reg2_thinx.R

jyothi

Sun May 21 11:09:49 2017

```
orders_export_1 <- read.csv('/Users/jyothi/Desktop/thinx/orders_export_1.csv', comment.char="~")
orders_export_2 <- read.csv('/Users/jyothi/Desktop/thinx/orders_export_2.csv', comment.char="~")
orders_export <- read.csv('/Users/jyothi/Desktop/thinx/orders_export.csv', comment.char="~")

# Merging three datasets
mergedf <- rbind( orders_export, orders_export_1,orders_export_2 )

#  Remove # sign before Name field
mergedf$Name <- substring(mergedf$Name, 2)
mergedf$Billing.Zip <- substring(mergedf$Billing.Zip, 2)
mergedf$Shipping.Zip <- substring(mergedf$Shipping.Zip, 2)

# Subseting useful columns
subDf <- subset(mergedf, select=c("Name", "Created.at","Lineitem.name","Lineitem.price","Lineitem.quanti

# Selecting only Hiphugger items
p1 <- 'Hiphugger'
df1 <- subset(subDf, grepl(p1,Lineitem.name ) )

summary(df1)
```

```
##      Name                                Created.at
##  Length:57081       2015-12-14 11:15:27 -0500:   14
##  Class :character   2016-03-04 15:44:05 -0500:   13
##  Mode  :character   2016-01-12 10:22:11 -0500:   10
##                     2016-02-08 20:14:40 -0500:    8
##                     2016-02-25 17:06:11 -0500:    7
##                     2015-12-30 11:50:05 -0500:    7
##                     (Other)                  :57022
##                 Lineitem.name   Lineitem.price Lineitem.quantity
##  Hiphugger - M / Black :15130   Min.   :34     Min.   : 1.000
##  Hiphugger - S / Black :11311   1st Qu.:34     1st Qu.: 1.000
##  Hiphugger - L / Black : 9781   Median :34     Median : 1.000
##  Hiphugger - XL / Black: 4593   Mean   :34     Mean   : 1.448
##  Hiphugger - XS / Black: 3228   3rd Qu.:34     3rd Qu.: 2.000
##  Hiphugger - M / Beige : 3029   Max.   :34     Max.   :41.000
##  (Other)               :10009
##  Lineitem.discount
##  Min.   :  0.000
##  1st Qu.:  0.000
##  Median :  0.000
##  Mean   :  2.978
##  3rd Qu.:  3.400
##  Max.   :160.590
##  NA's   :1
```

```r
# Converting Created.at from string to date
df1$Created.date <- as.POSIXct(df1$Created.at ,format= "%Y-%m-%d %H:%M:%S")

# There is one NA in discount.price
df1 <- na.omit(df1)

# Finding Price after discount
attach(df1)
df1$PAD <- with(df1, (Lineitem.price  -(Lineitem.discount/Lineitem.quantity)))
df1$Order.price <- with(df1, (Lineitem.price*Lineitem.quantity)-Lineitem.discount)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(lubridate)
library(car)
# Summary Statistics
OR<- df1 %>% group_by(Created.at.month=floor_date(Created.date)) %>%
summarize(totalsales=sum(Order.price) )
QR<- df1 %>% group_by(Created.at.month=floor_date(Created.date)) %>%
summarize(noofitems=sum(Lineitem.quantity) )
DR<- df1 %>% group_by(Created.at.month=floor_date(Created.date)) %>%
summarize(totaldiscount=sum(Lineitem.discount) )

sum_df <- cbind(OR,QR,DR)
sum_df<- sum_df[ -c(3 , 5) ]
sum_df["Unitprice"]  <- sum_df$totalsales / sum_df$noofitems
# Final dataset for creating a model
sum_df[49082:49092,]
```

```
##          Created.at.month totalsales noofitems totaldiscount Unitprice
## 49082 2016-03-31 23:44:47       86.7         3          15.3      28.9
## 49083 2016-03-31 23:45:56       28.9         1           5.1      28.9
## 49084 2016-03-31 23:46:33       30.6         1           3.4      30.6
## 49085 2016-03-31 23:52:17       34.0         1           0.0      34.0
## 49086 2016-03-31 23:52:36       68.0         2           0.0      34.0
## 49087 2016-03-31 23:53:57       34.0         1           0.0      34.0
## 49088 2016-03-31 23:54:02      144.5         5          25.5      28.9
## 49089 2016-03-31 23:54:25       68.0         2           0.0      34.0
## 49090 2016-03-31 23:56:55       91.8         3          10.2      30.6
## 49091 2016-03-31 23:57:03       54.4         2          13.6      27.2
## 49092 2016-03-31 23:59:05       34.0         1           0.0      34.0
```

```r
### Creating a linear model
linear_model <- lm( sum_df$noofitems ~sum_df$Unitprice)
summary(linear_model)
```

```
##
## Call:
## lm(formula = sum_df$noofitems ~ sum_df$Unitprice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -2.904  -0.353  -0.353   0.647 149.647
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9.224492   0.102384   90.10   <2e-16 ***
## sum_df$Unitprice -0.231522   0.003137  -73.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.482 on 49090 degrees of freedom
## Multiple R-squared:  0.09988,    Adjusted R-squared:  0.09986
## F-statistic:  5447 on 1 and 49090 DF,  p-value: < 2.2e-16
```

```r
# Test to explain Autocorrelation
durbinWatsonTest(linear_model)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.02230079      1.955396   0.002
##  Alternative hypothesis: rho != 0
```

```r
# In this model both intercept and Coefficient are significant. And the model overall significance
# is very high indicated by p-value: < 2.2e-16. Errors are very less for the degree of freedom
# Only problem in this model is R sqaure is very less. But to be a a good model R quare need not be  hi
# Some times even 10% explenation of variantion is also good enough.
# durbinWatsonTest  The output shows that there is no autocorrelation issues in the model
################################################
# The coefficient indicates that for every additional raise in item price by one unit there is decrese
# in demand by  0.23.
################################
```