**빅데이터 프로그래밍 프로젝트 보고서**

1. 개요

빅데이터 프로그래밍 기말 프로젝트를 위하여 데이터를 수집한 후 분석하는 과정을 설명합니다. 사용되는 데이터는 국내 웹 사이트에서 크롤링한 데이터입니다. 웹 사이트는 국내 최대의 온라인 서점인 yes24의 베스트 셀러를 대상으로 하며 저는 각 베스트 셀러의 가격과 베스트 셀러의 페이지 간에 상관관계 있는지 분석을 할 생각입니다. 크롤링 작업에는 python3를 사용할 것이고 분석 작업에는 프로젝트의 주제인 R이 사용됩니다.

2. 동기

대학생이라면 누구나 학교에서 공부를 하던 도서관에 가던 수 많은 책들을 접하게 됩니다. 그리고 각 책들은 저마다 다른 페이지 수와 다른 가격을 갖고 있습니다. 흔히 책 페이지의 수와 책의 가격에는 상관관계보다 인과관계가 어울린다고 생각하시는 경우가 많습니다. 재화를 생산하기 위한 비용이 증가하면 그만큼 가격이 올라가기 때문입니다. 하지만 저의 생각은 다릅니다. 책의 가격은 생산비용보다 책 안에 기록되어 있는 정보의 가치에 따라 정해진다고 생각합니다. 실제로 페이지의 수가 많다고해서 항상 책의 가격이 높은 것은 아닙니다. 즉 책의 가격은 생산 원가보다 그 책이 지니고 있는 가치에 더 큰 영향을 받는다고 볼 수 있습니다.

하지만 다른 시점으로 볼 때 페이지의 수는 책의 가격에 영향을 줄 수도 있다고 생각합니다. 장문의 글은 대부분 단문의 글 보다 더 가치있는 정보를 가지기 마련입니다. 장문의 글은 상대적으로 더 많은 정보를 수록하게 되고 장편의 이야기를 소개할 수 있습니다. 이런 시점에서 본다면 책 페이지의 수가 책의 가치와 상관관계가 있을 것이라고 저는 생각을 해보았습니다. 따라서 저는 이번 프로젝트의 주제로 ‘책 페이지 수’와 ‘책의 가치’ 간의 상관관계 유무를 밝힐 것입니다.

3. 분석 작업

우선 yes24페이지에서 어떤 부분을 크롤링 할 것인지에 대하여 설명 드리도록 하겠습니다.

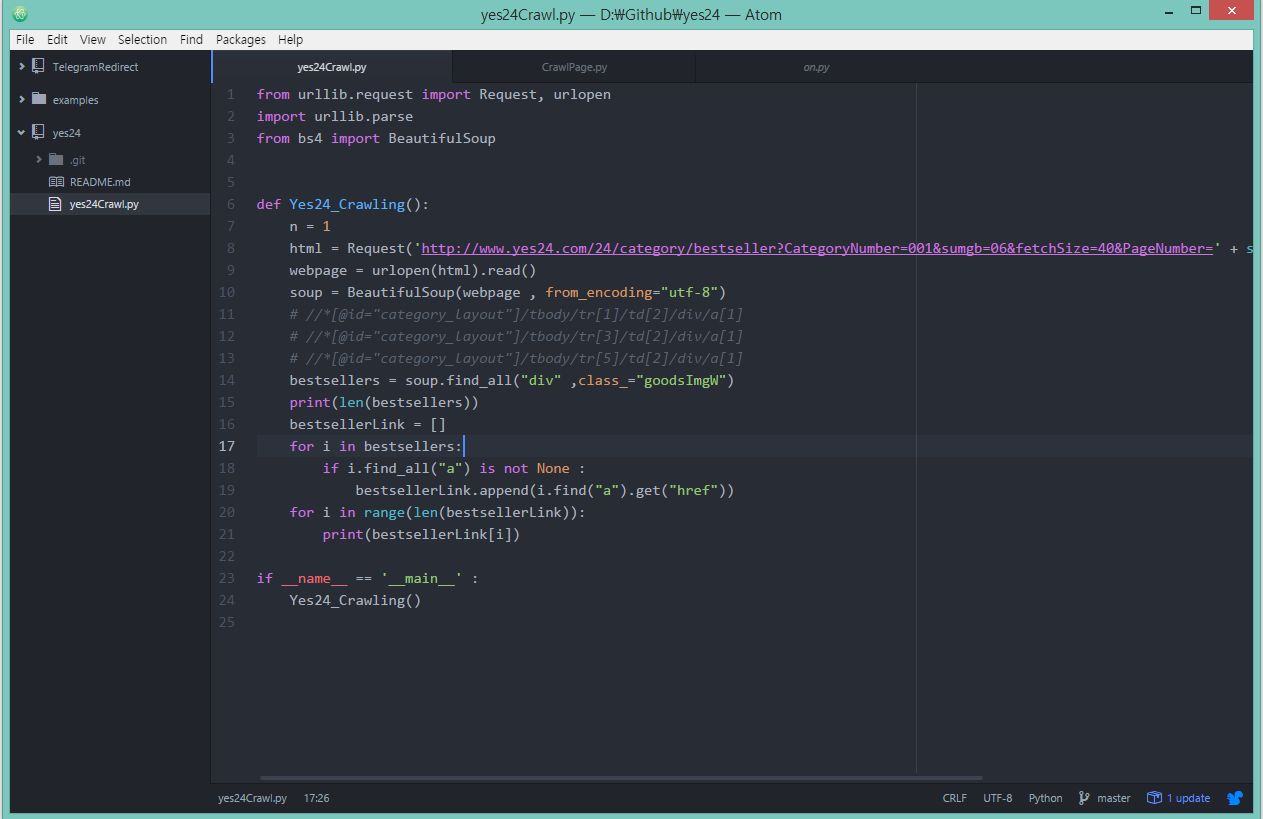


저는 yes24의 페이지에서 국내도서 종합 베스트셀러 페이지를 크롤링해올 예정입니다. 페이지의 주소는 ‘http://www.yes24.com/24/category/bestseller?CategoryNumber=001&sumgb=06&fetchSize=40&PageNumber=1’이며 마지막 PageNumber 매개변수에 들어가는 숫자에 따라 베스트셀러가 진열되어 있는 페이지가 달라지게 됩니다. 한 페이지에는 40권의 책이 존재하며 페이지는 총 250페이지까지 존재합니다. 저의 주제는 페이지 수와 책의 가격간의 상관관계를 구하는 것입니다. 그리고 저는 페이지 수를 독립변수로 책의 가격을 종속변수로 설정할 예정입니다. 따라서 제가 이 웹으로부터 크롤링해야 하는 정보는 책 페이지의 수와 책의 가격입니다.

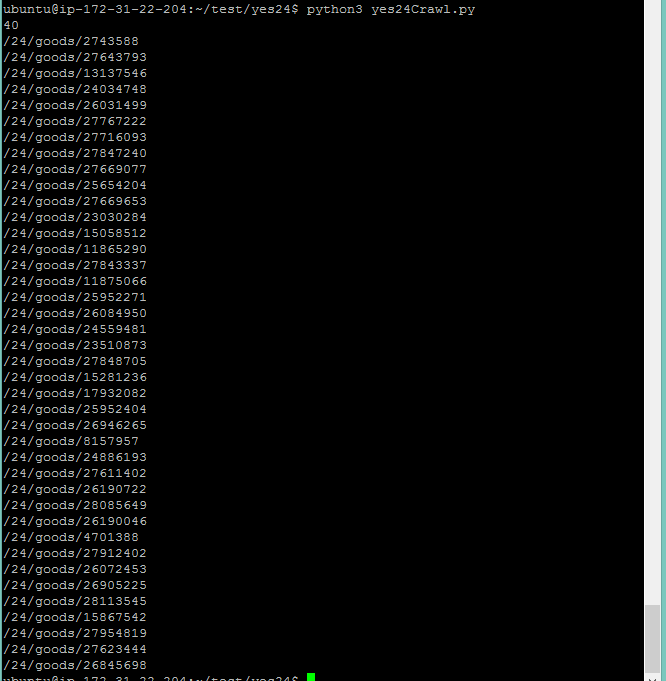
하지만 위의 이미지와 같이 첫 번째 페이지에서는 책 페이지 수와 같은 책에 대한 상세 정보가 나와 있지 않습니다. 따라서 저는 베스트셀러가 진열되어 있는 지금과 같은 페이지에서 각 책에 대한 상세 페이지들로 이동하는 링크들을 모두 수집하기로 하였습니다. 그리고 수집된 각 책의 상세 페이지들에서 다시 한 번 크롤링 작업을 진행하여 상세 페이지에서 책에대한 상세 정보를 수집하도록 하였습니다.

첫 번째 베스트셀러 리스트 페이지에서 책에 대한 상세 페이지 링크를 크롤링 하는 과정은 다음과 같습니다.

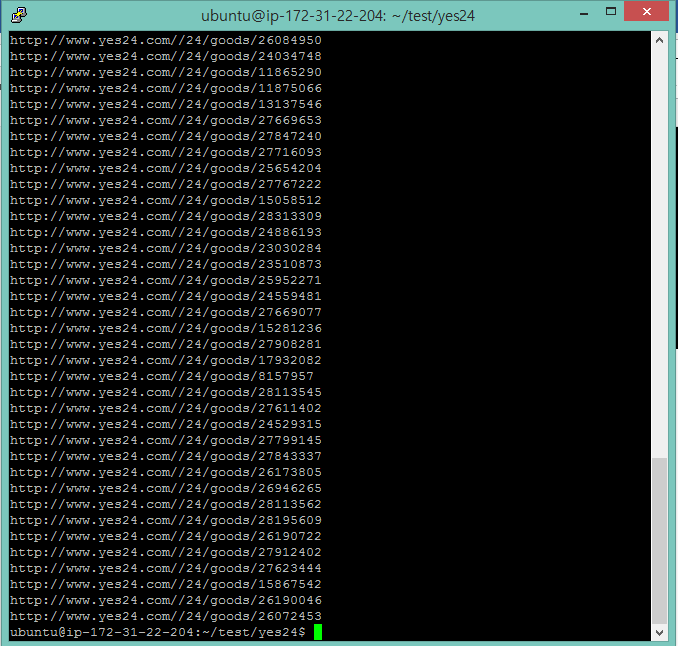
우선 저는 Atom 에디터를 사용하여 코드를 작성하였습니다. 제가 데이터 마이닝 작업에 사용하는 언어는 Python3인데 Python3에서 크롤링 작업을 하는데 주로 사용되는 라이브러리는 Beautifulsoup4 입니다. 따라서 작업을 하기에 앞서 이 라이브러리를 설치해야 합니다. 또한 html 소스를 가지고 오기 위해서 Requests라는 라이브러리도 필요합니다.



아톰 에디터로 베스트셀러 리스트에 있는 각 책들의 상세페이지로 이동하는 링크들을 출력하는 코드를 작성하였습니다. 링크의 주소 전체가 출력되는 것이 아니라. ‘/24/goods/2743588’과 같이 yes24 페이지의 하위 디렉토리와 파일 경로만 따로 출력이 됩니다. 따라서 실제 베스트셀러들의 상세 페이지로 이동하기 위해서는 기본 yes24 페이지 주소 ‘http://www.yes24.com/'에 각 베스트셀러 상세페이지의 위치 ‘/24/goods/2743588’을 합해주어야 합니다. 즉 첫 번째 베스트셀러의 상세페이지는 ‘http://www.yes24.com/24/goods/2743588'과 같습니다. 밑에 결과 화면은 각 베스트셀러 상세페이지의 위치를 출력한 결과를 보여주고 있습니다.



따라서 실제 베스트셀러 상세페이지 주소를 만들기 위해 두 문자열을 결합하였습니다. bestsellerLink 배열의 bestsellerLink.append(i.find("a").get("href")) 를  
bestsellerLink.append('http://www.yes24.com/'+i.find("a").get("href")) 로 수정하여 배열에 추가했습니다.



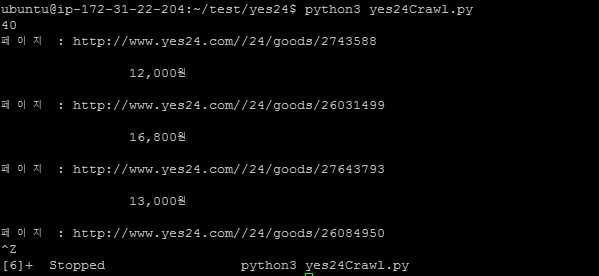
다시 추출된 베스트셀러 책들의 상세페이지 링크들입니다.



상세페이지는 다음과 같은 화면으로 구성되어 있습니다. 제가 여기서 필요한 것은 도서명, 도서의 가격 그리고 도서의 페이지 수 입니다. 따라서 저는 위와 같이 세 부분으로 나누어 크롤링을 진행하였습니다.



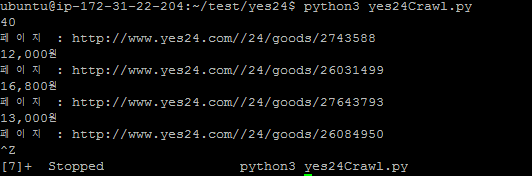
보기에는 평범한 데이터 같아 보이지만 실제로는 사이에 공백이 들어가 있는 값들이 많아 크롤링을 하는 작업을 여러 번 반복하게 되었습니다. 또한 콤마(,)와 같은 기호는 가독성을 높이지만 숫자로 처리할 수 없기 때문에 생략을 해주어야 했습니다.



공백을 제거하기 전 가격을 출력한 화면입니다. 저는 처음 공백을 제거하기 위하여 다음과 같은 줄을 추가 하였습니다.

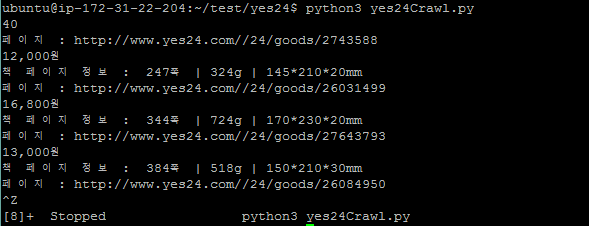
price = price.replace(" ","")

price = price.replace("\n","")

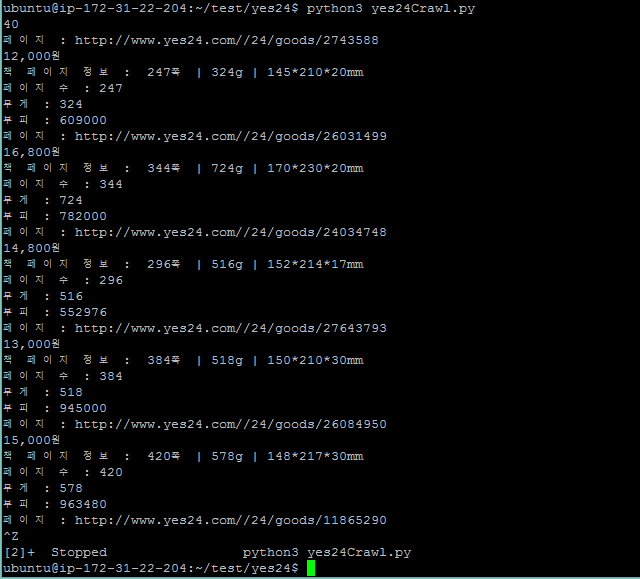


하지만 추후에 csv 파일로 데이터를 옮길 때 예상치 못한 문자가 따라오는 경우가 생겨 replace 함수 대신 좌우의 공백을 제거해 주는 strip이라는 함수로 대체하게 되었습니다.

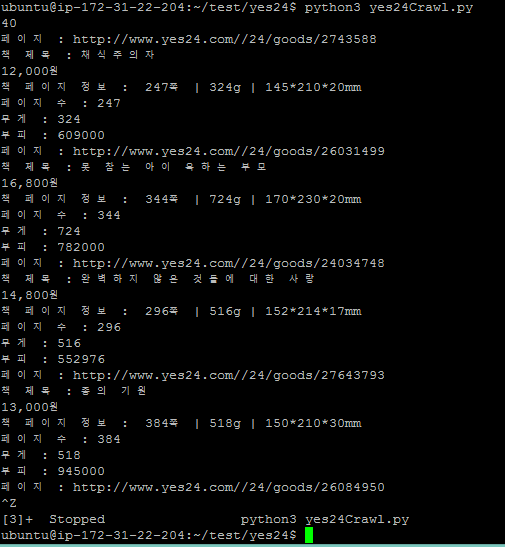




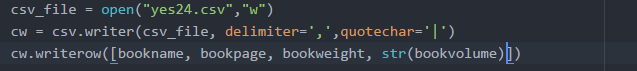
책의 페이지 정보는 책의 무게, 책의 크기(책의 부피)와 함께 기록되어 있었습니다. 저의 원래 계획은 책의 페이지 정보만 수집하는 것이었지만 더 다양한 분석 결과를 얻어 보기 위해 책의 무게와 책의 부피도 함께 크롤링 하기로 결정하였습니다. 책의 정보가 기록되어 있는 이 html element 요소는 pdSize라는 클래스를 이 페이지 내에서 유일하게 갖고 있었기 때문에 크롤링 하는데 큰 어려움이 없었습니다. 하지만 수 많은 책들의 상세페이지에서 이 부분을 가지고 오기 시작하였을 때에는 큰 어려움이 있었습니다. 페이지 정보가 누락되었거나 무게가 기록되어야 할 부분에 부피가 기록되어 있거나 부피의 계산이 잘못 이루어진 경우가 가끔씩 있었습니다. 심지어는 19세이상 성인도서로 분류되어 상세 페이지 링크로의 접근이 어려운 경우도 있었습니다. 따라서 이런 예외 상황을 결측치 NA로 바꾸어 처리하거나 필요한 경우 데이터를 건너뛰는 작업을 수행하였습니다. 저는 이러한 작업을 모두 마친 후 데이터에서 필요한 부분만을 추출해 내기 위해 split 함수와 replace 함수를 사용하였습니다. 예를 들어 페이지 정보에 포함되어 있는 “쪽”이라던가 무게의 단위인 ‘g’, 부피의 단위인 ‘mm’ 그리고 그 사이의 공백을 처리하는 작업을 진행하였습니다. 특히 부피의 경우 세 수를 곱해야 하는 작업이기 때문에 세 수를 곱하는 과정을 따로 만들었습니다. 부피는 3차원 값을 지니어야 하지만 가끔 부피의 계산이 제곱 밀리미터로 표현되어 있는 경우 이 값은 결측치 NA로 수정하였습니다.



위와 같이 저는 페이지의 수와 무게를 정리하고 부피를 세제곱 밀리미터 단위로 계산하여 결과를 다시 출력하였습니다.



마지막으로 책의 제목까지 위와 같은 방법으로 수집을 완료하여 출력하였습니다.



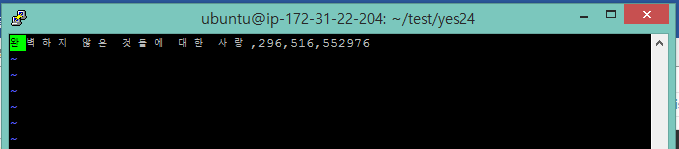
R에서 데이터 분석을 하기 위해서는 데이터 마이닝 작업의 결과를 csv로 저장해야 한다고 생각하였습니다. 따라서 위와 같이 Python에서 흔히 csv파일을 만들 때 사용하는 방식대로 코드를 작성하였습니다.



저는 크롤링 작업을 AWS의 EC2 인스턴스에서 수행하였습니다. 그리고 그 인스턴스의 운영체제는 Ubuntu를 사용했습니다. 특별한 이유가 있어서 크롤링 작업을 로컬이 아닌 외부에서 진행한 것은 아니고 기존에 EC2 인스턴스에 Beautifulsoup4와 Requests등 파이썬 라이브러리등을 미리 설치해 두었기 때문에 추가 적인 설치 작업을 피하고자 이와 같은 방식으로 진행하였습니다. 다만 우분투에서 touch를 통하여 빈 파일을 생성할 때 그 파일에 대한 권한은 rw-r--r-- (644)로 지정되어 있기 때문에 다음과 같은 순서로 파일의 접근 권한(모든 유저 읽고 쓰고 실행 가능 rwxrwxrwx)을 변경해주어야 했습니다.

Sudo touch yes24.csv

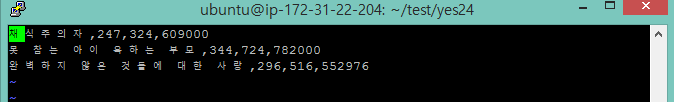
Sudo chmod 777 yes24.csv



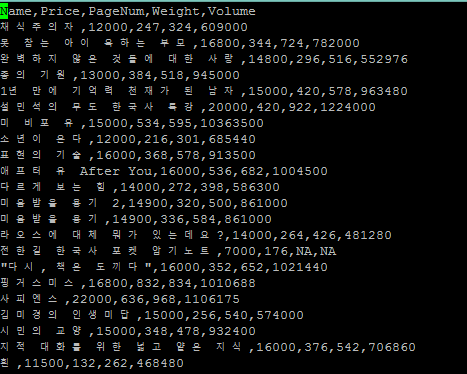
그럼 이제 yes24.csv 파일로 접근이 가능해져 csv파일에 책의 정보를 추가할 수 있었습니다. 프로그램을 시작하는 부분에서 yes24.csv 파일을 write 옵션으로 열었기 때문에 새로운 책의 정보가 기록될 때마다 기존의 책의 정보가 삭제되는 문제가 있었습니다. 따라서 저는 write옵션에서 append로 옵션을 수정하였습니다.

csv\_file = open("yes24.csv","a")

w 🡪 a 옵션으로 수정을 하였다.



그 결과 책들의 정보들을 이어서 기록할 수 있었습니다. csv 파일의 가장 앞 부분에 header를 기록해주기 위하여 각 열의 이름을 Name(도서명), Price(도서 가격), PageNum(페이지 수), Weight(책의 무게), Volume(책의 부피)으로 정하였습니다. 데이터 수집 과정의 최종 결과 화면 입니다.



이렇게 R을 이용한 데이터 분석에 필요한 크롤링 작업을 마쳤습니다.

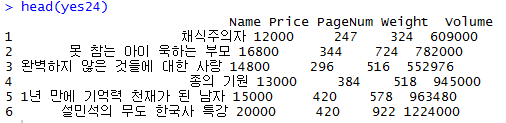
모든 데이터의 수 : 9845 개

\*코드는 가장 마지막에 첨부

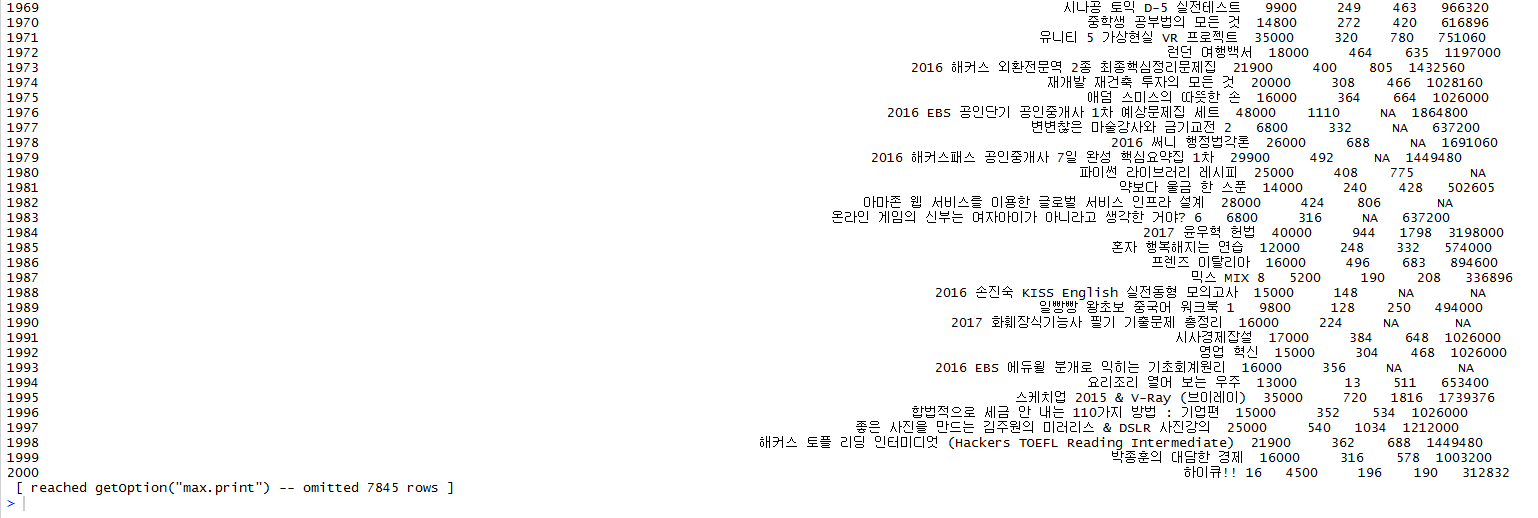
4. 데이터 분석

우선 작업 디렉토리를 설정하고 read.csv를 이용하여 데이터 마이닝 작업의 결과로 얻은 데이터를 불러왔습니다.

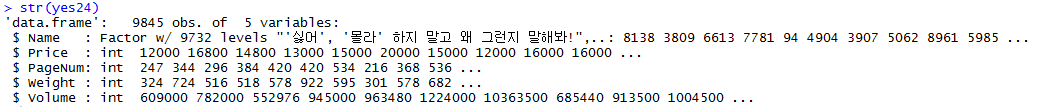




베스트셀러 중에서도 랭킹 1위는 최근 맨부커 상을 수상한 채식주의자 입니다.



결과가 너무 많아 모든 데이터가 나오지는 않고 2000번째 행까지 책의 제목과 책의 가격, 책의 무게와 책의 부피가 출력되었습니다. 나머지 7845개는 출력되지 않았습니다.

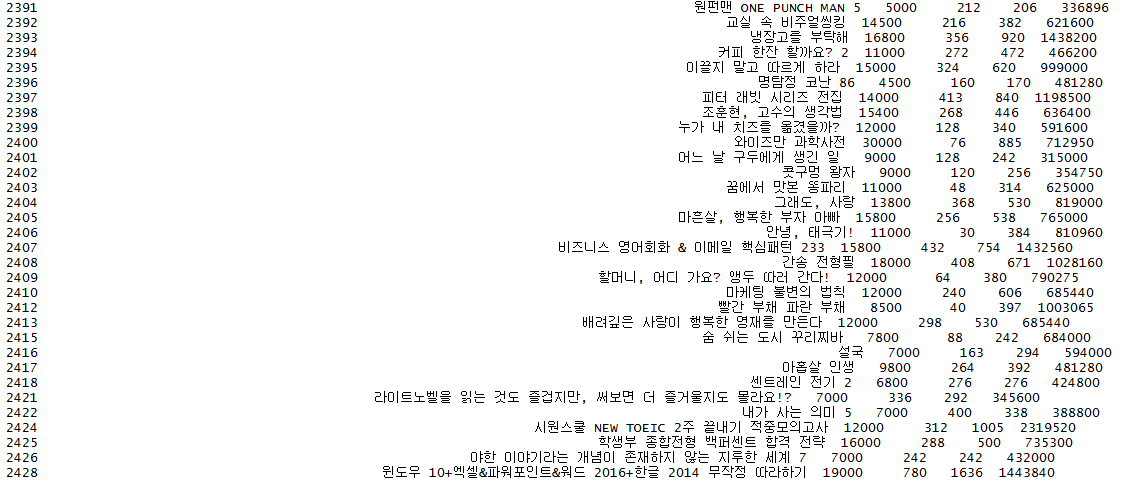


기본적인 자료들은 책 제목만 Factor 형으로 되어있고 나머지 열들은 정수형으로 정의되어 있습니다.

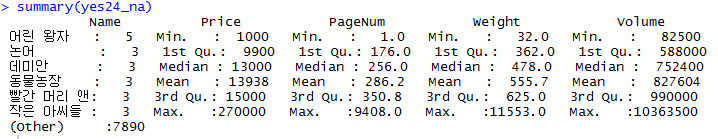
자료에는 많은 결측치(NA)가 포함되어 있기 때문에 결측치가 포함되어 있는 항은 제거하도록 하였습니다. 결측치가 있는 행을 제거하기 위해 na.omit(yes24)과 같은 명령어를 사용하였습니다.



결측치로 인해 제거된 행은 모두 1935줄 입니다. 그 출력 결과는 다음과 같습니다.



우선 기초 통계량을 구하기 위하여 summary 함수를 이용하였습니다.

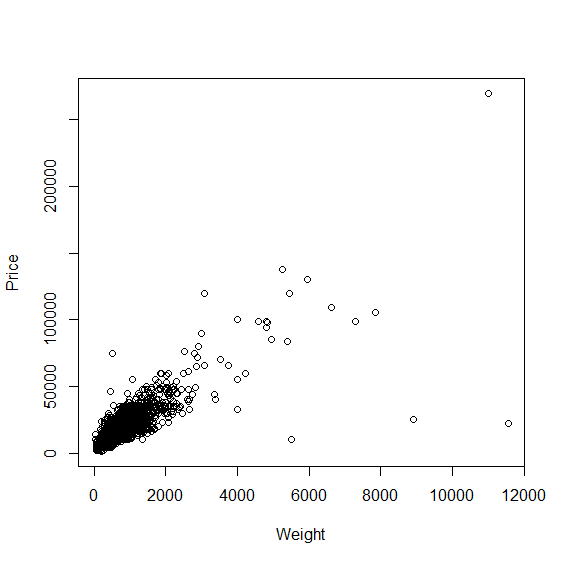
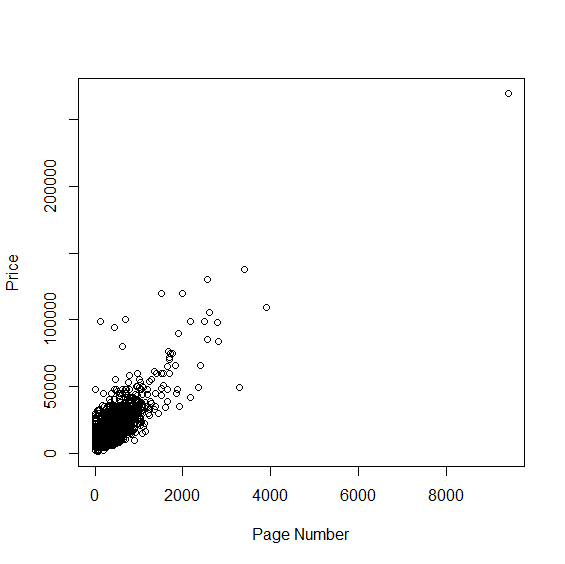


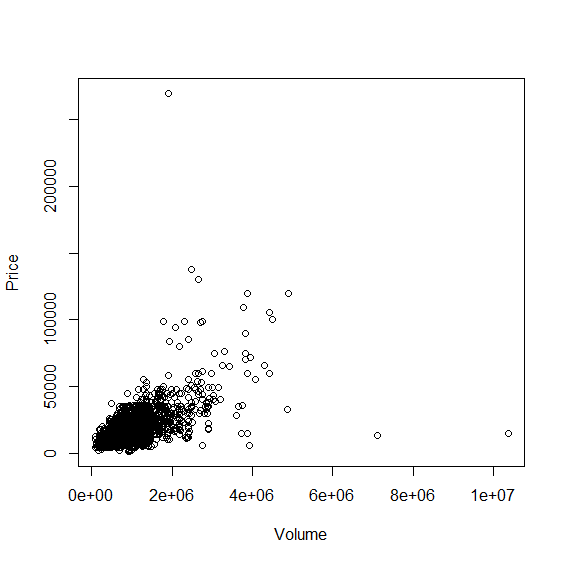
이를 통해 도서의 가격, 페이지 수, 무게, 부피 각각에 대한 최소, 최대값, 1분위수, 중위수, 3분위수, 그리고 평균을 구할 수 있었습니다.

그 후 가격과 각 요소들 간의 산점도를 그리기 위해 plot 함수를 사용하였다. 하지만 데이터의 양이 너무 크다는 메시지와 함께 다음과 같은 에러 메시지가 나왔다.



이에 대한 해결책은 간단하였습니다. RStudio에서 실행하지 않고 R 콘솔에서 실행하는 방법을 사용했습니다.





왼쪽 위부터 차례대로 첫 번째 산점도는 ‘페이지 수와 가격’에 관한 것이고 두 번째 산점도는 ‘책의 무게와 가격’ 세 번째 산점도는 ‘책의 부피와 가격’을 나타낸 것입니다. 언뜻 보기에도 세 경우 모두 양의 상관관계를 조금씩 가지고 있는듯 하지만 정확히 확인해 보기 위하여 상관계수를 구하는 절차를 진행하기로 하였습니다.

저는 피어슨 상관계수를 구하는 함수인 cor을 사용하였습니다.

가격과 페이지의 수



가격과 책의 무게



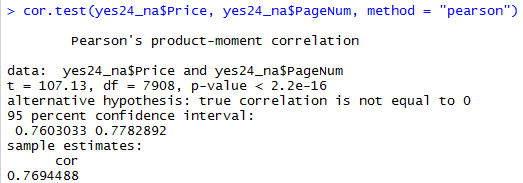
가격과 책의 부피



세 관계 모두 양의 상관관계를 지니고 있었고 책의 무게, 페이지의 수, 부피의 순서대로 상관계수가 높게 나타났습니다. 특히 책의 무게가 책의 가격과 가장 높은 상관관계를 지니고 있다는 사실이 흥미로웠습니다. 일반적으로 상관계수가 +0.3과 +0.7 사이이면, 뚜렷한 양적 선형관계이며, 상관계수가 +0.7에서 +1.0 사이이면 강한 양적 선형관계라고 합니다. 따라서 위의 결과를 보았을 때 책의 가격과 책의 부피는 뚜렷한 양적 상관관계를 지니고 있고, 책의 가격과 책의 페이지 수, 책의 가격과 책의 부피는 강한 양적 상관관계를 지니고 있다고 볼 수 있습니다.

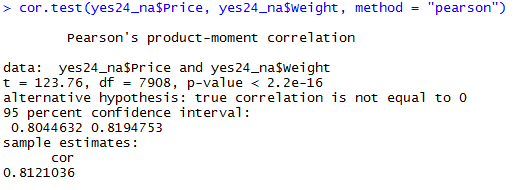
다음으로 피어슨 상관계수를 이용한 통계적 유의성을 검증하기 위해 신뢰구간과 p-value 값을 함께 구해보았습니다.

가격과 페이지의 수

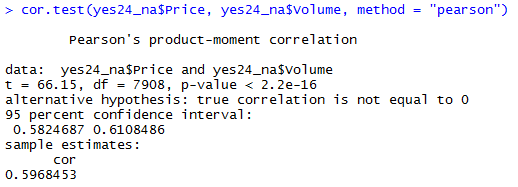


우선 상관관계는 위와 마찬가지로 0.7694488이 나왔고 신뢰구간은 0.7603033 ~ 0.7782892로 나왔습니다. 이 신뢰구간이 95%의 신뢰 수준이라고 할 때 p-value가 2.2e-16으로 0.05보다 매우 작으므로 두 변수 간에 차이가 없다는 귀무가설을 기각하고 대립가설을 채택할 수 있습니다. 즉 두 변수 사이에는 관련성이 매우 높다는 것을 알 수 있습니다.

가격과 책의 무게

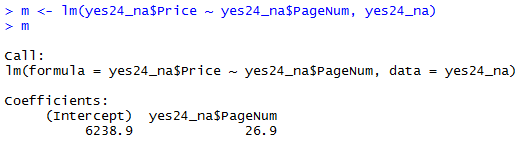


가격과 책의 부피

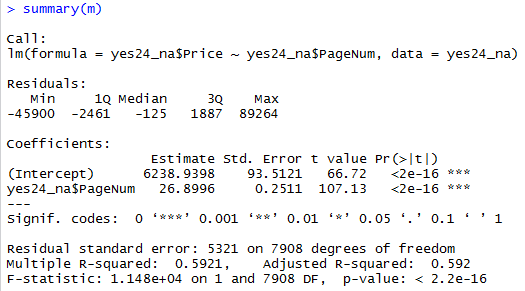


나머지 두 경우 모두도 p-value가 0.05보다 작기 때문에 귀무가설을 기각하고 대립가설을 채택하며 두 변수 사이의 관련성이 높음을 보여주고 있습니다.

그 다음으로 책의 가격과 각 요소들에 대한 단순선형회귀 분석을 진행하였습니다. 대표적으로 책의 가격과 책 페이지 수에 대한 결과 이미지 입니다.



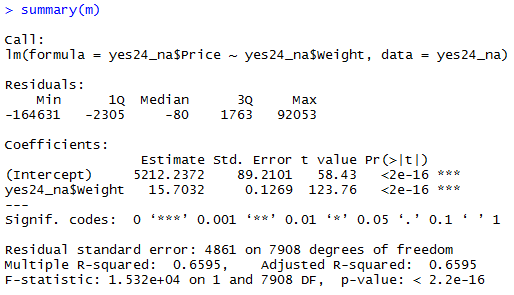
즉 Intercept 값인 6238.9에 26.9 \* 페이지 수를 더하는 형태로 책의 가격을 정의할 수 있습니다.



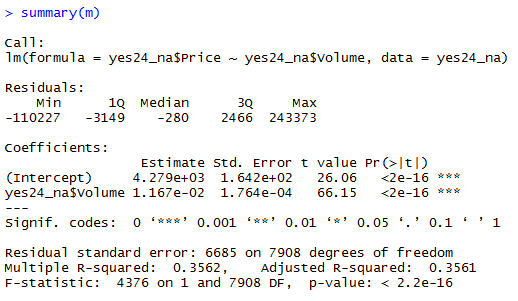
다음과 같이 분석 결과를 보았을 때 유의 수준이 0에 가까운 \*\*\* 수준이므로 페이지의 수 변수에 대한 유의 수준이 굉장히 낮다고 볼 수 있습니다. 또한 결정계수인 Multiple R-squared 값과 Adjusted R-squared 값이 0.0001로 그 차이가 매우 적게 나타났는데 이는 페이지 수와 가격에 따라서 추정된 회귀선이 변수 사이의 관계를 완전히 잘 설명해 주고 있다는 것을 의미합니다. 마지막으로 결정계수가 0.4 이상일 때 데이터의 유용성을 판단할 수 있는데 여기서는 결정계수가 0.5921로 0.4보다 크기 때문에 의미있는 데이터라고 판단할 수 있습니다.

같은 방식으로 다른 변수와의 결과 입니다.

가격과 책의 무게

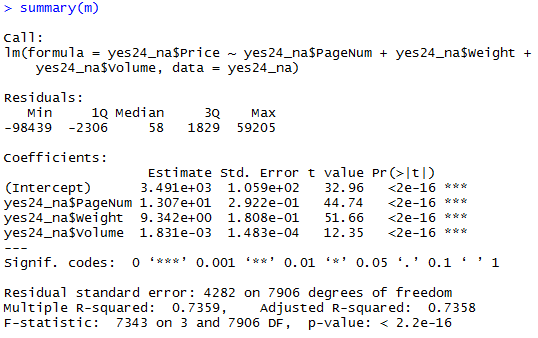


가격과 책의 부피



세 값 모두 유의수준은 굉장히 낮게 나왔지만 가격과 부피와의 관계에서 결정계수가 0.3562로 다소 낮게 측정되었습니다.

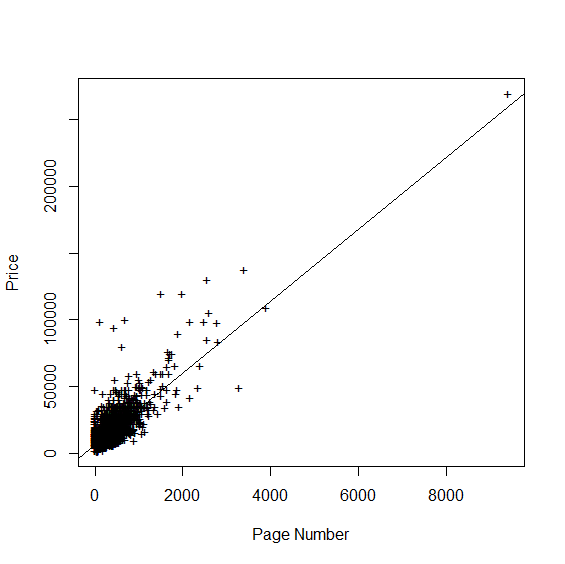
중선형회귀분석



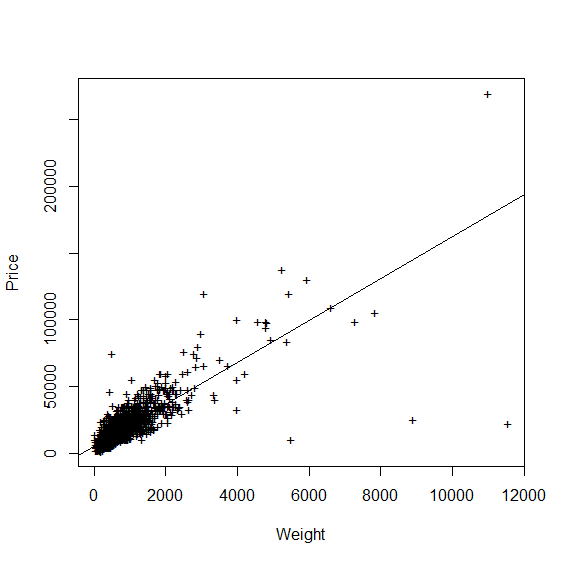
또한 중선형회귀분석을 통해 독립변수를 세 가지로 두었을 때 유의수준은 0에 가까운 수준으로 낮았고 결정계수 역시 0.4보다 높은 수준으로 종속변수와 독립변수사이에 관련성이 높다는 결론을 얻을 수 있었습니다.

지금의 분석결과에서 나온 회귀 계수를 산점도에 선으로 추가해 보았습니다.

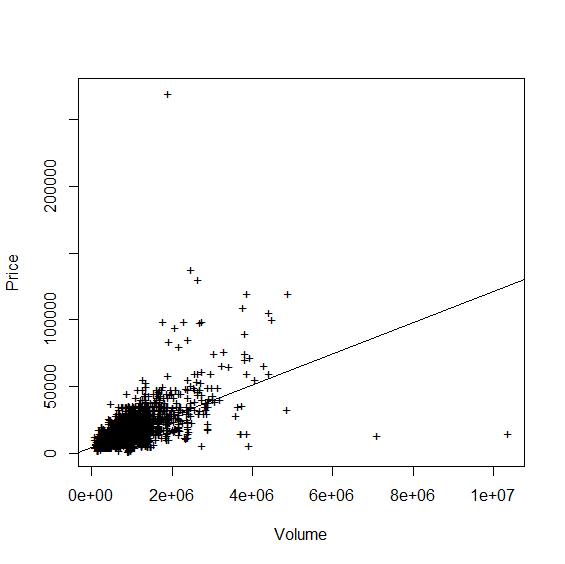
가격과 페이지 수



가격과 무게



가격과 부피



위의 세 그래프에서 볼 수 있듯이 가격과 각 변수 사이의 상관관계는 가격과 책의 무게 사이의 관계가 가장 높게 나타났으며 가격과 부피 사이의 관계가 상관관계가 가장 낮은 것으로 나타났습니다. 독립변수의 변화량에 따른 종속변수의 변화량은 페이지 수가 가장 크며 부피와의 관계가 가장 나타난 것으로 나타났습니다. 전반적으로 세 변수 모두 가격과 양의 상관관계를 가지고 있으며 유의수준은 모두 낮은 수준으로 나타났습니다.

5. 분석 결과의 시사점 및 결론

처음에는 yes24 국내 베스트셀러의 책의 페이지 수에 따른 책의 가격을 분석하기 위해 두 변수 간의 상관관계를 위해 프로젝트를 진행하였습니다. 도중에 책 페이지의 수 뿐만 아니라 책의 무게와 책의 부피도 함께 데이터를 수집하게 되면서 더 다양한 분석 결과를 도출해 볼 수 있었습니다.

분석 결과 책의 가격과 책의 페이지 수, 책의 무게, 책의 부피 사이에는 모두 양의 상관관계가 존재 한다는 것을 알 수 있었습니다. 세 변수의 상관계수는 각각 0.7694488, 0.8121036, 0.5968453이었고 세 변수의 유의 수준 모두 0.05에 한참 못 미치면서 가격과의 관련성이 높음을 보여주었습니다. 이 결과는 페이지 수가 많을수록, 책의 무게가 무거울수록, 책의 부피가 클수록 책의 가격이 높다는 것을 시사합니다. 흥미로운 사실은 책 페이지의 수와 책의 가격 간의 상관성보다 책 무게와 책의 가격 간의 상관계수가 높다는 사실입니다. 책 페이지 수 자체보다는 책 무게가 책의 가격을 결정하는데 있어 더 크게 작용한다는 의미입니다.

6. 전체 코드

**\* R 코드**

# 작업 디렉토리 설정

setwd("D://R/")

# 크롤링 한 csv 파일 읽기

yes24 <- read.csv("yes24.csv")

# 상위 몇 줄만 출력

head(yes24)

# 전체 출력

yes24

# 데이터 형태 보기

str(yes24)

# 결측치 제거

yes24\_na <- na.omit(yes24)

yes24\_na <- data.frame(yes24\_na, stringsAsFactors = F)

str(yes24\_na)

#Factor를 num으로 변경 (에러 해결해서 안쓰임)

#yes24\_na$PageNum <- as.numeric(as.character(yes24\_na$PageNum))

#yes24\_na$PageNum <- suppressWarnings(as.numeric(as.character(yes24\_na$PageNum)))

yes24\_na

# 기초 통계량 요약

summary(yes24\_na)

# 산점도(scatter plot)

plot(yes24\_na$PageNum, yes24\_na$Price, xlab="Page Number", ylab="Price", pch="+")

plot(yes24\_na$Weight, yes24\_na$Price, xlab="Weight", ylab="Price", pch="+")

plot(yes24\_na$Volume, yes24\_na$Price, xlab="Volume", ylab="Price", pch="+")

#jitter

plot(jitter(yes24\_na$PageNum), jitter(yes24\_na$Price), xlab="Page Number", ylab="Price", cex=.2)

# 가격과 페이지 수에 대한 피어슨 상관 계수

cor(yes24\_na$Price, yes24\_na$PageNum)

# 가격과 무게

cor(yes24\_na$Price, yes24\_na$Weight)

# 가격과 부피

cor(yes24\_na$Price, yes24\_na$Volume)

# 통계적 유의성 검증

cor.test(yes24\_na$Price, yes24\_na$PageNum, method = "pearson")

cor.test(yes24\_na$Price, yes24\_na$Weight, method = "pearson")

cor.test(yes24\_na$Price, yes24\_na$Volume, method = "pearson")

# 단순선형회귀 분석

# 가격 - 페이지 수

m <- lm(yes24\_na$Price ~ yes24\_na$PageNum, yes24\_na)

m

summary(m)

plot(yes24\_na$PageNum, yes24\_na$Price, xlab="Page Number", ylab="Price", pch="+")

abline(coef(m))

# 가격 - 무게

m <- lm(yes24\_na$Price ~ yes24\_na$Weight, yes24\_na)

m

summary(m)

plot(yes24\_na$Weight, yes24\_na$Price, xlab="Weight", ylab="Price", pch="+")

abline(coef(m))

# 가격 - 부피

m <- lm(yes24\_na$Price ~ yes24\_na$Volume, yes24\_na)

m

summary(m)

plot(yes24\_na$Volume, yes24\_na$Price, xlab="Volume", ylab="Price", pch="+")

abline(coef(m))

# 중선형회귀 분석

m <- lm(yes24\_na$Price ~ yes24\_na$PageNum + yes24\_na$Weight + yes24\_na$Volume, data = yes24\_na)

m

coef(m)

summary(m)

# 책 가격과 책의 밀도 간의 상관 관계

c <- yes24\_na$Weight/yes24\_na$Volume

cor(yes24\_na$Price, c)

**\* Python3 코드 (데이터 마이닝)**

from urllib.request import Request, urlopen

import urllib.parse

from bs4 import BeautifulSoup

import csv

def Yes24\_Crawling():

csv\_file = open("yes24.csv","w")

cw = csv.writer(csv\_file, delimiter=',')

cw.writerow(["Name","Price","PageNum","Weight","Volume"])

for n in range(1,250):

html = Request('http://www.yes24.com/24/category/bestseller?CategoryNumber=001&sumgb=06&fetchSize=40&PageNumber=' + str(n), headers={'User-Agent':'Mozilla/5.0'})

webpage = urlopen(html).read()

soup = BeautifulSoup(webpage, from\_encoding="utf-8")

bestsellers = soup.find\_all("div" ,class\_="goodsImgW")

print(len(bestsellers))

bestsellerLink = []

for i in bestsellers:

if i.find\_all("a") is not None :

bestsellerLink.append('http://www.yes24.com/'+i.find("a").get("href"))

for i in range(len(bestsellerLink)):

print("페이지 : " + bestsellerLink[i])

detailhtml = Request(bestsellerLink[i])

detailpage = urlopen(detailhtml).read()

detailsoup = BeautifulSoup(detailpage, from\_encoding="utf-8")

if detailsoup.find("span", class\_="cname") is not None: # 19세 이상 도서면 건너 뛰기

bookname = detailsoup.find("span", class\_="cname").next\_sibling.next\_element.next\_element.next\_element

else :

continue

print("책 제목 : " + bookname)

price = detailsoup.find("th", scope="row").next\_sibling.next\_element.next\_element.strip()

price = price.replace(",","")

price = price.replace("원","")

print("가격 : " + price)

if detailsoup.find("p", class\_="pdSize") is not None:

pdSize = detailsoup.find("p", class\_="pdSize").next\_element

else :

continue

print("책 페이지 정보 : " + pdSize)

if "쪽" not in pdSize:

continue

pdSizeList = pdSize.split("|")

bookpage = "NA"

bookweight = "NA"

bookvolume = "NA"

for j in range(len(pdSizeList)):

if "쪽" in pdSizeList[j] and '확인' not in pdSizeList[j]:

bookpage = pdSizeList[j]

bookpage = bookpage.replace("쪽","")

bokkpage = bookpage.replace(" ","")

elif "g" in pdSizeList[j]:

bookweight = pdSizeList[j]

bookweight = bookweight.replace(" ","")

bookweight = bookweight.replace("g","")

else :

volumeList = []

pdSizeList[j] = pdSizeList[j].replace("mm","")

pdSizeList[j] = pdSizeList[j].replace(" ","")

volumeList = pdSizeList[j].split("\*")

if len(volumeList) != 3 :

continue

else:

bookvolume = (int)(volumeList[0]) \* (int)(volumeList[1]) \* (int)(volumeList[2])

print("페이지 수 : " + bookpage)

print("무게 : " + bookweight)

print("부피 : " + str(bookvolume))

csv\_file = open("yes24.csv","a")

cw = csv.writer(csv\_file, delimiter=',')

bookname = bookname.strip()

price = str(price).strip()

bookpage = str(bookpage).strip()

bookweight = str(bookweight).strip()

bookvolume = str(bookvolume).strip()

cw.writerow([bookname, str(price), str(bookpage), str(bookweight), str(bookvolume)])

if \_\_name\_\_ == '\_\_main\_\_' :

Yes24\_Crawling()