SCHOOL OF MANAGEMENT
FUDAN UNIVERSITY

数据挖掘第一次作业
数据清洗与转换

School of Management, Fudan University

1.数据导入以及数据预处理

- 导入数据集
- 删除含义不明确的数据（司机状态为‘V’），以及缺失值
- 删除重复行
- 去掉日期为4月8号的数据
- 数据预处理之后，数据集从350647行减少到332419行
- 将预处理好的数据作为清洗前的数据

```
taxi=read.csv('Taxi Raw Data.csv',sep=',')
dim(taxi)
```

```
## [1] 350647      8
```

```
#去掉v和NA
taxi$RoofBaconStatus[taxi$RoofBaconStatus=='V']=NA
taxi=na.omit(taxi)
taxi$RoofBaconStatus=as.numeric(taxi$RoofBaconStatus)
#去掉重复重复行
taxi= taxi %>% distinct(TaxiID,GpsMeasureTime,.keep_all=T)
#去掉日期为4.8号的一行数据
taxi=taxi[-44070,]
dim(taxi)
```

```
## [1] 332419      8
```

2.将数据按订单整理

```
driver$order=0
taxi$order=0
j=1
for(i in 1:(nrow(taxi)-1)){
    if(taxi$PassengerState[i]==0 & taxi$RoofBaconStatus[i]==0){
      taxi$order[i]=j
      driver$order[driver$TaxiID==taxi[i,]$TaxiID]=j
      if(taxi$PassengerState[i+1]!=0 | taxi$RoofBaconStatus[i+1]!=0){
j=j+1}
    }
    if(taxi[i,]$TaxiID != taxi[i+1,]$TaxiID){j=1}
}
#选出有效的属于订单的行
taxiorder=taxi[taxi$order != 0,]
taxiorder=taxiorder[,-c(2,3,4)]
taxiorder[1:10,]
```

```
##    TaxiID GpsMeasureTime Longitude Lantitude Speed order
## 14  10125  2015/4/1 0:12  121.5593  31.21621   1.6     1
## 15  10125  2015/4/1 0:13  121.5644  31.21685   0.0     1
## 16  10125  2015/4/1 0:14  121.5635  31.21682   7.3     1
## 17  10125  2015/4/1 0:15  121.5575  31.21609   4.2     1
## 18  10125  2015/4/1 0:16  121.5556  31.22397   6.8     1
## 19  10125  2015/4/1 0:17  121.5548  31.22721   0.3     1
```

- 选出属于有效订单的行:

有效订单:“载客+营运”

不属于订单的行，order都为0

- 给统一司机的不同订单进行编号
- 删去一些没用的变量，如onfreeway
- 将所有订单行（及order不为0）存于另一个data.frame
- 从332419行数据中筛选出187418行属于订单的行

2.将数据按订单整理

- 将数据整理为订单的形式
- 总共得到7111个订单

```
##      taxiid order      start.time start.long start.lan        end.time end.long
## 1     10125     1  2015/4/1 0:12   121.5593  31.21621  2015/4/1 0:49 121.5067
## 54    10125     2  2015/4/1 0:54   121.5067  31.26116  2015/4/1 1:02 121.4890
## 369   10125     3  2015/4/1 6:26   121.4367  31.25671  2015/4/1 6:42 121.3952
## 388   10125     4  2015/4/1 6:45   121.3976  31.24167  2015/4/1 7:06 121.4160
## 419   10125     5  2015/4/1 7:16   121.4233  31.19211  2015/4/1 7:26 121.4370
## 434   10125     6  2015/4/1 7:31   121.4387  31.22175  2015/4/1 7:44 121.4538
## 450   10125     7  2015/4/1 7:47   121.4578  31.19837  2015/4/1 8:15 121.4859
## 480   10125     8  2015/4/1 8:17   121.4881  31.24651  2015/4/1 8:36 121.4431
## 500   10125     9  2015/4/1 8:39   121.4429  31.20095  2015/4/1 9:15 121.3231
## 612   10125    10 2015/4/1 10:36   121.3231  31.19592 2015/4/1 11:36 121.6037
##       end.lan night.time day.time dis.night   dis.day  price
## 1    31.25832         37        0 1.6950000 0.0000000 39.700
## 54   31.27885          8        0 0.5716667 0.0000000 21.100
## 369  31.24336          0       16 0.0000000 0.7433333 21.800
## 388  31.18208          0       21 0.0000000 1.4100000 23.360
## 419  31.21679          0       10 0.0000000 0.8833333 19.200
## 434  31.19404          0       13 0.0000000 0.6583333 19.200
## 450  31.24597          0       28 0.0000000 1.4566667 27.000
## 480  31.20089          0       19 0.0000000 1.3066667 21.800
## 500  31.19251          0       36 0.0000000 2.6216667 29.600
## 612  31.24411          0       60 0.0000000 5.2600000 43.344
```

变量名解释：

taxiid：司机ID

order：订单编号

start.time：订单开始时间

start.long：订单开始经度

start.lan：订单开始纬度

end.time：订单结束时间

end.long：订单结束经度

end.lan：订单结束纬度

night.time：订单夜间行驶时间

day.time：订单白天行驶时间

dis.night：夜间行驶距离

dis.day：白天行驶距离

price：应付给司机的价格

3.计算清洗前各指标

```
##   driver.num order order.ave mile.ave income.ave passenger.time work.time
## 1        249  7111  28.55823 307.6357   992.4844         759.49  1128.996
##   vancant.time
## 1    0.3272873
```

变量阐释:

driver.num：司机总数

order：订单总数

order.ave：司机平均每天的订单数

mile.ave：司机平均每天的行驶里程

income.ave：司机平均每天的收入金额

passenger.time：司机平均每天的载客营运时长

work.time：司机平均每天的工作时长

vancant.time：司机平均每天的空驶时间比例

4.进行数据清洗

- 删除订单级别的噪音/孤立点数据

```
#删除订单级别的噪音/孤立点数据
#删除 距离≤0.5km, 时间>360min 的数据
order.clean=order[order$ttldistance > 0.5,]
order.clean=order[order$ttltime >=5 & order$ttltime <= 360,]
#删除 距离/时间(km/min) <0.1, 距离/时间(km/min) >2
order.clean=order.clean[order.clean$ttldistance/order.clean$ttltime >= 0.1 &
                        order.clean$ttldistance/order.clean$ttltime <= 2, ]
#5846
write.csv(order.clean,file='taxi.byorder.cleaned.csv')
```

  - 删除单个订单行驶距离≤0.5km的数据
  - 删除单个订单行驶时间>360min ， 以及时间<5min的数据
  - 删除 距离/时间(km/min) <0.1, 距离/时间(km/min) >2的数据

4.进行数据清洗

- 删除司机级别的噪音/孤立点数据

```
#删除司机级别的噪音/孤立点数据
driver.clean=driver.clean[driver.clean$income < 1500 &
                          driver.clean$income > 200,]
driver.clean=driver.clean[driver.clean$order < 50 &
                          driver.clean$order > 5,]
driver.clean=driver.clean[driver.clean$passenger.time < 720 &
                          driver.clean$passenger.time > 60,]
driver.clean=driver.clean[driver.clean$time.ave < 40 &
                          driver.clean$time.ave > 5,]
driver.clean=driver.clean[driver.clean$dis.ave < 20 &
                          driver.clean$dis.ave > 2,]
driver.clean=driver.clean[driver.clean$speed.ave < 120 &
                          driver.clean$speed.ave > 20,]
#206
write.csv(driver.clean,file='taxi.bydriver.cleaned.csv')
dim(driver.clean)
```

```
## [1] 206   7
```

清洗完后剩下206位司机

5.计算清洗后各指标

```
##        driver.num order order.ave mile.ave income.ave passenger.time work.time
## before        249  7111  28.55823 307.6357   992.4844        759.4900 1128.9960
## after         206  5244  25.45631 209.9601   791.8024        482.3544  873.5291
##        vancant.time
## before    0.3272873
## after     0.4478096
```

变量阐释:

driver.num：司机总数

order：订单总数

order.ave：司机平均每天的订单数

mile.ave：司机平均每天的行驶里程

income.ave：司机平均每天的收入金额

passenger.time：司机平均每天的载客营运时长

work.time：司机平均每天的工作时长

vancant.time：司机平均每天的空驶时间比例

- 可以看到，除了空驶时间占比有所上升，其余指标均有所下降
- 清理完后的数据显然更加合理