2018 정보통계보험수리학과 빅데이터 경진대회

New York taxi data analysis

(날씨에 따른 뉴욕 교통 혼잡도 변화 분석)

팀명 : 소통

팀원 : 소혜빈, 윤홍빈, 이진영, 이효진

요약

어떠한 도시가 악천후에도 원활한 교통을 유지하는 것은 중요한 문제이다. 본 분석에서는 2013년의 New York taxi data와 weather data를 이용해 뉴욕의 날씨에 따른 교통 혼잡도 변화를 알아보고자 한다. 교통 혼잡도를 대표하는 변수로는 뉴욕 택시의 속력을 이용하였으며, 각 날씨 및 출퇴근 시간인지 아닌지의 여부에 따라 택시 속력에 변화가 있는지를 회귀분석을 이용해 검정하였다.

제 1장 서론

1. 분석 계기

택시는 도시 전역에서 광범위하게 운행되는 차량으로 이러한 택시 데이터를 통해 도시의 정보를 알 수 있지 않을까 하는 아이디어를 도출하였다. 그 중 택시의 속력은 해당 도시의 전반적인 교통 체증 정도를 대표할 수 있을 것이라 생각했다.

또한 세계의 각 도시에는 눈 또는 비가 왔을 때에도 원활한 교통을 유지하기 위한 체계가 있다. 그렇다면 뉴욕은 이러한 체계를 잘 갖추고 있는지, 갖추고 있다면 얼마나 잘 갖추고 있는지에 대해 의문을 갖게 되었다.

이에 뉴욕의 날씨에 따른 택시 속력의 변화를 회귀분석으로 검정하여, 악천후에 대비한 뉴욕의 교통 체계가 얼마나 잘 운영되고 있는지 확인해보고자 하였다.

2. 본 분석의 목적과 의의

본 분석은 뉴욕의 날씨에 따른 교통 혼잡도의 변화를 검정하고자 하는 것을 목적으로 한다. 이를 위해 날씨에 따른 뉴욕 택시의 속력 변화를 회귀분석을 이용해 검정한다. 날씨에 따른 뉴욕 택시의 속력 변화가 없거나 작을수록 뉴욕은 악천후에 대비한 교통 체계가 잘 갖춰져 있다고 볼 수 있을 것이다.

이러한 정보는 향후 뉴욕 교통 체계의 개선에 사용될 수 있다. 또한 해당 분석 방법론을 다른 도시에도 적용한다면 교통 정책의 평가 또는 도시들 간의 교통 체계 비교 평가에 사용될 수 있을 것이다.

제 2장 본론

1. 분석 가설의 설정

.- 귀무가설 : 날씨(비, 눈)는 교통체증에 영향을 준다.(속력 DOWN)

- 대립가설 : 날씨는 교통체증에 영향을 주지 않는다.(속력 UP)

2. 데이터 수집 및 처리과정

mapreduce분석에 앞서 우리는 mapreduce함수를 이용하여 하고자 하는 연산을 최대한 map함수를 이용하여 name node의 메모리를 최대한 적게 사용하고자 함에 목적을 두었다.

1) map함수를 이용한 전처리

날씨 data같은경우 1년의 data이기때문에 356개의 데이터로 이루어져있어 mapreduce로 하지않고 전처리를 해주었다.

그외에 taxi데이터의 전처리는 map함수로 모두 처리해주었다.

① 택시데이터 전처리 과정

[데이터 정제 전]

```
> summary(values(from.dfs(mr)))
18/12/18 16:09:51 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptr
erval = 0 minutes.
Deleted /tmp/file6792126dc7aa
  medallion         hack_license       vendor_id    payment_type  fare_amount       surcharge
 Length:738831      Length:738831      CMT:371980   CRD:387503   Min.   :  2.50   Min.   :0.0000
 Class :character   Class :character   VTS:366851   CSH:348892   1st Qu.:  6.50   1st Qu.:0.0000
 Mode  :character   Mode  :character                DIS:   501   Median :  9.00   Median :0.0000
                                                   NOC:  1624   Mean   : 11.66   Mean   :0.3201
                                                   UNK:   311   3rd Qu.: 13.00   3rd Qu.:0.5000
                                                                Max.   :450.00   Max.   :8.5000

    mta_tax         tip_amount      tolls_amount      total_amount      rate_code      store_and_fwd_flag
 Min.   :0.0000   Min.   :  0.00   Min.   : 0.0000   Min.   :  2.50   1      :722833   :366871
 1st Qu.:0.5000   1st Qu.:  0.00   1st Qu.: 0.0000   1st Qu.:  7.70   2      : 11887   N:363599
 Median :0.5000   Median :  0.80   Median : 0.0000   Median : 10.50   5      :  1984   Y:  8361
 Mean   :0.4984   Mean   :  1.27   Mean   : 0.2015   Mean   : 13.95   4      :  1205
 3rd Qu.:0.5000   3rd Qu.:  2.00   3rd Qu.: 0.0000   3rd Qu.: 15.50   3      :   872
 Max.   :0.5000   Max.   :147.50   Max.   :20.0000   Max.   :450.50   0      :    30
                                                                      (Other):    20
 pickup_datetime    dropoff_datetime   passenger_count trip_time_in_secs trip_distance    pickup_longitude
 Length:738831      Length:738831      Min.   :0.000   Min.   :    0.0   Min.   : 0.000   Min.   :-1930.04
 Class :character   Class :character   1st Qu.:1.000   1st Qu.:  360.0   1st Qu.: 1.000   1st Qu.:  -73.99
 Mode  :character   Mode  :character   Median :1.000   Median :  553.0   Median : 1.700   Median :  -73.98
                                       Mean   :1.699   Mean   :  682.8   Mean   : 2.769   Mean   :  -72.64
                                       3rd Qu.:2.000   3rd Qu.:  883.0   3rd Qu.: 3.050   3rd Qu.:  -73.97
                                       Max.   :6.000   Max.   :10680.0   Max.   :98.100   Max.   :   80.84

 pickup_latitude    dropoff_longitude dropoff_latitude
 Min.   :-3084.26   Min.   :-740.02   Min.   :-3117.57
 1st Qu.:   40.74   1st Qu.: -73.99   1st Qu.:   40.73
 Median :   40.75   Median : -73.98   Median :   40.75
 Mean   :   40.02   Mean   : -72.60   Mean   :   39.99
 3rd Qu.:   40.77   3rd Qu.: -73.96   3rd Qu.:   40.77
 Max.   :  473.99   Max.   :  80.84   Max.   :  400.78
                    NA's   :4         NA's   :4
```

[데이터 정제 후]

```
> summary(values(from.dfs(m)))
  medallion         hack_license       vendor_id    payment_type  fare_amount       surcharge         mta_tax
 Length:700383      Length:700383      CMT:352743   CRD:369479   Min.   :  2.50   Min.   :0.0000   Min.   :0.000
 Class :character   Class :character   VTS:347640   CSH:329308   1st Qu.:  6.50   1st Qu.:0.0000   1st Qu.:0.500
 Mode  :character   Mode  :character                DIS:   358   Median :  9.00   Median :0.0000   Median :0.500
                                                   NOC:  1014   Mean   : 11.63   Mean   :0.3226   Mean   :0.499
                                                   UNK:   224   3rd Qu.: 13.00   3rd Qu.:0.5000   3rd Qu.:0.500
                                                                Max.   :323.00   Max.   :8.5000   Max.   :0.500

   tip_amount       tolls_amount      total_amount     rate_code      store_and_fwd_flag pickup_datetime
 Min.   :  0.000   Min.   : 0.0000   Min.   :  3.00   1:687315    :347640           Length:700383
 1st Qu.:  0.000   1st Qu.: 0.0000   1st Qu.:  7.70   2: 10412   N:345022           Class :character
 Median :  0.900   Median : 0.0000   Median : 10.50   3:   770   Y:  7721           Mode  :character
 Mean   :  1.268   Mean   : 0.2008   Mean   : 13.92   4:  1070
 3rd Qu.:  2.000   3rd Qu.: 0.0000   3rd Qu.: 15.50   5:   807
 Max.   :100.000   Max.   :19.8500   Max.   :327.80   6:     9
 dropoff_datetime   passenger_count trip_time_in_secs trip_distance    pickup_longitude   pickup_latitude
 Length:700383      Min.   :1.000   Min.   :   8.0    Min.   : 0.090   Min.   :-1930.04   Min.   :-0.01891
 Class :character   1st Qu.:1.000   1st Qu.: 360.0    1st Qu.: 1.080   1st Qu.:  -73.99   1st Qu.:40.73689
 Mode  :character   Median :1.000   Median : 554.0    Median : 1.790   Median :  -73.98   Median :40.75384
                    Mean   :1.707   Mean   : 683.5    Mean   : 2.835   Mean   :  -73.98   Mean   :40.75068
                    3rd Qu.:2.000   3rd Qu.: 883.0    3rd Qu.: 3.100   3rd Qu.:  -73.97   3rd Qu.:40.76796
                    Max.   :6.000   Max.   :9509.0    Max.   :81.700   Max.   :    0.00   Max.   :73.98441
 dropoff_longitude dropoff_latitude      speed
 Min.   :-740.02   Min.   :-3117.57   Min.   : 5.000
 1st Qu.: -73.99   1st Qu.:   40.74   1st Qu.: 9.375
 Median : -73.98   Median :   40.75   Median :12.203
 Mean   : -73.97   Mean   :   40.74   Mean   :13.785
 3rd Qu.: -73.97   3rd Qu.:   40.77   3rd Qu.:16.271
 Max.   :   0.00   Max.   :  400.78   Max.   :84.944
```

- 정확한 데이터 분석을 위해서 분석에 영향을 줄 만한 대표적인 이상

데이터들(trip_distance, trip_time_in_secs, passenger_count =0)을 일단 제외해 주었다.

- rate_code의 값이 1~6을 벗어나 0또는 그 외의 값들이 있었기 때문에 rate_code = 1~6으로 지정해주었다.

- mta_tax, tip_amount, fare_amount와 같이 비용에 관련된 값이 0보다 작은 경우를 제외해주었다.

- 택시를 타고 내린 위도, 경도가 0인경우를 제외해 주었다.

- 뉴욕의 택시는 기본요금이 $2.5인 것을 감안하여 total_amount(전체요금)이 $2.5보다 낮은 데이터도 제외해 주었다.

- 속력(trip_distance/(trip_time_in_secs/3600)을 구한 후 speed < 5(mile), speed >= 85(mile)인 경우는 비상식적으로 느리거나 빠르다고 생각했기 때문에 처리해 주었다

② 날씨 데이터 전처리

[변수명 지정 전]

```
> summary(ress)
      V1                 V2                 V3                 V4                V5                V6                V7
 Length:3469        Length:3469        Length:3469        Min.   : 0.220   Min.   :0.000   Min.   : 0.0000   Min.   : 0.0000
 Class :character   Class :character   Class :character   1st Qu.: 3.800   1st Qu.:0.000   1st Qu.: 0.0000   1st Qu.: 0.0000
 Mode  :character   Mode  :character   Mode  :character   Median : 5.140   Median :0.000   Median : 0.0000   Median : 0.0000
                                                          Mean   : 5.581   Mean   :0.135   Mean   : 0.1096   Mean   : 0.5455
                                                          3rd Qu.: 6.930   3rd Qu.:0.050   3rd Qu.: 0.0000   3rd Qu.: 0.0000
                                                          Max.   :22.820   Max.   :5.810   Max.   :27.3000   Max.   :23.0000
                                                          NA's   :36       NA's   :1       NA's   :1         NA's   :1
      V8               V9
 Min.   : 13.00   Min.   :-1.00
 1st Qu.: 48.00   1st Qu.:36.00
 Median : 64.00   Median :49.00
 Mean   : 62.98   Mean   :48.69
 3rd Qu.: 79.00   3rd Qu.:64.00
 Max.   :104.00   Max.   :84.00
 NA's   :1        NA's   :1
```

- 이 때의 데이터는 변수명이 정확히 명시되있지 않았기 때문에 따로 변수명을 지정해 주었다.
- original data에는 2009~2018년의 데이터가 모두 들어가 있었기 때문에 우리가 사용하는 taxi data에 맞는 2013년의 데이터만 따로 추출해 주었다.

[변수명 지정, 2013년 데이터 추출 후]

```
> colnames(ress) <- c("STATION","NAME","DATE","AWND","PRCP","SNOW","SNWD","TMAX","TMIN")
> weather <- ress[-1,]
> weather <- weather[weather$DATE>="2013-01-01" & weather$DATE<="2013-12-31",]
> summary(weather)
   STATION             NAME              DATE                AWND            PRCP            SNOW            SNWD
 Length:365         Length:365         Length:365         Min.   : 0.890   Min.   :0.0000   Min.   :0.0000   Min.   : 0.000
 Class :character   Class :character   Class :character   1st Qu.: 3.800   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.000
 Mode  :character   Mode  :character   Mode  :character   Median : 5.140   Median :0.0000   Median :0.0000   Median : 0.000
                                                          Mean   : 5.496   Mean   :0.1269   Mean   :0.0811   Mean   : 0.177
                                                          3rd Qu.: 6.710   3rd Qu.:0.0400   3rd Qu.:0.0000   3rd Qu.: 0.000
                                                          Max.   :15.430   Max.   :4.1600   Max.   :6.3000   Max.   :11.000

      TMAX            TMIN
 Min.   :20.00   Min.   :11.00
 1st Qu.:47.00   1st Qu.:35.00
 Median :64.00   Median :49.00
 Mean   :62.38   Mean   :48.52
 3rd Qu.:79.00   3rd Qu.:64.00
 Max.   :98.00   Max.   :83.00
```

- 날씨 데이터에선 이상 값을 제외하거나 할 필요없이 우리가 필요한 데이터(일별로 눈, 비가 왔는지)를 추출하였다(맑음 : 0, 비 : 1, 눈 : 2)

```
> summary(rain_snow2)
  pickup_date            weather
 Min.   :2013-01-01   Min.   :0.0000
 1st Qu.:2013-04-02   1st Qu.:0.0000
 Median :2013-07-02   Median :0.0000
 Mean   :2013-07-02   Mean   :0.3589
 3rd Qu.:2013-10-01   3rd Qu.:1.0000
 Max.   :2013-12-31   Max.   :2.0000
```

추출된 데이터는 택시 데이터와 날짜를 기준으로 map 함수 내에서 합쳐주었다.

```
res <- res %>% select(pickup_date, pickup_time, speed)
res <- merge(res, weather.dat, by="pickup_date")
```

res는 위 각종 전처리를 마친 택시데이터이며, 합치기 전에 필요한 변수들 (위에 보다시피 pickup_date, pickup_time, speed) 만 뽑아낸 상태.

[World & Now] 뉴욕, 교통체증과의 전쟁

홍인혁 | 입력 : 2018.04.09 17:10:45 수정 :2018.04.09 17:19:23

`미국 금융수도`로 불리는 뉴욕 맨해튼 미드타운에서 택시를 타고 5마일(약 8㎞) 떨어져 있는 약속장소로 이동한다면 얼마나 걸릴까. 딥은 평균 1시간 이상이다. 맨해튼 시내의 악명 높은 교통체증으로 인해 택시의 평균 시속이 4.7마일(2016년 기준)에 불과하기 때문이다. 2012년엔 평균 6.7마일이었는데 해마다 차량의 이동 속도가 떨어지고 있다. 교통정보분석업체 인릭스에 따르면 미국 전역에서 가장 막히는 도로 중 하나가 맨해튼 피프스 애비뉴다.

아래 그림에는 speed의 min이 5라고 나와있으나 추후에 사전정보를 이용하여 최소값을 2로 설정하였다.

```
> summary(values(from.dfs(m2)))
  pickup_date          pickup_time        speed
 Min.   :2013-01-01   Min.   : 0.00   Min.   : 5.00
 1st Qu.:2013-03-28   1st Qu.: 9.00   1st Qu.: 9.00
 Median :2013-06-24   Median :14.00   Median :12.00
 Mean   :2013-06-28   Mean   :13.52   Mean   :13.56
 3rd Qu.:2013-09-30   3rd Qu.:19.00   3rd Qu.:16.19
 Max.   :2013-12-31   Max.   :23.00   Max.   :85.00
```

데이터 전처리를 모두 해준후, 우리는 peak시간대와 nonpeak시간대의 교통량 뿐만아니라 더불어 날씨가 미치는 교통체증이 다를것이라고 생각하였다. 따라서 peaktime을 7:00~9:00, 17:00~19:00 로 지정하여 데이터를 peak.dat로 분류해주었다. 그외 나머지 시간대를 non_peaktime으로 지정하여 마찬가지로 non_peak.dat로 분류하였다.

회귀분석에 필요한 X matrix를 lm함수와 model.metrix함수를 이용하여 도출해내었다. peak시간대의 X matrix는 X.peak로 , non_peak시간대의 X matrix는 X.non_peak로 지정해주었다.

beta hat을 구하기 위해 crossprod 함수를 사용하여 XtX, XtY, YtY를 peak, nonpeak 따로따로 구해주었고, 제곱합을 구해주기 위해 Jmatrix를 구해준후 YtJY를 도출해내었다 REDUCE함수를 사용해 최대한 beta hat이나, 제곱합을 결과값으로 내고 싶었으나, 어려움이 있어 따라서 어쩔수 없이 peak와 non_peak의 XtX,XtY,YtY를 결과값으로 낸 후, local 에서 solve와 그외 연산을 해주었다. 하지만, XtX와 XtY 그리고 YtY는 데이터가 그리 크지 않기 때문에 local에서 충분히 연산을 해주어도 된다고 판단하였다.

```
> SST.peak =v$peak.YtY - v$YtJY.peak;SST.peak
         [,1]
[1,] 7260718
> SST.non_peak=v$non_peak.YtY - v$YtJY.non_peak;SST.non_peak
         [,1]
[1,] 22282851
>
> peak.SSE<-v$peak.YtY-t(peak.beta.hat)%*%v$peak.XtY;peak.SSE
         [,1]
[1,] 7179310
> non_peak.SSE<-v$non_peak.YtY-t(non_peak.beta.hat)%*%v$non_peak.XtY;non_peak.SSE
         [,1]
[1,] 22199123
>
> (peak.SSR<-SST.peak-peak.SSE)
         [,1]
[1,] 81408.14
> (non_peak.SSR<-SST.non_peak-non_peak.SSE)
         [,1]
[1,] 83727.57
>
>
> (peak.MSR<-peak.SSR/(3-1))
         [,1]
[1,] 40704.07
> (peak.MSE<-peak.SSE/(peak.n-3))
         [,1]
[1,] 33.12651
> (peak.F<-peak.MSR/peak.MSE)
         [,1]
[1,] 1228.746
> (non_peak.MSR<-peak.SSR/(3-1))
         [,1]
[1,] 40704.07
> (non_peak.MSE<-peak.SSE/(non_peak.n-3))
         [,1]
[1,] 14.84393
> (non_peak.F<-non_peak.MSR/non_peak.MSE)
         [,1]
[1,] 2742.136
> (Rsq_peak<-peak.SSR/SST.peak)
           [,1]
[1,] 0.01121213
> (Rsq_non_peak<-non_peak.SSR/SST.non_peak)
            [,1]
[1,] 0.003757489
```

제 3장 결론

1. 결과의 시사점

많은 오류로, (그것은 전처리나 인식하지 못한 논리적 오류 일 수 있다.) 현재까지의 결과는 예상과는 달리 R^2 값이 0에 굉장히 가까워 날씨가 택시의 속력에 영향을 미친다는 것을 회귀분석을 통해 설명하기에는 힘들었다. 우리가 설정한 회귀선에 설명력이 매우 낮았다.

2. 분석의 한계

- 데이터가 생각보다 많이 민감해서 변수를 조금만 다른 순간에 정의해도 코드 상의 논리는 틀리지 않았는데 결과가 완전히 잘못 나오는 경우도 발생했다.

```
keyval(1,
       list(peak.XtX, peak.XtY,
            peak.YtY, non_peak.XtX,
            non_peak.XtY, non_peak.YtY,
            YtJY.peak,YtJY.non_peak))
```

```
# res<-list(peak.XtX, peak.XtY,
#      peak.YtY, non_peak.XtX,
#      non_peak.XtY, non_peak.YtY,
#      YtJY.peak,YtJY.non_peak)
#
# keyval(1,res)
```

(왼쪽, 정상적인 값, 오른쪽, NULL)

- 위도와 경도가 0인 데이터들을 제외하긴 했지만 뉴욕을 벗어나 있는 위도, 경도의 값은 제외하지 못한점이 아쉬웠다.

- files[1]을 이용해 테스트하면서 코드를 완성했는데, 전체 파일을 돌려보니 결과가 NULL이 나왔다.

날씨 데이터를 가변수화 한 것이 문제였던 것으로 추측한다.

	pickup_date	weather
1	2013-01-25	2
2	2013-02-02	2
3	2013-02-03	2
4	2013-02-05	2
5	2013-02-08	2
6	2013-02-09	2
7	2013-03-08	2
8	2013-03-16	2
9	2013-03-18	2
10	2013-12-08	2
11	2013-12-10	2
12	2013-12-14	2
13	2013-12-17	2

하둡 분산 시스템에 의해 쪼개진 데이터들 각각에 map 함수가 동작할 때, 날씨 정보와 merge한다. 그 후, Merge된 자료가 map 함수에서 날씨에 따른 속력으로 분석되어지고

그 결과가 model.matrix로 나오게 된다. 이때, 날씨 데이터에 눈이 오는 날짜는 오직 이때 뿐이므로, 많은 분산 클러스터들에서 변수가 하나인 모델 매트릭스를 만들어 낼 것이다. 그러나 일부 클러스터들은 변수가 2개인 모델 매트릭스를 만들었을 것이다. 그렇게 각각 구해진 XtX 행렬들은 행과 열의 개수가 2x2, 3x3으로 다르기 때문에, reduce에서 정상적으로 합쳐질 수가 없다.

(자세한 설명은 다음 – 겨울과 겨울이 아닌 때 ...) 만약 이 문제를 해결하거나, (예를 들어 모델 매트릭스를 직접 만들어 사용하는 것이다. 그러나 시간이 충분하지 못했다.) 가변수가 아닌 연속변수를 사용했다면 전체 데이터를 분석할 수 있었을 것이다. 그랬을 때 결과는 우리의 예측과 더 가까웠을 수도 있다.

-겨울와 겨울이 아닌 때 (눈이 온 달과 눈이 오지 않은 달) 파일을 나눠 분석해보았지만 여전히 좋은 결과를 얻을 수 없었다. 하지만 NULL 값이 나오지는 않았다.

```
winter <- c(files[1],files[7],files[8],files[9])
winter
notWinter <-c(files[2],files[3],files[4],files[5],files[6],files[10],files[11],files[12])
notWinter
```

-날씨 데이터에는 강수량과 강설량이 존재했으나, 강우량은 존재하지 않았다. 초반엔 강수량과 강우량을 착각한 채 분석을 진행해오다가, 눈만 내린 날이 전혀 발견되지 않는다는 사실을 발견하고, 원인을 찾아본 결과 강수량은 강우량과 강설량은 합친 값으로, 강설량과는 단위도 달랐다. 아쉽게도 시간이 부족하여 급한데로 두가지 방법을 고안해내어 시도해보았는데,

방법 1) 길이 미끄러운 날과, 안 미끄러운 날. 즉 비나 눈이 온 날과 맑은 날 2개의 가변수를 사용하는 것과,

```
colnames(var_weather) <- c("PRCP")
rain_snow <- data.frame(NULL)
for (i in 1:365){
  if (var_weather[i,1] == 0 ){
    rain_snow <- rbind(rain_snow, 0)
  }else if (var_weather[i,1] > 0 ){
    rain_snow <- rbind(rain_snow, 1)
  }
}
```

방법 2) 수치는 애석하게도 무시하고, 강수량과 강설량이 모두 0일 경우 맑은 날, 강수량이 존재 하지만, 눈이 오지 않는 날을 비가 오는 날로, 나머지 (분명히 비와 눈이 함께 온 날이 존재 하겠지만) 모든 날은 눈이 온 날로 가변수화 하였다.

```
colnames(var_weather) <- c("PRCP","SNOW")
rain_snow <- data.frame(NULL)
for (i in 1:365){
  if (var_weather[i,1] == 0 &  var_weather[i,2] == 0){
    rain_snow <- rbind(rain_snow, 0)
  }else if (var_weather[i,1] > 0 & var_weather[i,2] == 0){
    rain_snow <- rbind(rain_snow, 1)
  }else{
    rain_snow <- rbind(rain_snow, 2)
  }
}
```