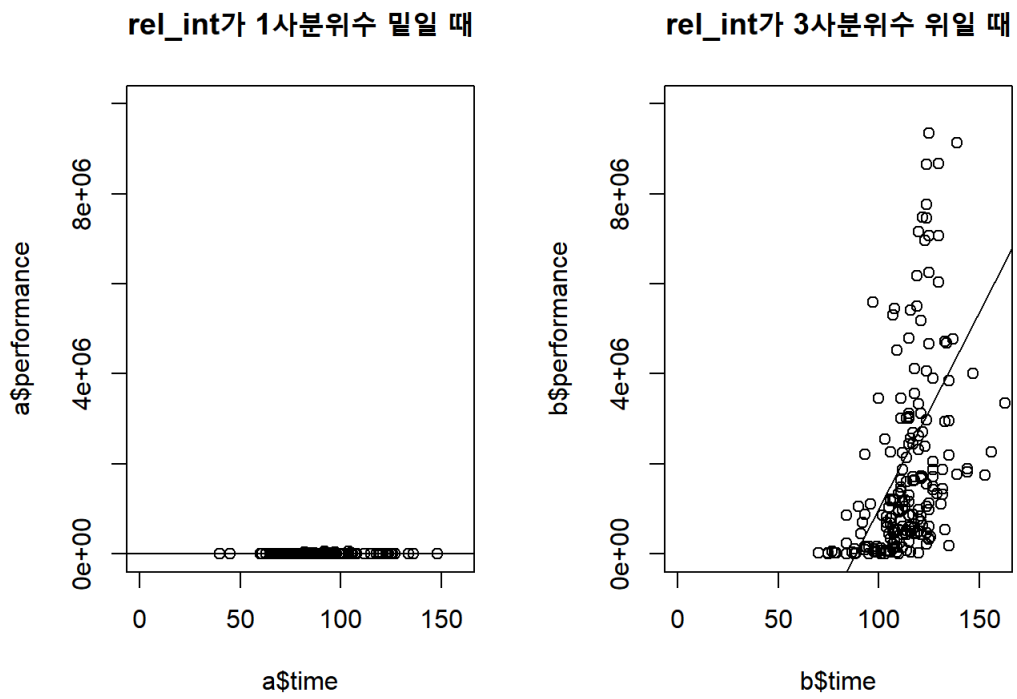


iii. per dis와 num vid의 교호작용





iv. rel_int, time의 교호작용



결론: 교호작용이 있는 변수가 진짜 많다.

그리고, 어떤 한 변수가 엄청 작으면, 아무리 다른 변수가 크더라도 관객수는 적다. 이렇게 해도,,,

```
getavgRsquareLM<-function(num)
{
  Rsquare<-c()
  for(i in 1:num)
  {
    index<-sample(1:nrow(movies),500)

    moviestrain<-movies[index,]
    moviestest<-movies[-index,]
    model1<-lm(performance~per_dis:num_staff+
                           per_dis:per_month+
                           per_dis:num_vid+
                           rel_int:time
                          ,data=moviestrain)
    model1<-step(model1,direction="both")
    moviestest$pred<-(predict(object=model1,moviestest))
    Rsquare<-c(Rsquare,cor(moviestest$pred,moviestest$performance)^2)
  }
  return(Rsquare)
}

result<-getavgRsquareLM(1000)
```

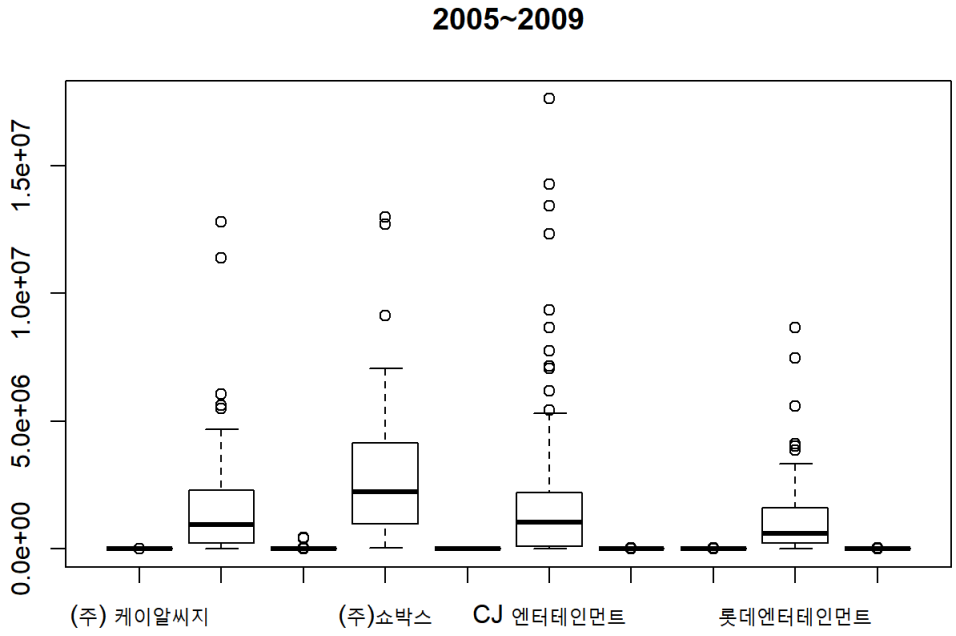
평균적으로 48프로정도 나온다,,

```
summary(result)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3227  0.4526  0.4786  0.4799  0.5086  0.6035
```

boxplot에서 알 수 있듯이 2005년부터 2009년까지 배급사별 관객수 분포는 거의 다 대칭이지 않은 것 같다. 그렇다면, 이 분포를 잘 표현하기 위해선 어떤 지표를 더 사용하면 좋을까?

```
t<-names(sort(table(movies1$distributor),decreasing=T))[1:10]
t1<-movies1[movies1$distributor %in% t,]
t1$distributor<-as.character(t1$distributor)
boxplot(t1$performance~t1$distributor,main="2005~2009")
```



skewness와 표준편차를 활용하자.

```
library(moments)
movies$skew_dis<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  movies$skew_dis[i]<-skewness((movies1 %>% filter(distributor==movies$distributor[i]))$performance)
}

movies$sd_dis<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  movies$sd_dis[i]<-sd((movies1 %>% filter(distributor==movies$distributor[i]))$performance)
}

movies$per_dis1<-1:nrow(movies)
for(i in 1:nrow(movies))
{
  movies$per_dis1[i]<-median((movies1 %>% filter(distributor==movies$distributor[i]))$performance)
}
```

4. 모델 적합

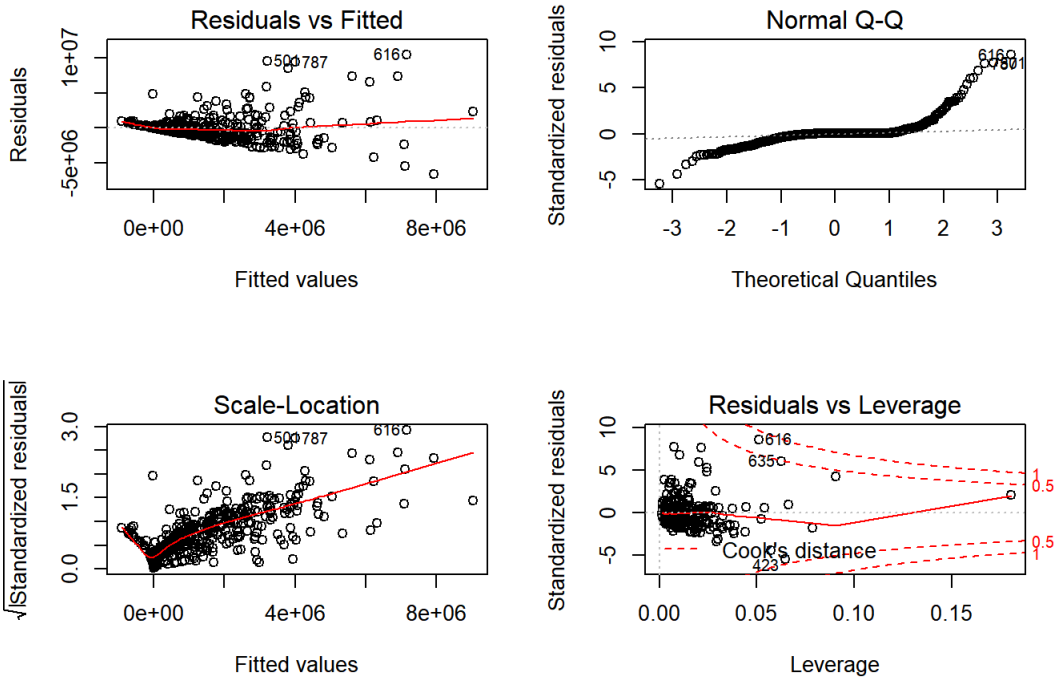
```
model1<-lm(performance~skewdis:per_dis+
                        num_staff:sddis1+
                        num_staff:per_dis+
                        per_dis:num_vid+
                        rel_int:time:per_month
                       ,data=movies)
model1<-step(model1,direction="both")
```

```
## Start:  AIC=23846.93
## performance ~ skewdis:per_dis + num_staff:sddis1 + num_staff:per_dis +
##     per_dis:num_vid + rel_int:time:per_month
##
##                           Df  Sum of Sq        RSS   AIC
## - per_dis:num_staff        1 1.1424e+10 1.3224e+15 23845
## <none>                                  1.3223e+15 23847
## - skewdis:per_dis          1 3.1747e+13 1.3541e+15 23865
## - num_staff:sddis1         1 4.0329e+13 1.3627e+15 23870
## - per_dis:num_vid          1 4.0437e+13 1.3628e+15 23871
## - rel_int:time:per_month   1 2.1296e+14 1.5353e+15 23972
##
## Step:  AIC=23844.94
## performance ~ skewdis:per_dis + num_staff:sddis1 + per_dis:num_vid +
##     rel_int:time:per_month
##
##                           Df  Sum of Sq        RSS   AIC
## <none>                                  1.3224e+15 23845
## + per_dis:num_staff        1 1.1424e+10 1.3223e+15 23847
## - skewdis:per_dis          1 3.4534e+13 1.3569e+15 23865
## - per_dis:num_vid          1 4.7752e+13 1.3701e+15 23873
## - num_staff:sddis1         1 1.9457e+14 1.5169e+15 23960
## - rel_int:time:per_month   1 2.1338e+14 1.5357e+15 23970
```

```
vif(model1)
```

```
##        skewdis:per_dis       num_staff:sddis1        per_dis:num_vid
##               3.757219               3.459092               3.076039
## rel_int:time:per_month
##               1.460601
```

```
par(mfrow=c(2,2))
plot(model1)
```




```
par(mfrow=c(1,1))
```

```
getavgRsquareLM<-function(num)
{
  Rsquare<-c()
  for(i in 1:num)
  {
    index<-sample(1:nrow(movies),500)

    moviestrain<-movies[index,]
    moviestest<-movies[-index,]
    model1<-lm(performance~skewdis:per_dis+
                           num_staff:sddis1+
                           num_staff:per_dis+
                           per_dis:num_vid+
                           rel_int:time:per_month
                          ,data=moviestrain)
    model1<-step(model1,direction="both")
    moviestest$pred<-(predict(object=model1,moviestest))
    Rsquare<-c(Rsquare,cor(moviestest$pred,moviestest$performance)^2)
  }
  return(Rsquare)
}
result<-getavgRsquareLM(1000)
```

결정계수의 평균은 약 51프로 정도로 올랐다.

```
summary(result)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3774  0.4822  0.5164  0.5146  0.5468  0.6521
```

5. 앙상블 모델의 적용

```
##
## Attaching package: 'neuralnet'
```

```
## The following object is masked from 'package:dplyr':
##
##     compute
```

```
##
## Attaching package: 'e1071'
```

```
## The following objects are masked from 'package:moments':
##
##     kurtosis, moment, skewness
```

```
## Warning: package 'randomForest' was built under R version 3.4.4
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

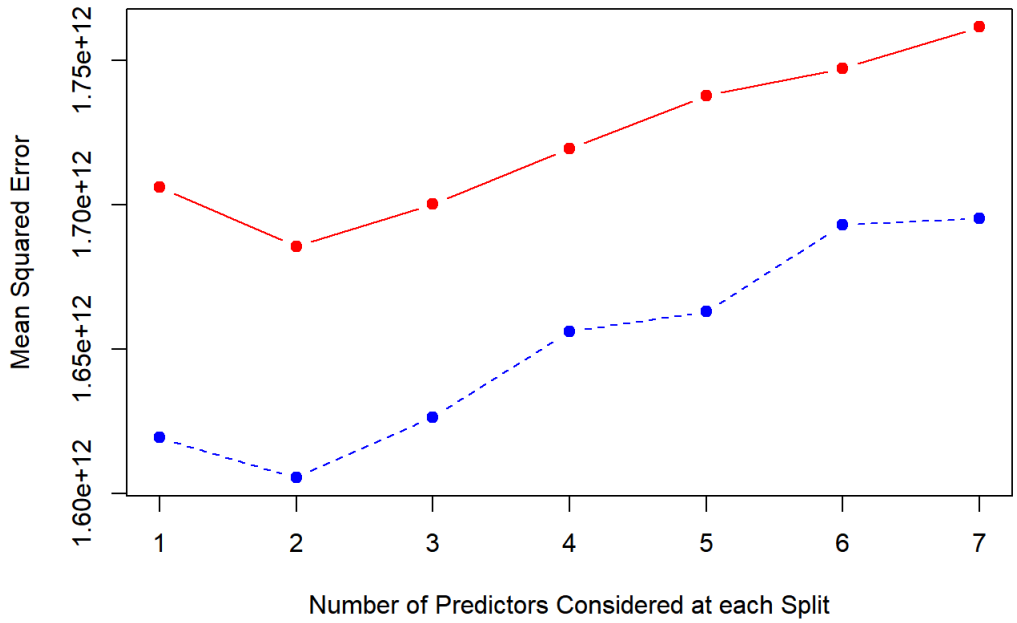
1. RF(Random Forest)

랜덤 포레스트 함수를 하기 전에 예측력을 키울 수 있는 최적의 parameter를 찾아보자. (mtry)

```
oob.err<-rep(0,7)
test.err<-rep(0,7)

a<-movies[,c("po_dir","num_vid","per_month","time","per_dis1","rel_int","num_staff","performance")]

for(i in 1:20)
{
  index<-sample(1:nrow(a),500)
  atrain<-a[index,]
  atest<-a[-index,]


  for(mtry in 1:7)
  {
    rf=randomForest(performance ~ . , data =a , subset = index,mtry=mtry,ntree=400)
    oob.err[mtry] = oob.err[mtry]+rf$mse[400]

    pred<-predict(rf,movies[-index,]) #Predictions on Test Set for each Tree
    test.err[mtry]= test.err[mtry]+with(a[-index,], mean( (performance - pred)^2))

    cat(mtry," ")
  }
}
```

```
## 1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6
7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6
7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6
7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6  7  1  2  3  4  5  6
7
```

```
oob.err<-oob.err/20
test.err<-test.err/20
matplot(1:mtry,
        cbind(oob.err,test.err),
        pch=19,
        col=c("red","blue"),
        type="b",
        ylab="Mean Squared Error",
        xlab="Number of Predictors Considered at each Split")
```



mtry라는 파라미터의 최적값

을 찾아준다.

2. SVM(Support Vector Machine) SVM도 마찬가지로 최적의 파라미터를 찾아보자.

```
tuneResult <- tune(svm, performance~sddis1+num_vid+per_month+time+per_dis1+rel_int+num_staff,data = movies,r
anges = list(epsilon = seq(0,1,0.1), cost = 2^(2:9))
)
```

epsilon과 cost라는 파라미터의 최적의 값을 찾아준다.

plot(tuneResult)

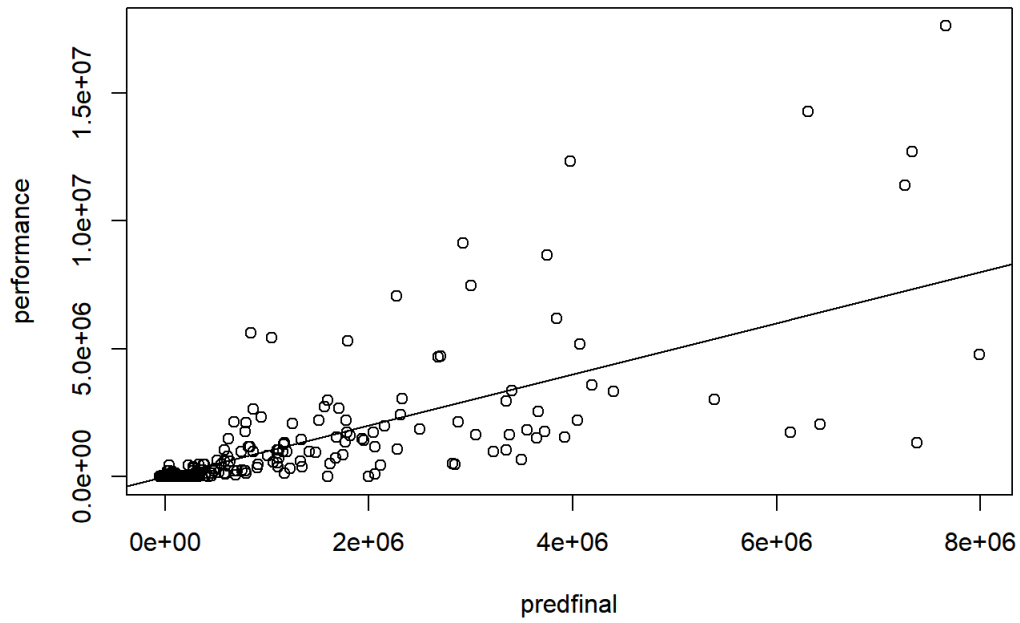
3. 다중회귀모형, 랜덤포레스트, SVM을 섞은 혼합 모델 생성

```
getavgRsquaretotal<-function(num,df)
{
  res<-c()
  library(neuralnet)
  library(dplyr)
  library(e1071)
  library(randomForest)
  lmpred<-df[,1]
  svmpred<-df[,2]
  rfpred<-df[,3]
  real<-df[,4]
  for(i in 1:num)
  {
    index<-sample(1:nrow(movies),500)

    moviestrain<-movies[index,]
    moviestest<-movies[-index,]
    predlm<-0
    predsvm<-0
    predrf<-0
    ##Linear Regression
    model1<-lm(performance~skewdis:per_dis+num_staff:sddis1+num_staff:per_dis+per_dis:num_vid+rel_int:time:p
er_month
               ,data=moviestrain)
    predlm<-predict(object=model1,moviestest)

    M<-max(movies$performance)
    m<-min(movies$performance)

    a<-normalizedataframe(movies[,c("sddis1","num_vid","per_month","time","per_dis1","rel_int","num_staff","
performance")])
    atrain<-a[index,]
    atest<-a[-index,]
    model <- svm(performance~sddis1+num_vid+per_month+time+per_dis1+rel_int+num_staff,atrain,epsilon=0.13,co
st=4)
    pre<-predict(model,atest)
    pre<-inversenormalization(x=pre,min = m,max = M)
    predsvm<-pre


    pre<-0
    a<-normalizedataframe(movies[,c("num_vid","per_month","time","per_dis","rel_int","num_staff","performanc
e")])
    atrain<-a[index,]
    atest<-a[-index,]
    rf<-randomForest(performance~.,data=a,subset=index,mtry=2,ntree=1000)
    pre<-predict(rf,a[-index,]) #Predictions on Test Set for each Tree
    pre<-inversenormalization(x=pre,min = m,max = M)
    predrf<-pre

    predfinal<-(predlm+predsvm+predrf)/3
    lmpred<-c(lmpred,predlm)
    svmpred<-c(svmpred,predsvm)
    rfpred<-c(rfpred,predrf)
    real<-c(real,moviestest$performance)
    res<-c(res,cor(predfinal,moviestest$performance)^2)

    print(paste(round(cor(predfinal,moviestest$performance)^2,digits=3),round(mean(res),digits=3),sep="이고
평균은 "))
  }
  return(list(res,data.frame(lmpred=lmpred,svmpred=svmpred,rfpred=rfpred,real=real)))
}

df<-data.frame(lmpred=c(0),svmpred=c(0),rfpred=c(0),real=c(0))
df<-getavgRsquaretotal(200,df)
```

```
summary(df[[1]])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4234  0.5382  0.5814  0.5739  0.6069  0.6864
```

6. Conclusion



최종적으로, training set에서 적합한 앙상블 모델의 test 데이터에서 예측치의 결정계수는 평균적으로 57프로 정도 되었다.