Assignment 3: Promotions Management

Group 11: Christina Wang, Kailin Fu, Shun Guan, Sylvia Lu, Yiran Huang

April 20, 2020

Contents

1 Promotional event planning

2 Estimating lift factors and promotion ROI analysis

In this part of the assignment, we analyze the effectiveness and ROI of different promotions for Hellman's 32 oz Mayonnaise. The analysis is based on account level data at Jewel-Osco and Dominick's Finer Foods in Chicago. Use the table (data frame) hellmans_df in the file Hellmans.RData. hellmans_DF contains the following variables:

- account
- product
- week
- units
- dollars
- feature_pctacv
- display_pctacv

2.1 Question

Create a price variable for Hellman's 32oz mayo. Then, although not strictly necessary (because the estimated coefficients will scale in a linear regression), you should divide the feature and display columns (variables) by 100. Examine the feature and display variables. Provide summary statistics (number of observations, mean, standard deviation) and histograms of these variables, separately for both accounts. To what extent do these two promotional instruments differ? Calculate the correlations between feature_pctacv, display_pctacv, and price (use the cor function in R). Comment on your findings. Do the correlations indicate a potential problem for your regression analysis to be performed below?

```r
hellmans_df$price = hellmans_df$dollars / hellmans_df$units
hellmans_df$feature = hellmans_df$feature_pctacv / 100
hellmans_df$display = hellmans_df$display_pctacv / 100

my_summary <- function(df, account) {
  df_local = df[df$account == account,]
    list("Feature Summary" =
       list("count" = length(df_local$feature),
            "mean" = mean(df_local$feature),
            "sd" = sd(df_local$feature)),
       "Display Summary" =
       list("count" = length(df_local$display),
            "mean" = mean(df_local$display),
            "sd" = sd(df_local$display)))
}

D_summary = my_summary(hellmans_df, "Dominicks")
J_summary = my_summary(hellmans_df, "Jewel")
print(D_summary)
```

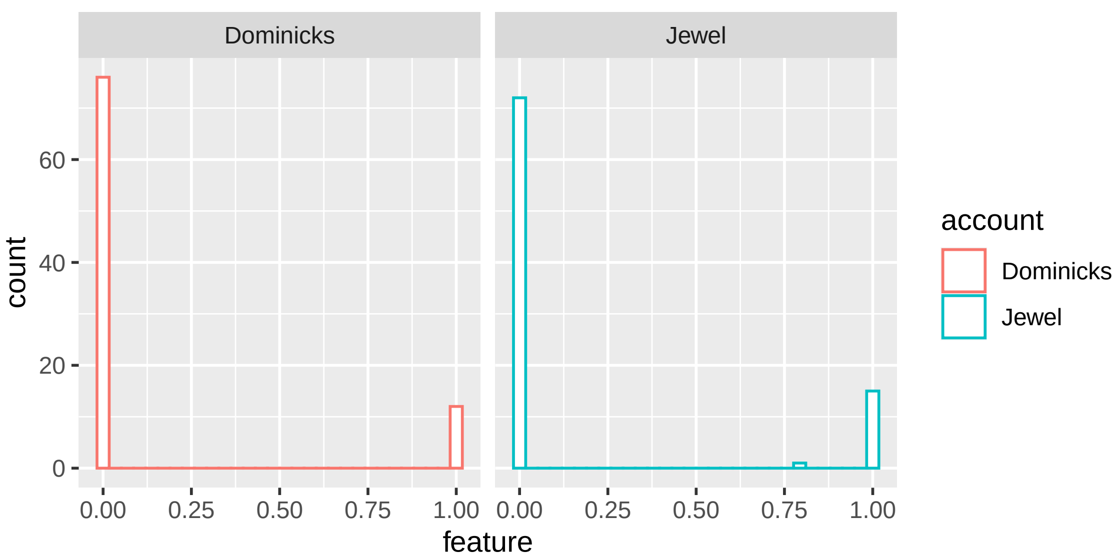
```
$`Feature Summary`
$`Feature Summary`$count
[1] 88

$`Feature Summary`$mean
[1] 0.1363636

$`Feature Summary`$sd
[1] 0.3451409
```

```
$`Display Summary`
$`Display Summary`$count
[1] 88

$`Display Summary`$mean
[1] 0.1206818

$`Display Summary`$sd
[1] 0.1762952
```

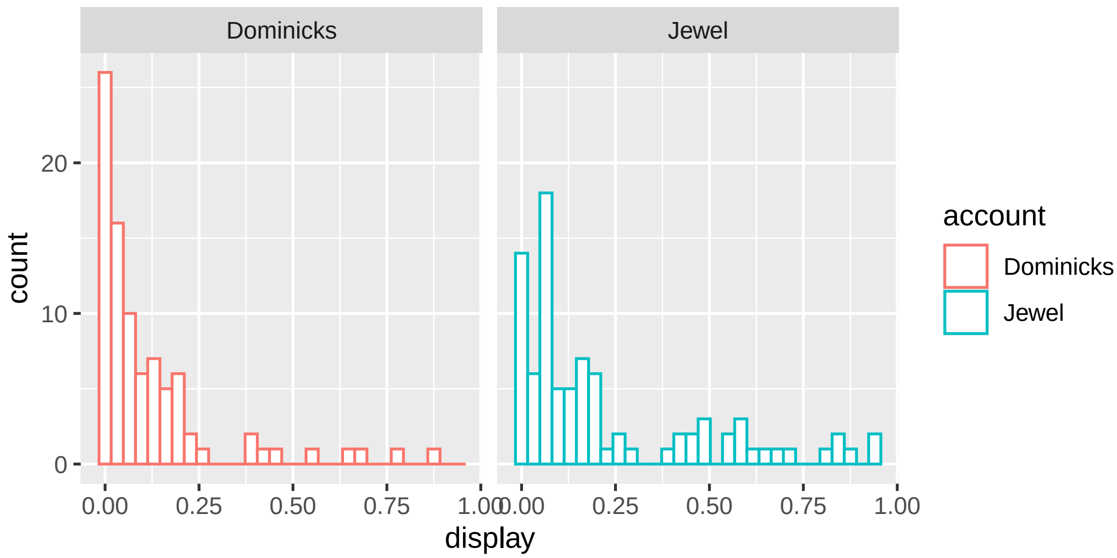
```
print(J_summary)
```

```
$`Feature Summary`
$`Feature Summary`$count
[1] 88

$`Feature Summary`$mean
[1] 0.1794318

$`Feature Summary`$sd
[1] 0.3834338


$`Display Summary`
$`Display Summary`$count
[1] 88

$`Display Summary`$mean
[1] 0.2298864

$`Display Summary`$sd
[1] 0.2581655
```

```
hellmans_df %>%
ggplot(data = ., aes(x=feature, color=account)) + geom_histogram(fill="white") + facet_grid(cols = vars
```



```r
hellmans_df %>%
ggplot(data = ., aes(x=display, color=account)) + geom_histogram(fill="white") + facet_grid(cols = vars
```




```r
#Correlations
cor(hellmans_df$feature_pctacv, hellmans_df$display_pctacv)
```

```
[1] 0.7599992
```

```r
cor(hellmans_df$feature_pctacv, hellmans_df$price)
```

```
[1] -0.5747241
```

```r
cor(hellmans_df$display_pctacv, hellmans_df$price)
```

```
[1] -0.6700056
```

2.2 Question

Estimate the log-linear demand model separately for each account, using price as the only explanatory variable. Then add the feature and display variables. Comment on the difference between the two regressions in terms of goodness of fit, and the price elasticity estimates. Is the change in price elasticity estimates as expected? What is the reason for this change? Are the coefficient estimates similar for both accounts?

```r
D_lm =
hellmans_df %>%
filter(account == "Dominicks") %>%
glm(log(units) ~ log(price), data = .)

summary(D_lm)
```

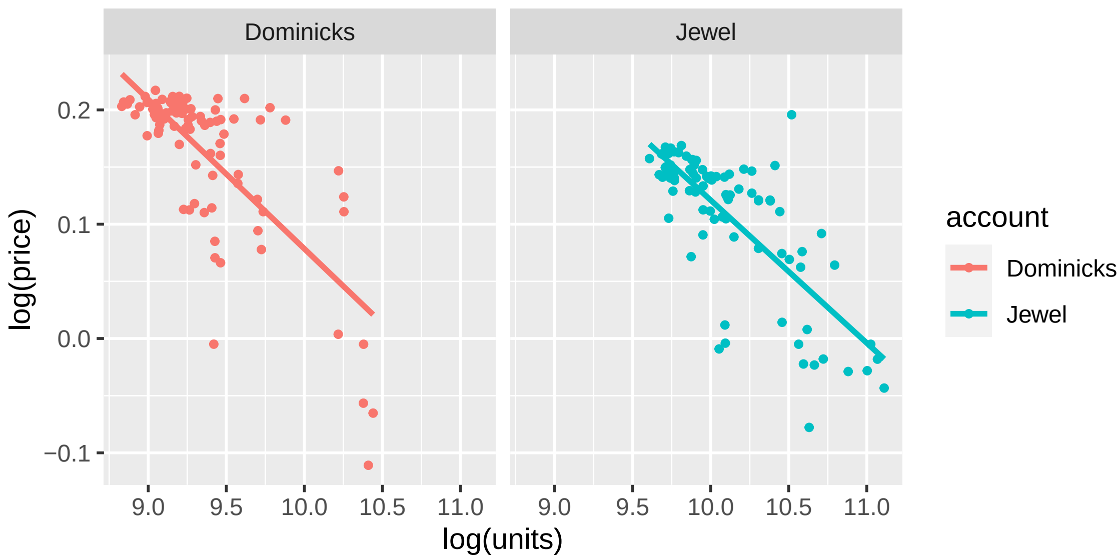
```
Call:
glm(formula = log(units) ~ log(price), data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.63760  -0.17214  -0.01628   0.10558   0.79349

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  10.0368     0.0719  139.60  < 2e-16 ***
log(price)   -4.1665     0.4107  -10.15  2.3e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.06617719)

    Null deviance: 12.5037  on 87  degrees of freedom
Residual deviance:  5.6912  on 86  degrees of freedom
AIC: 14.753

Number of Fisher Scoring iterations: 2
```

```r
J_lm =
hellmans_df %>%
filter(account == "Jewel") %>%
glm(log(units) ~ log(price), data = .)

summary(J_lm)
```

```
Call:
glm(formula = log(units) ~ log(price), data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.59230  -0.16883  -0.03486   0.15152   0.81131

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.60443    0.05259  201.66   <2e-16 ***
log(price)  -4.58359    0.42660  -10.74   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.06161774)

    Null deviance: 12.4124  on 87  degrees of freedom
Residual deviance:  5.2991  on 86  degrees of freedom
AIC: 8.4712

Number of Fisher Scoring iterations: 2
```

```
#Two linear model demand-price graph
hellmans_df %>%
ggplot(data = ., aes(x= log(units), y = log(price), color=account)) + geom_point(size = 1, alpha = 1) +
  facet_grid(cols = vars(account)) + geom_smooth(method = "lm", se = FALSE)
```




```
#Compare with the feature add model

D_d_lm =
hellmans_df %>%
filter(account == "Dominicks") %>%
glm(log(units) ~ log(price) + display, data = .)
summary(D_d_lm)
```

```

Call:
glm(formula = log(units) ~ log(price) + display, data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.43297  -0.14369  -0.02460   0.09584   0.59909

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.61572    0.08888 108.190  < 2e-16 ***
log(price)  -2.36500    0.44187  -5.352 7.25e-07 ***
display      1.07331    0.16833   6.376 9.04e-09 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.04529276)

    Null deviance: 12.5037  on 87  degrees of freedom
Residual deviance:  3.8499  on 85  degrees of freedom
AIC: -17.645

Number of Fisher Scoring iterations: 2
```

```r
J_d_lm =
hellmans_df %>%
filter(account == "Jewel") %>%
glm(log(units) ~ log(price) + display, data = .)
summary(J_d_lm)
```

```
Call:
glm(formula = log(units) ~ log(price) + display, data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.36125  -0.10576  -0.03313   0.09383   0.51352

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.09791    0.06263 161.233  < 2e-16 ***
log(price)  -1.89014    0.39966  -4.729 8.86e-06 ***
display      0.95534    0.09657   9.892 8.47e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.02897969)

    Null deviance: 12.4124  on 87  degrees of freedom
Residual deviance:  2.4633  on 85  degrees of freedom
AIC: -56.941

Number of Fisher Scoring iterations: 2
```

```r
D_d_f_lm =
hellmans_df %>%
filter(account == "Dominicks") %>%
glm(log(units) ~ log(price) + display + feature, data = .)
summary(D_d_f_lm)
```

```
Call:
glm(formula = log(units) ~ log(price) + display + feature, data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.34144  -0.13771  -0.02137   0.11067   0.61078

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.52123    0.08944 106.451  < 2e-16 ***
log(price)  -1.84318    0.45032  -4.093 9.74e-05 ***
display      0.83410    0.17653   4.725 9.14e-06 ***
feature      0.28531    0.08925   3.197  0.00196 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.04086078)

    Null deviance: 12.5037  on 87  degrees of freedom
Residual deviance:  3.4323  on 84  degrees of freedom
AIC: -25.748

Number of Fisher Scoring iterations: 2
```

```
J_d_f_lm =
hellmans_df %>%
filter(account == "Jewel") %>%
glm(log(units) ~ log(price) + display + feature, data = .)
summary(J_d_f_lm)
```

```
Call:
glm(formula = log(units) ~ log(price) + display + feature, data = .)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.36769  -0.12020  -0.02219   0.08526   0.49093

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 10.08881    0.06327 159.450  < 2e-16 ***
log(price)  -1.89735    0.39969  -4.747 8.39e-06 ***
display      1.06947    0.14891   7.182 2.56e-10 ***
feature     -0.09124    0.09062  -1.007    0.317
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.02897501)

    Null deviance: 12.4124  on 87  degrees of freedom
Residual deviance:  2.4339  on 84  degrees of freedom
AIC: -55.997

Number of Fisher Scoring iterations: 2
```

```
# add display improve the model, reduce deviance. However, add feature just improve it a little bit, si
```

2.3 Question

Consider the following three promotions:
(a) 15% TPR
(b) 15% TPR, 70% display
(c) 15% TPR, 70% display, 100% feature
Calculate the lift factors for each promotion for both accounts, based on the regression estimates in 2. Set estimates that are not statistically significant = 0.

```
lift_factor <- function(model, TPR = 0, DIS = 0, FEA = 0) {
  alpha  = if(summary(model)$coef[1,4] < 0.1) {summary(model)$coef[1,1]} else {0}

  beta_lp  = if(summary(model)$coef[2,4] < 0.1) {summary(model)$coef[2,1]} else {0}
#  print(beta_lp)

  beta_d  =
if(length(summary(model)$coef[,1]) >= 3 ){
  if(summary(model)$coef[3,4] < 0.1)
  {summary(model)$coef[3,1]} else {
      0}
} else {
  0
}
#  print(beta_d)

  beta_f  =
if(length(summary(model)$coef[,1]) >= 4 ){
  if(summary(model)$coef[4,4] < 0.1)
  {summary(model)$coef[4,1]} else {
      0}
} else {
  0
}

  lf = exp(beta_lp*log(1 - TPR) + beta_d*DIS + beta_f*FEA)
}


#(a)
#For Dominicks
print(lift_factor(D_d_f_lm, 0.15))
```

```
[1] 1.349254
```

```
#For Jewel
print(lift_factor(J_d_f_lm, 0.15))
```

```
[1] 1.361185
```

```
#(b)
#For Dominicks
print(lift_factor(D_d_f_lm, 0.15, 0.7))
```

```
[1] 2.419158
```

```
#For Jewel
print(lift_factor(J_d_f_lm, 0.15, 0.7))
```

```
[1] 2.877672
```

```r
#(c)
#For Dominicks
print(lift_factor(D_d_f_lm, 0.15, 0.7, 1))
```

```
[1] 3.217896
```

```r
#For Jewel
print(lift_factor(J_d_f_lm, 0.15, 0.7, 1))
```

```
[1] 2.877672
```

2.4 Question

Perform an ROI analysis of the three promotions, (a), (b), and (c), separately for the two retail accounts, Dominick's and Jewel-Osco. The promotions last for one week. Your analysis should follow the approach that we took in class, not the version of this approach taken by Booz Allen Hamilton in the first part of the assignment.

Note. Perform the analysis using units, not cases of Hellman's mayo. You will need the following data for your analysis:

- The regular price of the product at both accounts is \$1.20.
- The VCM for Hellman's is \$0.55 per unit.
- The manufacturer fully pays for the shelf price reduction. E.g., if the shelf price is reduced from \$1.20 to \$1.00, the manufacturer pays for this TPR through a \$0.20 per unit (off-invoice) allowance.
- The fixed cost (MDF) for the promotion involving display only is \$3,000 at Dominick's and \$5,000 at Jewel-Osco. The fixed cost for the promotion including feature and display is \$4,500 at Dominick's and \$6,800 at Jewel-Osco.

In order to estimate baseline sales, use the regression estimates and the regular price, and predict sales for display and feature = 0.

Using these data, and the lift factors found in 3, you can then fill in the cells in the blueprint of a spreadsheet below, for each of the three promotions at both accounts.

Consider both:

- No stockpiling (purchase acceleration)
- The case where 20 percent of the incremental units as predicted by the event lift are due to stockpiling (purchase acceleration), and hence not truly incremental

```
MDF_D_d = 3000
MDF_J_d = 5000
MDF_D_d_f = 4500
MDF_J_d_f = 6800

ROI_Summary <- function(model, TPR = 0, DIS = 0, FEA = 0, fixed_payment_cost = 0, regular_price = 1.2, 

baseline_units = exp(predict(D_d_f_lm, data.frame(price = regular_price, display = 0, feature = 0),
   type = "response"))

total_units = lift_factor(D_d_f_lm, TPR, DIS, FEA) * baseline_units

incremental_units = (total_units - baseline_units)

incremental_units_Stockpiling = incremental_units * Stockpiling

incremental_units_net = incremental_units - incremental_units_Stockpiling

promoted_price = (1 - TPR)*regular_price

promoted_margine = promoted_price - (regular_price - regular_margin)

incremental_contribution = promoted_margine * incremental_units_net

variable_cost = TPR * regular_price * baseline_units

event_cost = variable_cost + fixed_payment_cost

gross_contribution = incremental_contribution - event_cost
```

```r
ROI = gross_contribution/event_cost

list("Baseline units" = baseline_units,
     "Incremental units" = incremental_units,
     "Total units" = total_units,
     "% incremental units purchase acceleration" = Stockpiling,
     "Incremental units purchase acceleration" = incremental_units_Stockpiling,
     "Incremental units net of purchase acceleration" = incremental_units_net,
     "Incremental contribution" = incremental_contribution,
     "Variable cost" = variable_cost,
     "Fixed payment cost" = fixed_payment_cost,
     "Event cost" = event_cost,
     "Event gross contribution" = gross_contribution,
     "ROI" = ROI)
}


#For D
#(a)
ROI_Summary(D_d_f_lm, 0.15)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
      1
3405.77

$`Total units`
       1
13157.33

$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0

$`Incremental units net of purchase acceleration`
      1
3405.77

$`Incremental contribution`
       1
1260.135

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0
```

```
$`Event cost`
      1
1755.28

$`Event gross contribution`
        1
-495.1454

$ROI
         1
-0.2820891
```

```r
#(b)
ROI_Summary(D_d_f_lm, 0.15, 0.7)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
    1
13839

$`Total units`
       1
23590.55

$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0

$`Incremental units net of purchase acceleration`
    1
13839

$`Incremental contribution`
       1
5120.429

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
```

```
       1
3365.148

$ROI
       1
1.917157
```

```r
#(c)
ROI_Summary(D_d_f_lm, 0.15, 0.7, 1)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
       1
21627.93

$`Total units`
       1
31379.49

$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0

$`Incremental units net of purchase acceleration`
       1
21627.93

$`Incremental contribution`
       1
8002.336

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
6247.056

$ROI
       1
3.559008
```

```r
#Consider stockpiling is 20%
#(a)
ROI_Summary(D_d_f_lm, 0.15, Stockpiling = 0.2)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
      1
3405.77

$`Total units`
       1
13157.33

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
681.1539

$`Incremental units net of purchase acceleration`
       1
2724.616

$`Incremental contribution`
       1
1008.108

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
        1
-747.1724

$ROI
         1
-0.4256713
```

```r
#(b)
ROI_Summary(D_d_f_lm, 0.15, 0.7, Stockpiling = 0.2)
```

```
$`Baseline units`
```

```
       1
9751.556

$`Incremental units`
    1
13839

$`Total units`
       1
23590.55

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
2767.799

$`Incremental units net of purchase acceleration`
      1
11071.2

$`Incremental contribution`
       1
4096.343

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
2341.063

$ROI
       1
1.333726
```

```
#(c)
ROI_Summary(D_d_f_lm, 0.15, 0.7, 1, Stockpiling = 0.2)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
       1
21627.93
```

```
$`Total units`
       1
31379.49

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
4325.587

$`Incremental units net of purchase acceleration`
       1
17302.35

$`Incremental contribution`
       1
6401.869

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
4646.588

$ROI
       1
2.647206
```

```r
#For Jewel
#(a)
ROI_Summary(J_d_f_lm, 0.15)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
      1
3405.77

$`Total units`
       1
13157.33
```

```
$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0

$`Incremental units net of purchase acceleration`
      1
3405.77

$`Incremental contribution`
       1
1260.135

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
        1
-495.1454

$ROI
         1
-0.2820891
```

```r
#(b)
ROI_Summary(J_d_f_lm, 0.15, 0.7)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
    1
13839

$`Total units`
       1
23590.55

$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0
```

```
$`Incremental units net of purchase acceleration`
    1
13839

$`Incremental contribution`
       1
5120.429

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
3365.148

$ROI
       1
1.917157
```

```r
#(c)
ROI_Summary(J_d_f_lm, 0.15, 0.7, 1)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
       1
21627.93

$`Total units`
       1
31379.49

$`% incremental units purchase acceleration`
[1] 0

$`Incremental units purchase acceleration`
1
0

$`Incremental units net of purchase acceleration`
       1
21627.93

$`Incremental contribution`
```

```
       1
8002.336

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
6247.056

$ROI
       1
3.559008
```

```r
#Consider stockpiling is 20%
#(a)
ROI_Summary(J_d_f_lm, 0.15, Stockpiling = 0.2)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
      1
3405.77

$`Total units`
       1
13157.33

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
681.1539

$`Incremental units net of purchase acceleration`
       1
2724.616

$`Incremental contribution`
       1
1008.108

$`Variable cost`
      1
```

```
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
        1
-747.1724

$ROI
         1
-0.4256713
```

```r
#(b)
ROI_Summary(J_d_f_lm, 0.15, 0.7, Stockpiling = 0.2)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
    1
13839

$`Total units`
       1
23590.55

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
2767.799

$`Incremental units net of purchase acceleration`
      1
11071.2

$`Incremental contribution`
       1
4096.343

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
```

```
      1
1755.28

$`Event gross contribution`
       1
2341.063

$ROI
       1
1.333726
```

```r
#(c)
ROI_Summary(J_d_f_lm, 0.15, 0.7, 1, Stockpiling = 0.2)
```

```
$`Baseline units`
       1
9751.556

$`Incremental units`
       1
21627.93

$`Total units`
       1
31379.49

$`% incremental units purchase acceleration`
[1] 0.2

$`Incremental units purchase acceleration`
       1
4325.587

$`Incremental units net of purchase acceleration`
       1
17302.35

$`Incremental contribution`
       1
6401.869

$`Variable cost`
      1
1755.28

$`Fixed payment cost`
[1] 0

$`Event cost`
      1
1755.28

$`Event gross contribution`
       1
4646.588
```

```
$ROI
       1
2.647206
```