HW 2

David Gao

10/11/2022

Theory Section

1.

(a) (i)

The lowest bound for access fee should always be 0, as no one want to pay it.

(a) (ii)

The demand curve would be downward (negative slope), assume with constant marginal cost, the consumer surplus (area of triangle above MC cure and below demand curve) wound be the upper bound for the access fee.

(a) (iii)

If both elastic and inelastic demand curve have same intercepts on price for 0 unite product, then consumer gourp with elastic demand curve would be charged with higher access fee as there is more consumer surplus.

(b) (i)

If a two-part tariff is charged, large amount of consumer of this product might be lost and turn to purchase other substitutes.

(b) (ii) (1)

The optimal strategy would be charge access fee for those do not know substitutes, and do not charge access fee for those who know substitutes.

(b) (ii) (1) (a)

If the demand is inelastic, we can charge access fee, as demand quantity and total revenue will not dramatically increase even if we lower the unit price, thus, charge access fee will bring more income directly.

(b) (ii) (1) (b)

We can charge access fee for all consumers, as uninformed consumers do not know substitutes, we can charge maximum access fee equal to consumer surplus, and for informed consumers, same logic as part (a).

Empirical Section

1&2&3

```
library(tidyverse)
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.7
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

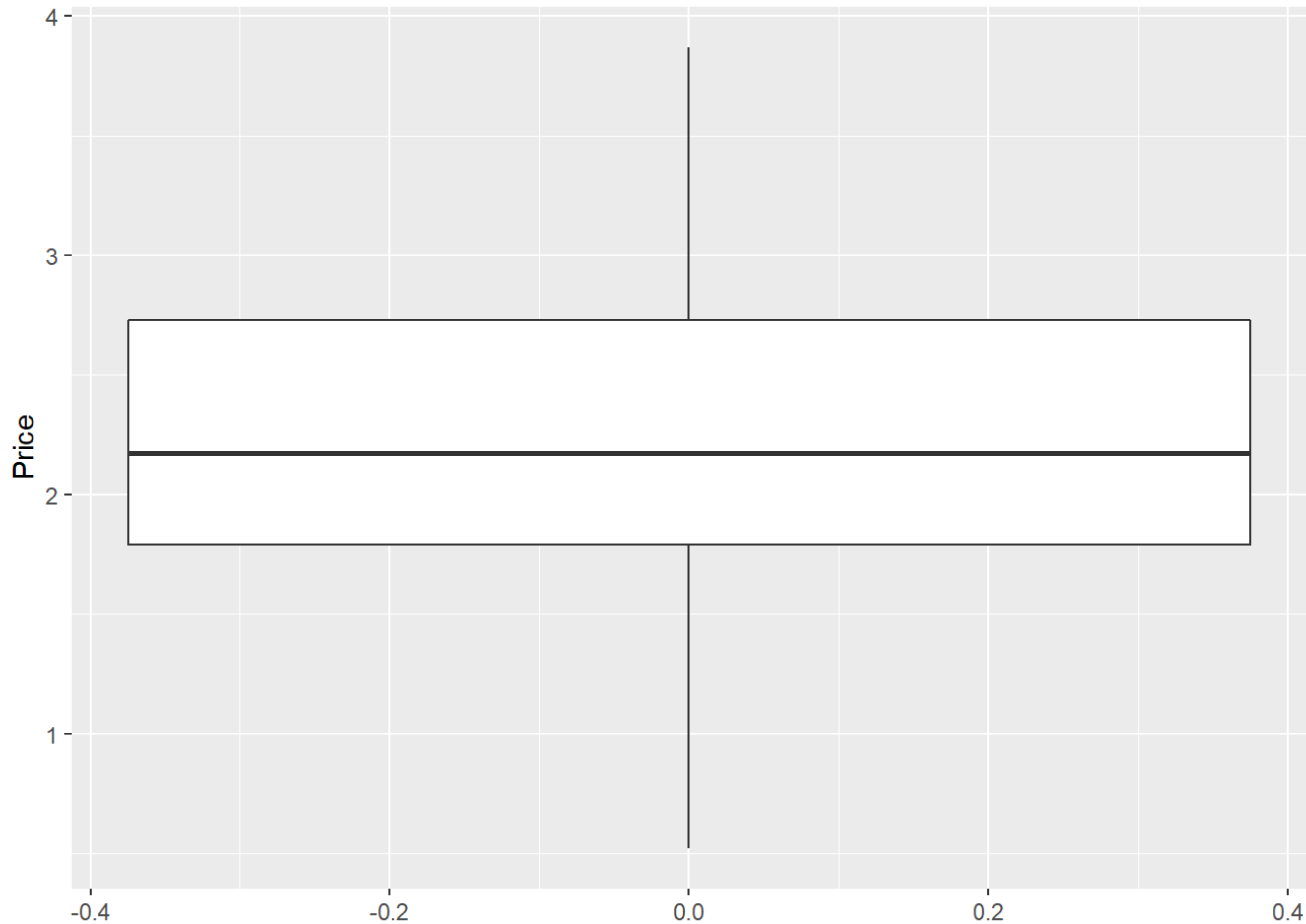
```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)

oj<- read.csv("C:\\UW\\AUT 2022\\ECON 487\\HW 2\\oj.csv")
```

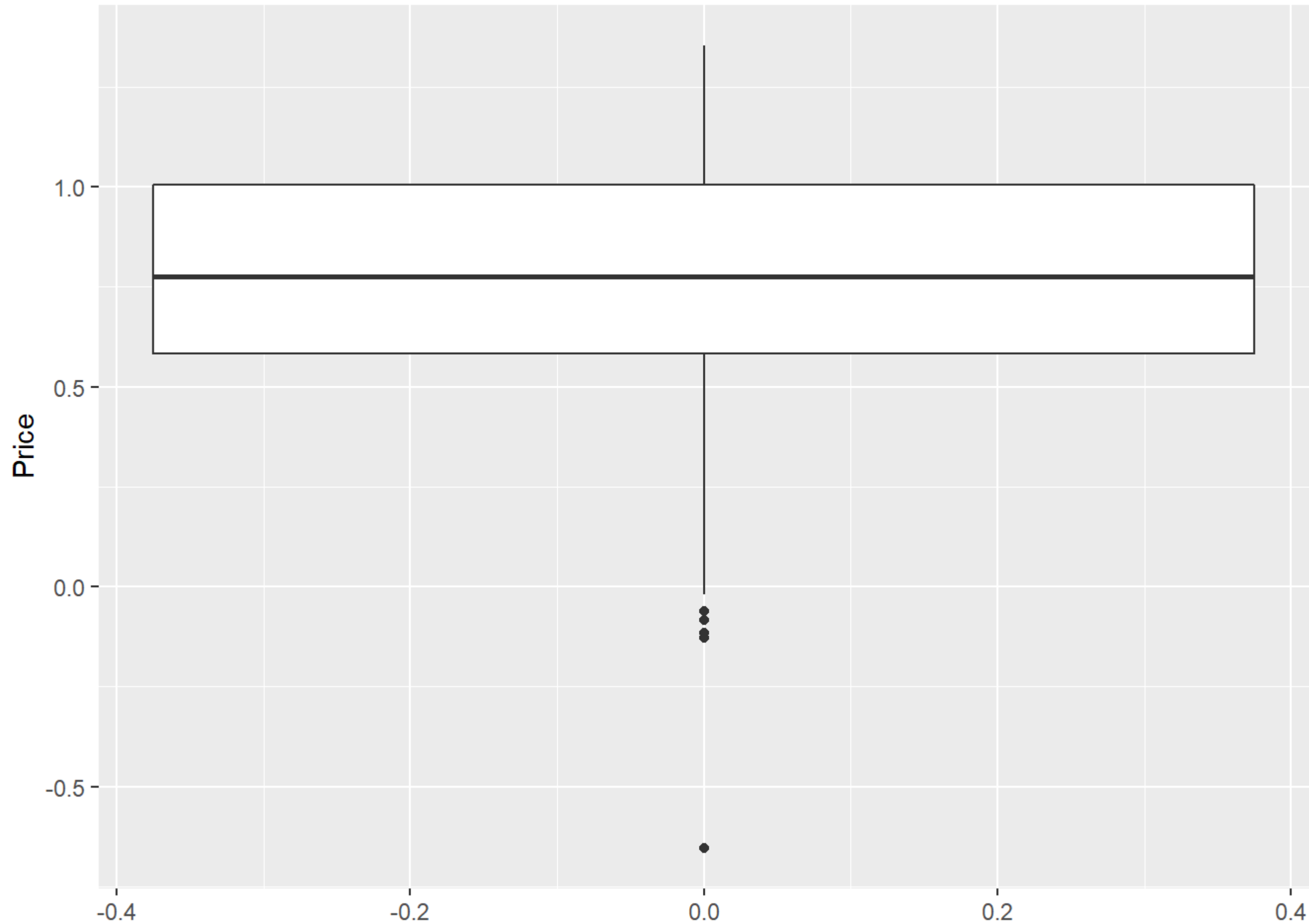
4 (a)

```
ggplot(data = oj)+
  geom_boxplot(mapping = aes(y = price))+
  labs(y = "Price")
```



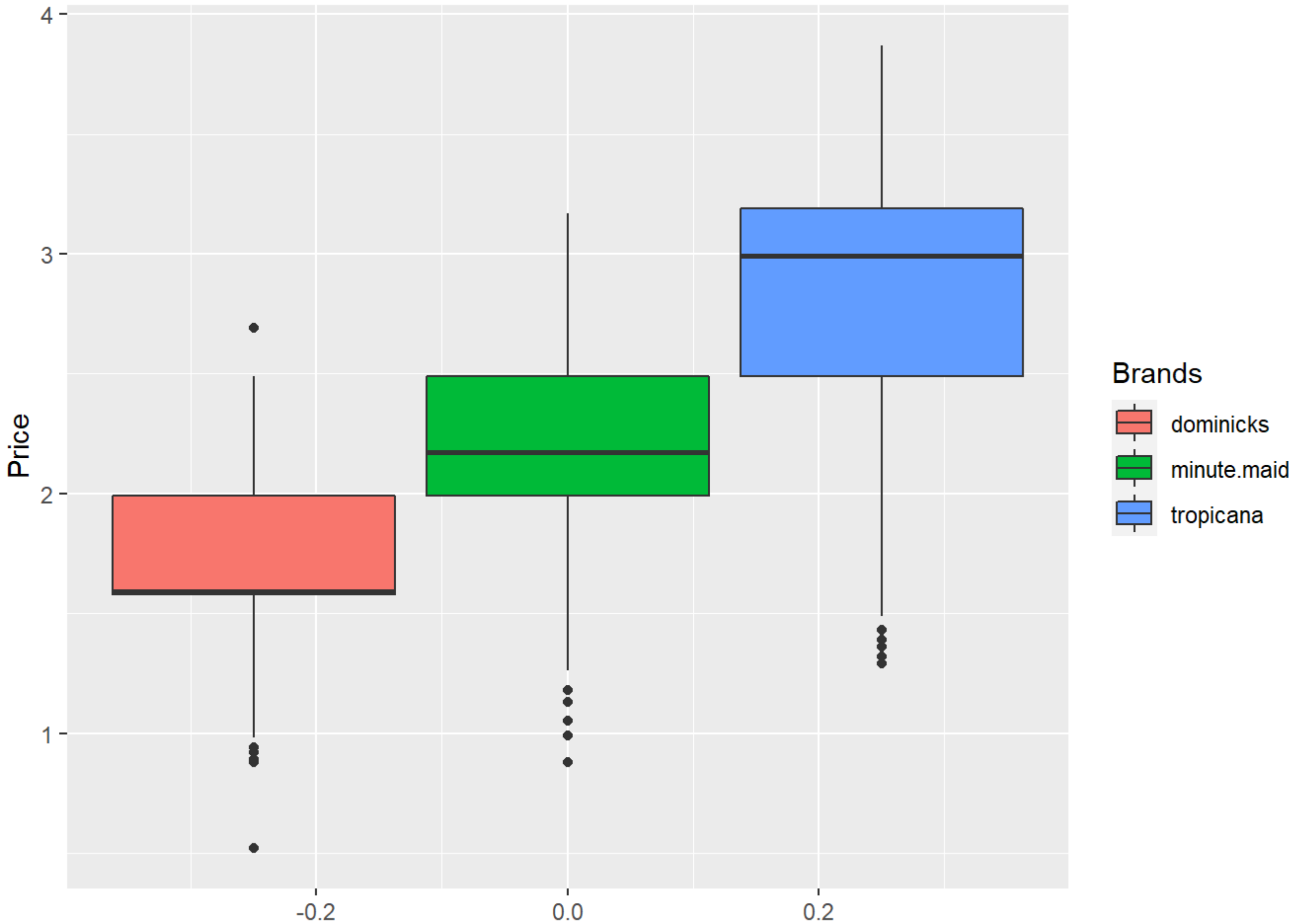
4 (b)

```
ggplot(data = oj)+
  geom_boxplot(mapping = aes(y = log(price)))+
  labs(y = "Price")
```



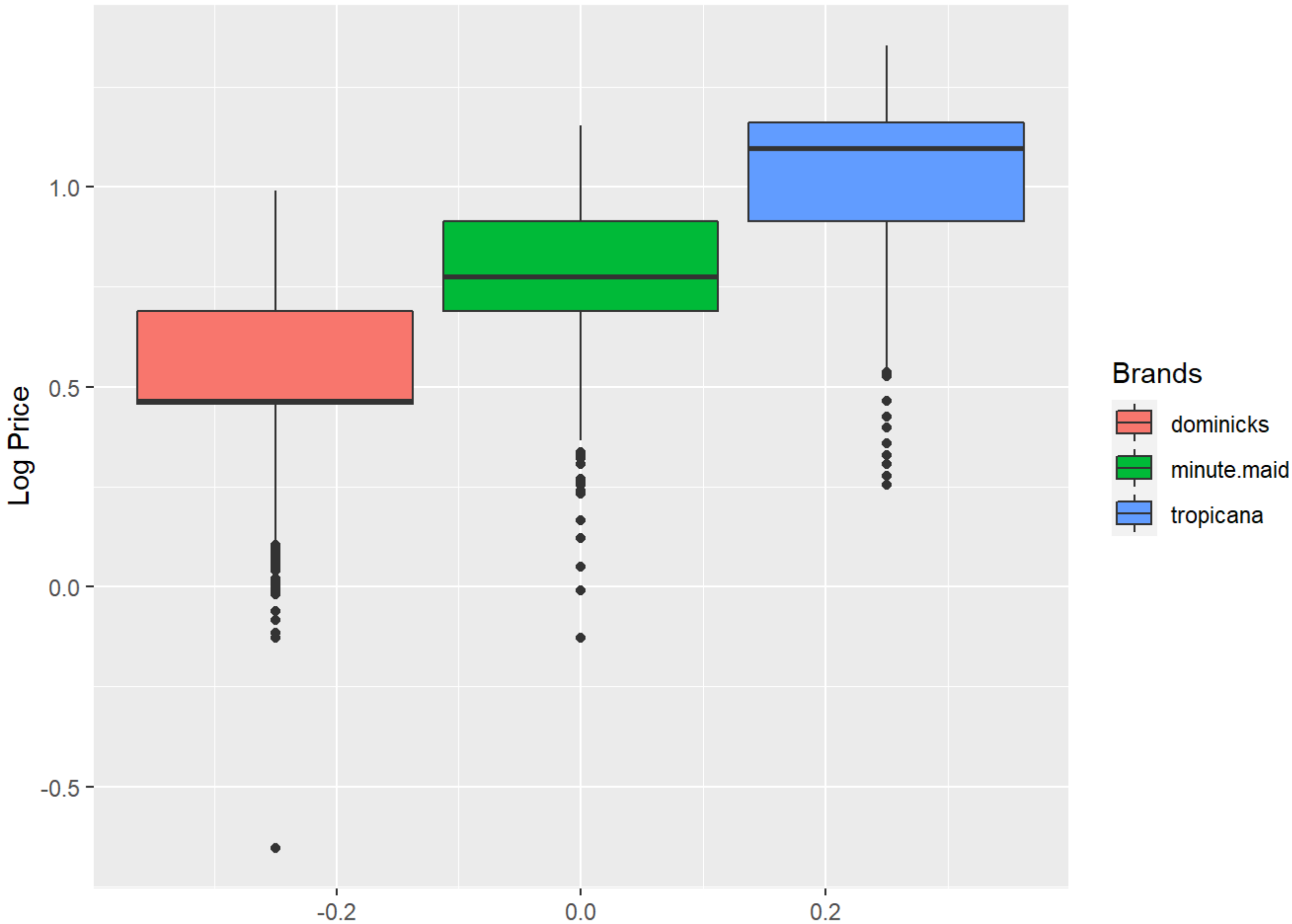
4 (c)

```
ggplot(data = oj)+
  geom_boxplot(mapping = aes(y = price, fill = brand))+
  labs(y = "Price", fill = "Brands")
```



4 (d)

```
ggplot(data = oj)+
  geom_boxplot(mapping = aes(y = log(price), fill = brand))+
  labs(y = "Log Price", fill = "Brands")
```

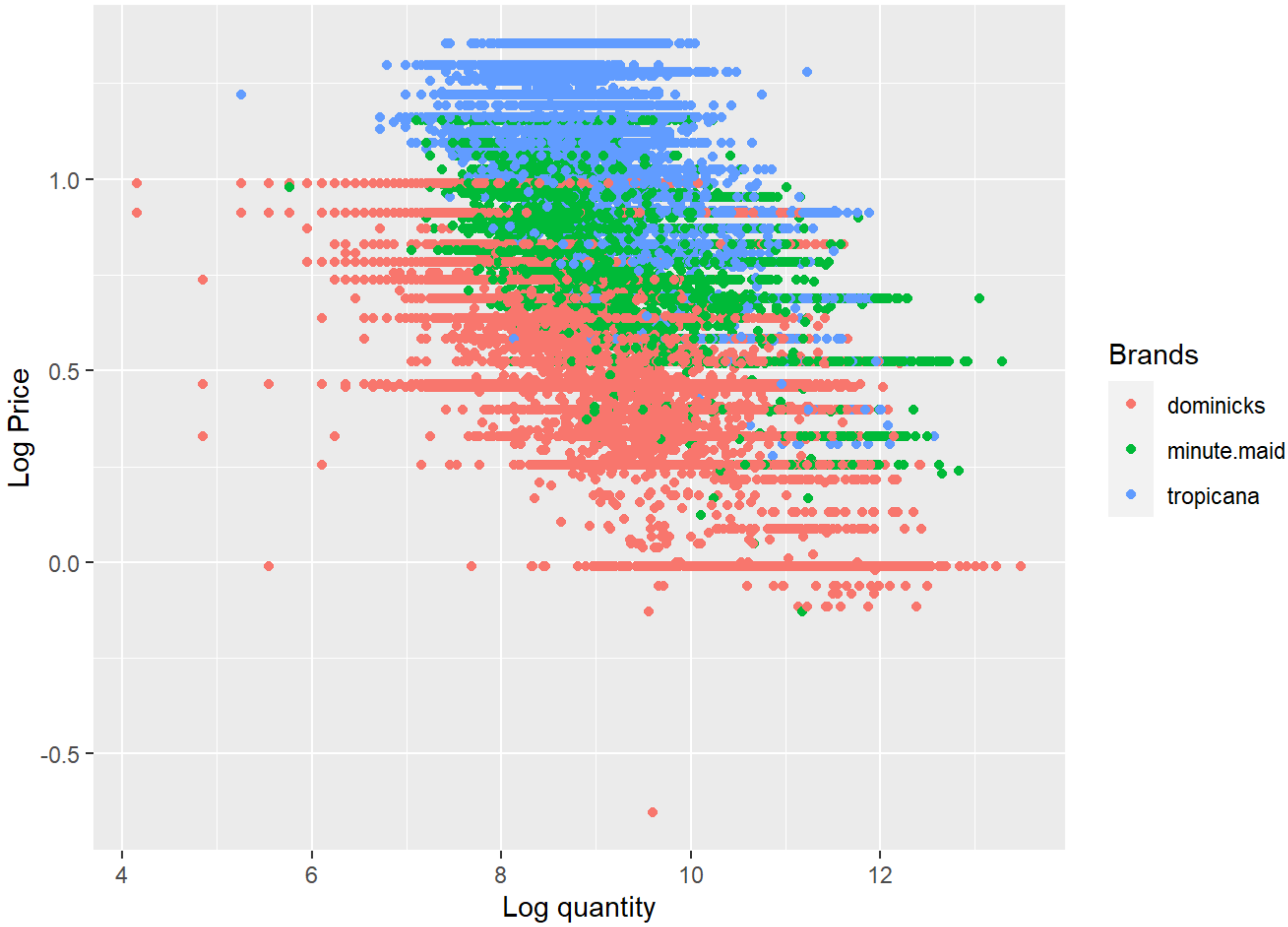


4 (e)

Form the plots, dominicks has lowest average price (also in log price), and tropicana has highest price (also in log price). And log price plot only inform more on spread of some outlier. As this is no exponential relation in plot, log price does not help on simplifying relations.

5

```
ggplot(data = oj)+
  geom_point(mapping = aes(x = logmove, y = log(price), color = brand))+
  labs(x = "Log quantity", y = "Log Price", color = "Brands")
```



This plot shows negative relation between log price and log quantity, which is similar to a demand curve. And from this plor, dominicks seems to still have lowest log unit price where as tropicana has the highest log unit price.

6 (a)

```
model1 <- lm(logmove ~ log(oj$price), data = oj)
summary(model1)
```

```
##
## Call:
## lm(formula = logmove ~ log(oj$price), data = oj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0441 -0.5853 -0.0330  0.5756  3.7264
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.42342    0.01535  679.04   <2e-16 ***
## log(oj$price) -1.60131    0.01836  -87.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9071 on 28945 degrees of freedom
## Multiple R-squared:  0.2081, Adjusted R-squared:  0.2081
## F-statistic:  7608 on 1 and 28945 DF,  p-value: < 2.2e-16
```

Currently, we are having Q(p) = a·p + b, thus, P(q) = $\frac{q}{a}$ - $\frac{b}{a}$. Then, as elasticity ϵ = $\frac{-1}{P'(q)} \cdot \frac{P(q)}{q}$. By replacing P'(q) with $\frac{1}{a}$, we have ϵ = $\frac{-1}{\frac{1}{a}} \cdot \frac{\frac{1}{a} \cdot (-q+b)}{q}$ = $\frac{q-b}{q}$ The elasticity is function ϵ = $\frac{q-10.42342}{q}$. And this elasticity function is well defined on positive integers.

6 (b)

```
model2 <- glm(logmove ~ log(oj$price) + factor(brand), data = oj)
summary(model2)
```

```
##
## Call:
## glm(formula = logmove ~ log(oj$price) + factor(brand), data = oj)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.3152  -0.5246  -0.0502   0.4929   3.5088
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              10.82882    0.01453  745.04   <2e-16 ***
## log(oj$price)            -3.13869    0.02293 -136.89   <2e-16 ***
## factor(brand)minute.maid  0.87017    0.01293   67.32   <2e-16 ***
## factor(brand)tropicana    1.52994    0.01631   93.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6296804)
##
##     Null deviance: 30079  on 28946  degrees of freedom
## Residual deviance: 18225  on 28943  degrees of freedom
## AIC: 68765
##
## Number of Fisher Scoring iterations: 2
```

We have three intercepts for three brands, and tropicana has the highest intercept at 12.3 and dominicks has the lowest intercept at 10.8. And this means tropicana has highest demand curve (highest price at same quantity) and dominicks has lowest demand curve (lowest price at same quantity).

6 (c)

```
dominicks_data <- oj %>%
  filter(brand == "dominicks")

minute.maid_data <- oj %>%
  filter(brand == "minute.maid")

tropicana_data <- oj %>%
  filter(brand == "tropicana")

model_d <- lm(logmove ~ log(price), data = dominicks_data)
summary(model_d)
```

```
##
## Call:
## lm(formula = logmove ~ log(price), data = dominicks_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4434 -0.6071 -0.0336  0.6024  3.4620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.95468    0.02424  451.87   <2e-16 ***
## log(price)  -3.37753    0.04238  -79.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9264 on 9647 degrees of freedom
## Multiple R-squared:  0.397,  Adjusted R-squared:  0.3969
## F-statistic:  6351 on 1 and 9647 DF,  p-value: < 2.2e-16
```

```
model_m <- lm(logmove ~ log(price), data = minute.maid_data)
summary(model_m)
```

```
##
## Call:
## lm(formula = logmove ~ log(price), data = minute.maid_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8259 -0.5581 -0.1049  0.4844  3.4901
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.84294    0.03552  333.45   <2e-16 ***
## log(price)  -3.32073    0.04378  -75.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7799 on 9647 degrees of freedom
## Multiple R-squared:  0.3736, Adjusted R-squared:  0.3735
## F-statistic:  5753 on 1 and 9647 DF,  p-value: < 2.2e-16
```

```
model_t <- lm(logmove ~ log(price), data = tropicana_data)
summary(model_t)
```

```
##
## Call:
## lm(formula = logmove ~ log(price), data = tropicana_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3490 -0.4060 -0.0174  0.4114  2.7828
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.91707    0.03370  353.58   <2e-16 ***
## log(price)  -2.71177    0.03196  -84.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6413 on 9647 degrees of freedom
## Multiple R-squared:  0.4274, Adjusted R-squared:  0.4273
## F-statistic:  7200 on 1 and 9647 DF,  p-value: < 2.2e-16
```

Currently, we are having Q(p) = a·p + b, thus, P(q) = $\frac{q}{a}$ - $\frac{b}{a}$. Then, as elasticity ϵ = $\frac{-1}{P'(q)} \cdot \frac{P(q)}{q}$. By replacing P'(q) with $\frac{1}{a}$, we have ϵ = $\frac{-1}{\frac{1}{a}} \cdot \frac{\frac{1}{a} \cdot (-q+b)}{q}$ = $\frac{q-b}{q}$. The elasticity for dominicks is function ϵ_d = $\frac{q-10.95468}{q}$. The elasticity for minute.maid is function ϵ_m = $\frac{q-11.84294}{q}$. The elasticity for tropicana is function ϵ_m = $\frac{q-11.91707}{q}$. And as all these elastisities are well defined on positive integer q, they make sense.

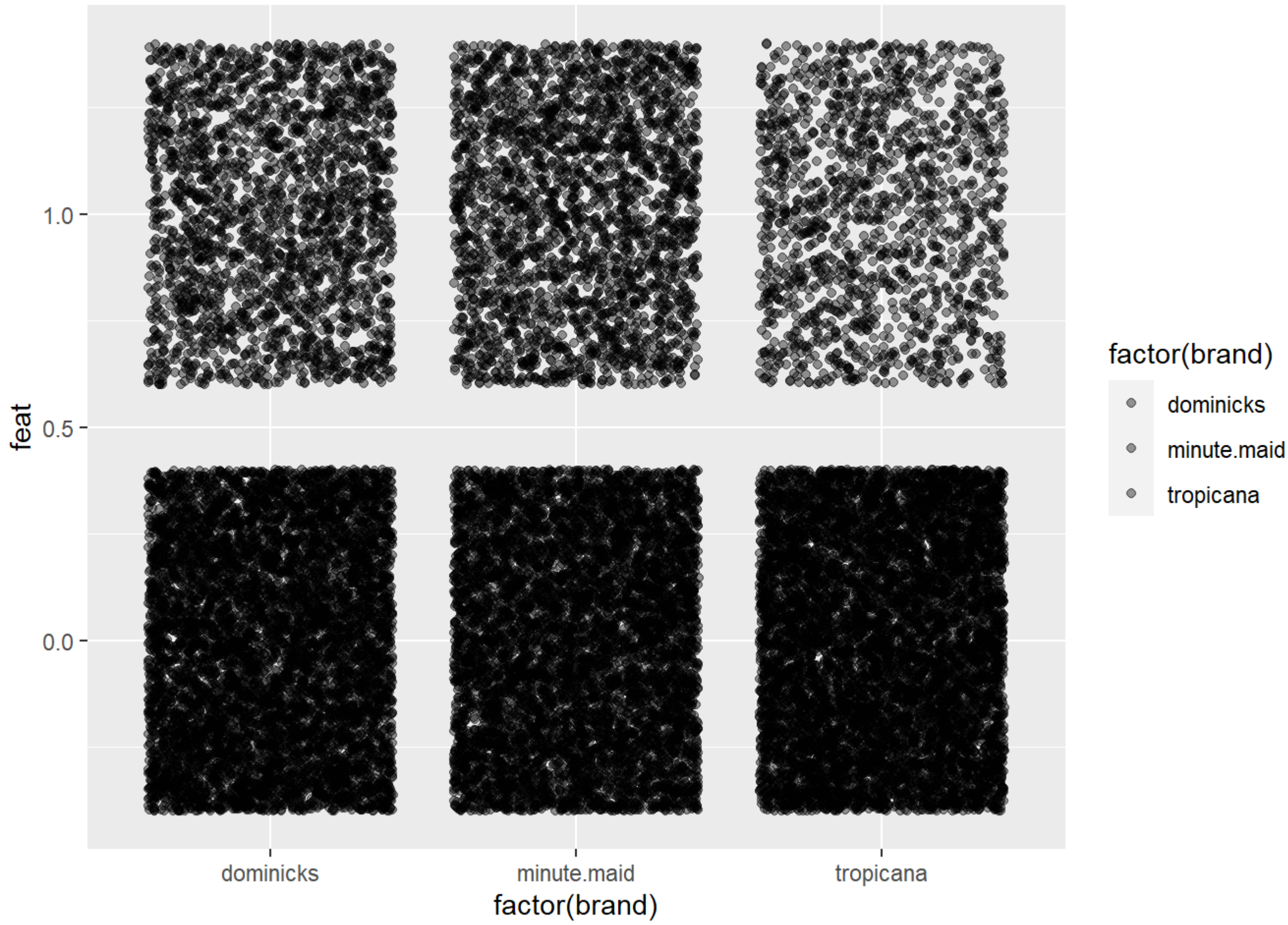
7 (a)

```
aggregate(oj[, 5:6], list(oj$brand), mean)
```

```
##       Group.1      feat    price
## 1   dominicks 0.2570215 1.735809
## 2 minute.maid 0.2885273 2.241162
## 3   tropicana 0.1662348 2.870493
```

The table presents the average price and feature rate of each brand.

```
ggplot(data = oj)+
  geom_point(mapping = aes(x = factor(brand), y = feat, fill = factor(brand)), position = "jitte
r", alpha = 0.4)
```



From the plot, looks like minute.maid featured the most as it has the lowest transparency (the darkest color for feat = 1), and tropicana has the featured the least. And this corresponds with the feature rate in the table above.

7 (b)

```
model1 <- glm(logmove ~ log(oj$price) + factor(feat), data = oj)
summary(model1)
```

```
##
## Call:
## glm(formula = logmove ~ log(oj$price) + factor(feat), data = oj)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5784  -0.4742   0.0274   0.5087   3.0171
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.82087    0.01503  653.62   <2e-16 ***
## log(oj$price) -1.15270    0.01688  -68.27   <2e-16 ***
## factor(feat)1  1.05709    0.01153   91.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6376391)
##
##     Null deviance: 30079  on 28946  degrees of freedom
## Residual deviance: 18456  on 28944  degrees of freedom
## AIC: 69127
##
## Number of Fisher Scoring iterations: 2
```

7 (c)

```
model1 <- glm(logmove ~ log(oj$price)*factor(feat) + factor(feat) + log(oj$price), data = oj)
summary(model1)
```

```
##
## Call:
## glm(formula = logmove ~ log(oj$price) * factor(feat) + factor(feat) +
##     log(oj$price), data = oj)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2995  -0.4706   0.0198   0.4950   3.0765
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   9.65926    0.01642  588.27   <2e-16 ***
## log(oj$price)                -0.95823    0.01869  -51.27   <2e-16 ***
## factor(feat)1                 1.71438    0.03041   56.38   <2e-16 ***
## log(oj$price):factor(feat)1 -0.97729    0.04190  -23.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6258971)
##
##     Null deviance: 30079  on 28946  degrees of freedom
## Residual deviance: 18115  on 28943  degrees of freedom
## AIC: 68590
##
## Number of Fisher Scoring iterations: 2
```

7 (d)

```
model1 <- glm(logmove ~ log(price) + factor(feat), data = dominicks_data)
summary(model1)
```

```
##
## Call:
## glm(formula = logmove ~ log(price) + factor(feat), data = dominicks_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.9633  -0.5380  -0.0042   0.5245   3.2311
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.47934    0.02488  421.12   <2e-16 ***
## log(price)    -2.90339    0.04045  -71.79   <2e-16 ***
## factor(feat)1  0.87732    0.02060   42.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.7224169)
##
##     Null deviance: 13730.1  on 9648  degrees of freedom
## Residual deviance:  6968.4  on 9646  degrees of freedom
## AIC: 24250
##
## Number of Fisher Scoring iterations: 2
```

```
model2 <- glm(logmove ~ log(price) + factor(feat), data = minute.maid_data)
summary(model2)
```

```
##
## Call:
## glm(formula = logmove ~ log(price) + factor(feat), data = minute.maid_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.68130  -0.39091  -0.00847   0.36918   2.79098
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   10.76757    0.03146  342.27   <2e-16 ***
## log(price)    -2.36931    0.03683  -64.33   <2e-16 ***
## factor(feat)1  1.11976    0.01474   75.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3805658)
##
##     Null deviance: 9366.2  on 9648  degrees of freedom
## Residual deviance: 3670.9  on 9646  degrees of freedom
## AIC: 18066
##
## Number of Fisher Scoring iterations: 2
```

```
model3 <- glm(logmove ~ log(price) + factor(feat), data = tropicana_data)
summary(model3)
```

```
##
## Call:
## glm(formula = logmove ~ log(price) + factor(feat), data = tropicana_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3305  -0.3880   0.0010   0.3908   2.7772
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.38555    0.03609  315.51   <2e-16 ***
## log(price)    -2.29149    0.03310  -69.23   <2e-16 ***
## factor(feat)1  0.58169    0.01816   32.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3717326)
##
##     Null deviance: 6927.8  on 9648  degrees of freedom
## Residual deviance: 3585.7  on 9646  degrees of freedom
## AIC: 17839
##
## Number of Fisher Scoring iterations: 2
```

Currently, we are having Q(p) = a·p + c·feat + b, thus, P(q) = $\frac{q}{a}$ - $\frac{c \cdot feat}{a}$ - $\frac{b}{a}$. Then, as elasticity ϵ = $\frac{-1}{P'(q)} \cdot \frac{P(q)}{q}$. By replacing P'(q) with $\frac{1}{a}$, we have ϵ = $\frac{-1}{\frac{1}{a}} \cdot \frac{\frac{1}{a} \cdot (-q + c \cdot feat + b)}{q}$ = $\frac{q-b}{q}$. The elasticity for dominicks is function ϵ_d = $\frac{q - 0.87732 \cdot feat - 10.47934}{q}$. The elasticity for minute.maid is function ϵ_m = $\frac{q - 1.11976 \cdot feat - 10.76757}{q}$. The elasticity for tropicana is function ϵ_m = $\frac{q - 0.58169 \cdot feat - 11.38555}{q}$.

7 (e)

```
model1 <- glm(logmove ~ log(oj$price) + factor(feat) + week, data = oj)
summary(model1)
```

```
##
## Call:
## glm(formula = logmove ~ log(oj$price) + factor(feat) + week,
##     data = oj)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5705  -0.4772   0.0281   0.5123   3.0133
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.7272667  0.0221244 439.662  < 2e-16 ***
## log(oj$price) -1.1351157  0.0171490 -66.191  < 2e-16 ***
## factor(feat)1  1.0580726  0.0115220  91.830  < 2e-16 ***
## week           0.0007922  0.0001375   5.761 8.44e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.6369307)
##
##     Null deviance: 30079  on 28946  degrees of freedom
## Residual deviance: 18435  on 28943  degrees of freedom
## AIC: 69096
##
## Number of Fisher Scoring iterations: 2
```

I add week variable to form formula $Q(p) = a \cdot p + b \cdot feat + c \cdot week + d$, where a = -1.135, b = 1.058, c = 0.00079, d = 9.727.

8 (a)

From 6 (c), we have Q(p) for each brand, and the dominicks has the smallest slope in Q(p), which means largest slope in demand curve P(q). And tropicana has the largest slope in Q(p), which means smallest slope in demand curve P(q). Thus, dominicks has the highest elasticity, and tropicana has the lowest elasticity.

8 (b)

The average prices match the elasticities, as highest elasticity requires lowest price to maintain the quantity and profit as shown in dominicks. And the lowest elasticity can set highest price to maximize the profit as quantity will not be influenced, as shown in tropicana. (using log price here as the model from previous parts are using log price)

8 (c)

```
# average log price
avg_d = mean(log(dominicks_data$price))
avg_d
```

```
## [1] 0.5269682
```

```
avg_m = mean(log(minute.maid_data$price))
avg_m
```

```
## [1] 0.7906856
```

```
avg_t = mean(log(tropicana_data$price))
avg_t
```

```
## [1] 1.034597
```

From previous parts, we know Q(q) = a·p + b (p for log price), then Q_d = -3.37753$$0.5269682 + 10.95468 = 9.174829, Q_m = -3.32073 $$0.7906856 + 11.84294 = 9.217287, Q_t = -2.71177$$1.034597 + 11.91707 = 9.111481.

The elasticity for dominicks is ϵ_d = $\frac{q-10.95468}{q}$ = -0.1939928. The elasticity for minute.maid is ϵ_m = $\frac{q-11.84294}{q}$ = -0.2848618. The elasticity for tropicana is ϵ_t = $\frac{q-11.91707}{q}$ = -0.307918. By function $\frac{p-c}{p} = \frac{-1}{\epsilon}$, log of C_d = -2.18946, log of C_m = -1.985, and log of C_t = -2.32538. Thus, C_d = 0.006465, C_m = 0.010351, C_t = 0.004727.

The unit cost are different, but differences are small. As seen, minute.maid has the highest cost, which can be caused by the high rate of feature which is the use of advertisements. And tropicana has the lowest unit cost as it has the lowest fearture rate, which means less advertisement cost.