Data_Exploration.Rmd

Team 10

November 16, 2015

```
library('data.table')
```

```
## Warning: package 'data.table' was built under R version 3.2.2
```

```
library('ggplot2')
```

```
## Warning: package 'ggplot2' was built under R version 3.2.2
```

```
library('reshape2')
```

```
## Warning: package 'reshape2' was built under R version 3.2.2
```

```
##
## Attaching package: 'reshape2'
##
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library('gridExtra')
```

```
## Warning: package 'gridExtra' was built under R version 3.2.2
```

```
setwd('C:/Users/db345c/Desktop/hw')
df <- read.csv('train.csv')
dt <- data.table(df)

# Get the dimensions and attribute data for the dataset
str(dt)
```

```
## Classes 'data.table' and 'data.frame':   647054 obs. of  7 variables:
##  $ ï..TripType          : int  999 30 30 26 26 26 26 26 26 26 ...
##  $ VisitNumber          : int  5 7 7 8 8 8 8 8 8 8 ...
##  $ Weekday              : Factor w/ 7 levels "Friday","Monday",..: 1 1 1 1
1 1 1 1 1 1 ...
##  $ Upc                  : num  6.81e+10 6.05e+10 7.41e+09 2.24e+09 2.01e+0
9 ...
##  $ ScanCount            : int  -1 1 1 2 2 2 1 1 1 -1 ...
##  $ DepartmentDescription: Factor w/ 69 levels "1-HR PHOTO","ACCESSORIE
S",..: 21 64 52 51 51 51 51 51 51 51 ...
##  $ FinelineNumber       : Factor w/ 5197 levels "","0","1","10",..: 6 4829 2
465 1837 24 24 24 1276 2462 1837 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
# Find the minimum value for each variable
apply(dt,2,min)
```

```
##           ï..TripType           VisitNumber               Weekday
##                 "  3"             "     5"              "Friday"
##                   Upc             ScanCount DepartmentDescription
##                    NA                 "-10"          "1-HR PHOTO"
##        FinelineNumber
##                    ""
```

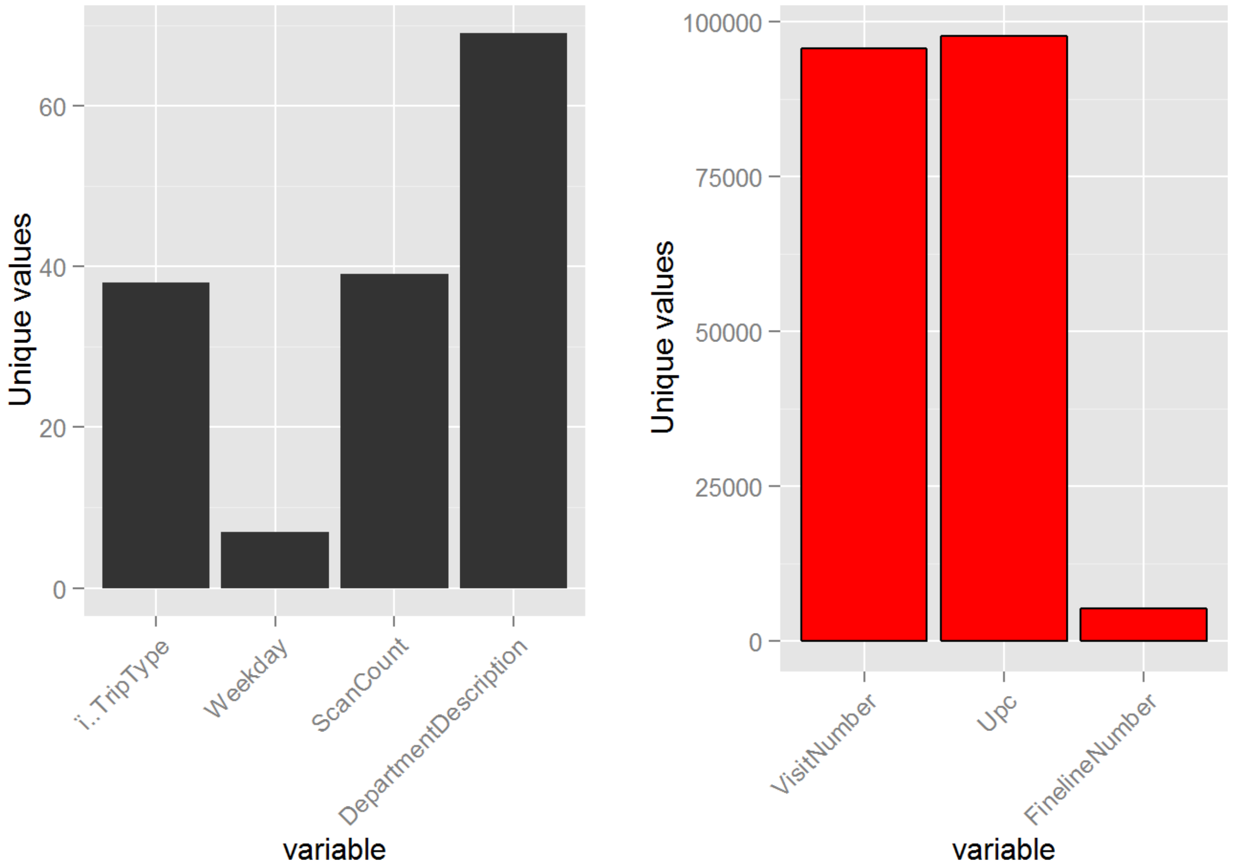
```
# The maximum value for each variable
apply(dt,2,max)
```

```
##           ï..TripType           VisitNumber               Weekday
##                 "999"              "191347"           "Wednesday"
##                   Upc             ScanCount DepartmentDescription
##                    NA                 " 71"            "WIRELESS"
##        FinelineNumber
##                "9998"
```

```
# Plot the number of unique values for each variable
unique_values <- data.table(melt(as.data.frame(lapply(dt,function(x) length(uni
que(x))))))
```

```
## No id variables; using all as measure variables
```

```
p1 <- ggplot(data=unique_values[unique_values$value < 1000,], aes(x = variabl
e, y = value)) +
  geom_bar(stat = "identity") + theme(axis.text.x = element_text(angle = 45, hj
ust = 1)) + ylab('Unique values')
p2 <- ggplot(data=unique_values[unique_values$value > 1000,], aes(x = variabl
e, y = value)) + geom_bar(stat = "identity", fill = "red", color = 'black') + t
heme(axis.text.x = element_text(angle = 45, hjust = 1)) + ylab('Unique values')
grid.arrange(p1, p2, ncol=2)
```

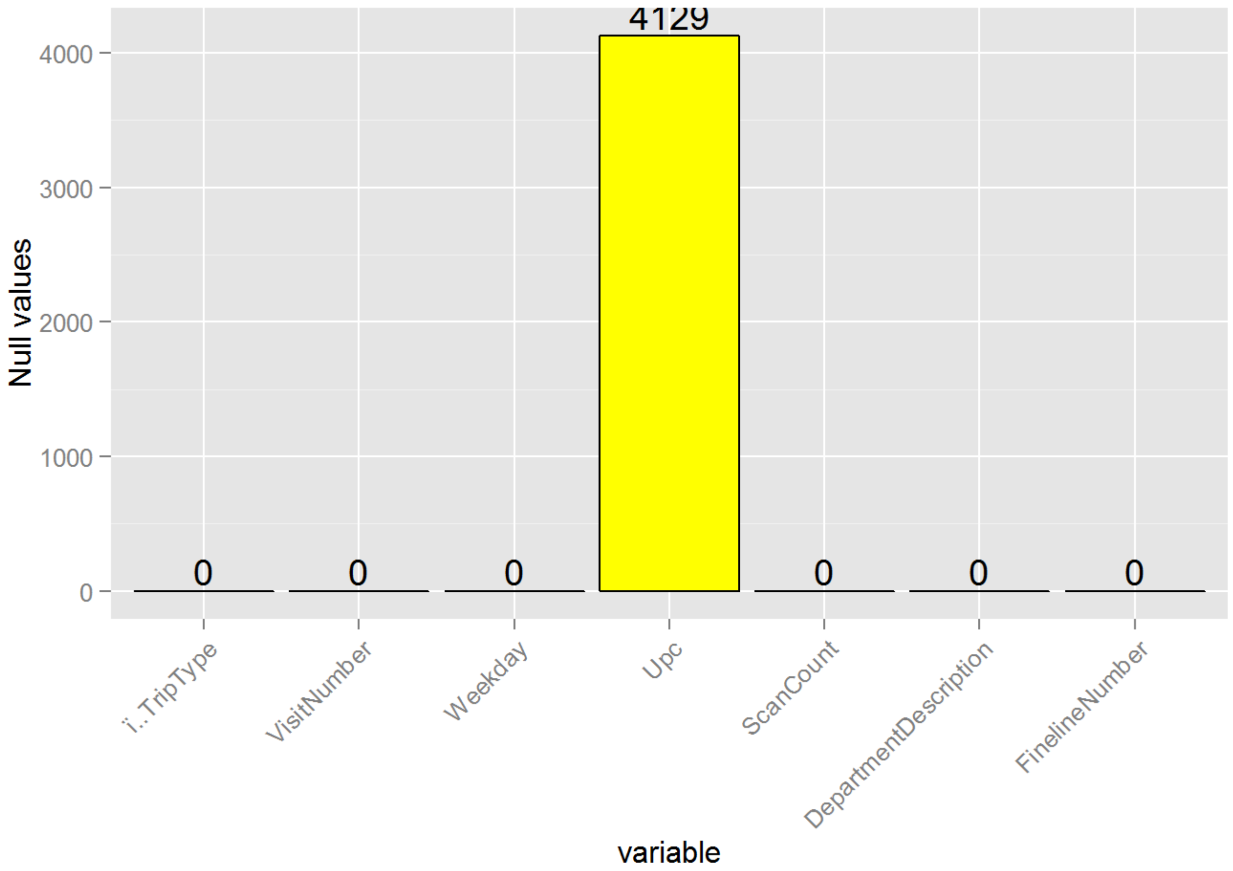


```
# Plot the number of null values for each attribute
null_values <- data.table(melt(as.data.frame(lapply(dt,function(x) sum(is.na
(x))))))
```

```
## No id variables; using all as measure variables
```

```
ggplot(data = null_values, aes(x = variable, y = value)) +
  geom_bar(stat = 'identity', fill = 'yellow', color = 'black') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_text(label = null_values$value, position=position_dodge(width=0.9), vjus
t=-0.25) +
  ylab('Null values')
```

```
## ymax not defined: adjusting position using y instead
```



```
# Clean up data: remove rows with null values, leaving 647,054 - 4,129 = 642,92
5 observations
dt <- dt[complete.cases(dt)]

# Remove duplicated records, on the assumption that multiple item purchases sho
uld
# be reflected as i + 1 ScanCount values, not repeated rows with ScanCount = 1
# This leaves 642,925 - 4,695 =  638,230 observations
dt <- dt[!duplicated(dt)]
nrow(dt)
```

```
## [1] 638230
```

```
# Add weekend column (0 - workday, 1 - Weekend)
print("Started creating 'Weekend' column" )
```

```
## [1] "Started creating 'Weekend' column"
```

```
dt$Weekend = 0
dt[dt$Weekday=="Saturday",]$Weekend = 1
dt[dt$Weekday=="Sunday",]$Weekend = 1
print("Finished creating 'Weekend' column")
```

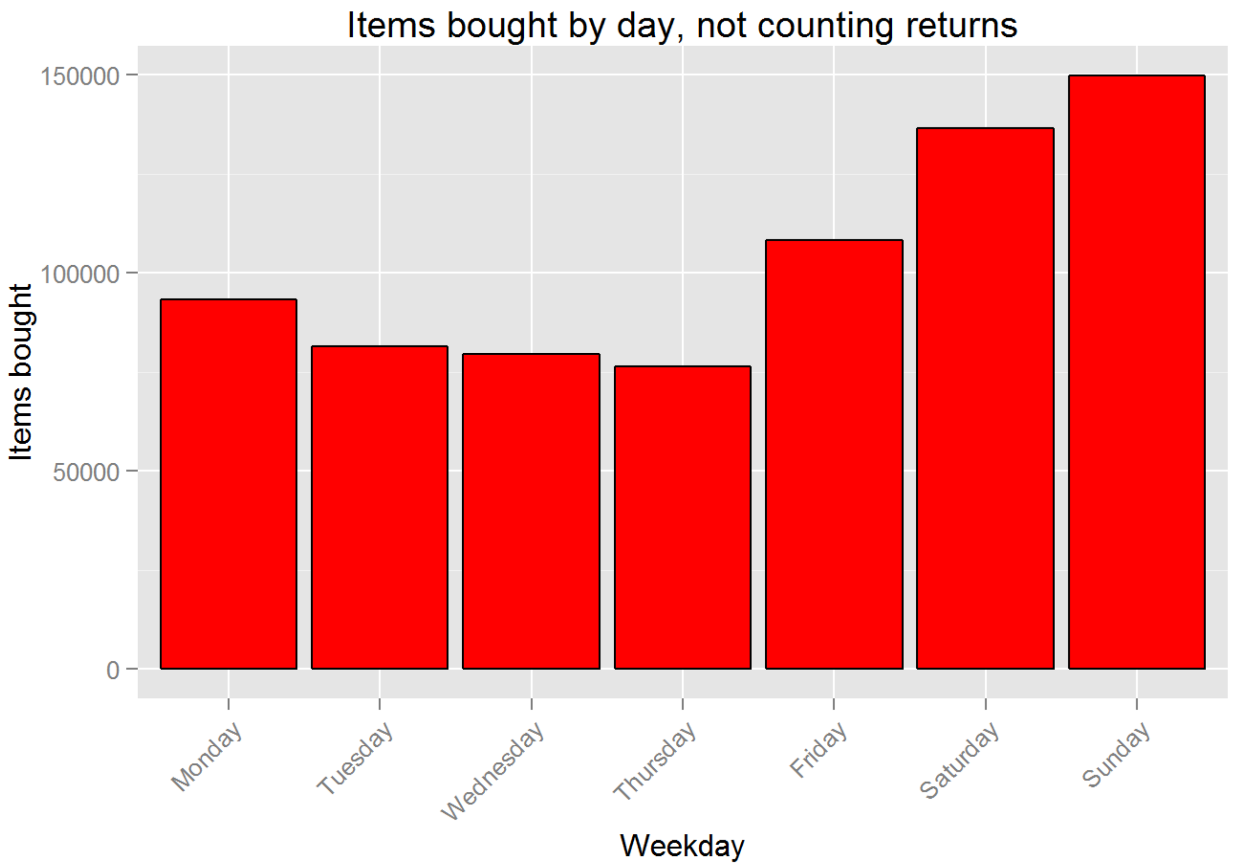
```
## [1] "Finished creating 'Weekend' column"
```

```
# Example exploration:
# total items bought on each day of week, ignoring returns.
# First reorder the weekday factor as per the days of the week.

dt$Weekday <- factor(dt$Weekday, levels= c("Monday","Tuesday", "Wednesday", "Th
ursday", "Friday", "Saturday", "Sunday"))
dt[order(dt$Weekday),]
```

```
##         ï..TripType VisitNumber Weekday         Upc ScanCount
##      1:           7       19709  Monday  7874214098         1
##      2:           7       19709  Monday  4119640482         2
##      3:           7       19709  Monday  7874211700         1
##      4:           7       19709  Monday  4223830241         1
##      5:           7       19709  Monday  4470003050         1
##     ---
## 638226:          39      191346  Sunday 32390001778         1
## 638227:          39      191346  Sunday  7874205336         1
## 638228:          39      191346  Sunday        4072         1
## 638229:           8      191347  Sunday  4190007664         1
## 638230:           8      191347  Sunday  3800059655         1
##         DepartmentDescription FinelineNumber Weekend
##      1:     GROCERY DRY GOODS           3559       0
##      2:     GROCERY DRY GOODS           3108       0
##      3:                 DAIRY           1404       0
##      4:   IMPULSE MERCHANDISE            125       0
##      5:       PRE PACKED DELI           7554       0
##     ---
## 638226:          PHARMACY OTC           1118       1
## 638227:          FROZEN FOODS           1752       1
## 638228:               PRODUCE           4170       1
## 638229:                 DAIRY           1512       1
## 638230:     GROCERY DRY GOODS           3600       1
```

```
ggplot(data = dt[ScanCount > 0 ,sum(ScanCount),by = Weekday], aes(x = Weekday,
y = V1)) +
geom_bar(stat = 'identity', fill = 'red', color = 'black') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylab('Items bought') + ggtitle('Items bought by day, not counting returns')
```



```
# Clean up
gc()
```

```
##           used (Mb) gc trigger  (Mb) max used  (Mb)
## Ncells  561783 30.1    1168576  62.5  1168576  62.5
## Vcells 6658008 50.8   22519097 171.9 28109661 214.5
```