Marketing Analytics II: Exercise 2

Davis Townsend

February 27, 2017

First we'll set our working directory and read in the data

```
setwd("C:/Users/Owner/Documents/MSBA/Marketing Analytics II/Datasets")
data = read.csv("2017 Chicago Pizza Data.csv", header = TRUE)
attach(data)
library(xtable)
library(stargazer)
```

1

a)

We'll create 6 new columns for the revenue of each pizza brand

```
data$Revenue1 <- with(data, price1 * sale1)
data$Revenue2 <- with(data, price2 * sale2)
data$Revenue3 <- with(data, price3 * sale3)
data$Revenue4 <- with(data, price4 * sale4)
data$Revenue5 <- with(data, price5 * sale5)
data$Revenue6 <- with(data, price6 * sale6)
```

b)

```
brand = c("DiGiorno", "Tombstone", "Jack's", "Freshchetta", "Red Baron",
    "Tony's")
AvgSalesRev <- c(mean(data$Revenue1), mean(data$Revenue2), mean(data$Revenue3),
    mean(data$Revenue4), mean(data$Revenue5), mean(data$Revenue6))

MedianSalesRev <- c(median(data$Revenue1), median(data$Revenue2),
    median(data$Revenue3), median(data$Revenue4), median(data$Revenue5),
    median(data$Revenue6))

TotalSalesRev <- c(sum(data$Revenue1), sum(data$Revenue2), sum(data$Revenue3),
    sum(data$Revenue4), sum(data$Revenue5), sum(data$Revenue6))

TotalUnitVolume <- c(sum(data$sale1), sum(data$sale2), sum(data$sale3),
    sum(data$sale4), sum(data$sale5), sum(data$sale6))

TotalRev <- sum(TotalSalesRev)
TotalVolume <- sum(TotalUnitVolume)
# calculate market share of sales revenue
MktShareRev <- c(TotalSalesRev[1]/TotalRev, TotalSalesRev[2]/TotalRev,
```

```
    TotalSalesRev[3]/TotalRev, TotalSalesRev[4]/TotalRev, TotalSalesRev[5]/TotalRev,
    TotalSalesRev[6]/TotalRev)
# calculate market share of unit sales volume
MktShareVolume <- c(TotalUnitVolume[1]/TotalVolume, TotalUnitVolume[2]/TotalVolume,
    TotalUnitVolume[3]/TotalVolume, TotalUnitVolume[4]/TotalVolume,
    TotalUnitVolume[5]/TotalVolume, TotalUnitVolume[6]/TotalVolume)

PizzaStats <- data.frame(AvgSalesRev, MedianSalesRev, TotalSalesRev,
    TotalUnitVolume, MktShareRev, MktShareVolume, row.names = brand)

library(xtable)
tab <- xtable(PizzaStats, digits = c(0, 2, 2, 2, 2, 2, 2))
# print(tab, type='latex')
```

code used to generate table above and below is the table generated

Table 1: Market Information of Frozen Pizza Category

	AvgSalesRev	MedianSalesRev	TotalSalesRev	TotalUnitVolume	MktShareRev	MktShareVolume
DiGiorno	451.86	328.83	1973715.41	363708	0.12	0.08
Tombstone	1176.67	794.50	5139681.06	1396924	0.31	0.32
Jack's	1228.83	829.38	5367535.28	1515024	0.32	0.34
Freshchetta	218.77	135.00	955577.36	180505	0.06	0.04
Red Baron	486.56	291.87	2125295.57	505826	0.13	0.11
Tony's	255.69	94.81	1116870.91	458303	0.07	0.10

c) there are difference between revenue based market share and unit volume base market share because revenue based market share takes price of the unit sold each week into account, meaning that one week may have higher revenue and less unit volume or vive versa depending on the price. Then, aggregated these effects get magnified and you get slightly different market shares of revenue v volume

d) For each brand, the average revenue is greater than the median revenue. This suggest a right tailed distribution caused by some extreme outliers far on the right side of the sales distribution that drag the average to the right, causing it to be greater than the median, which is the measure of the midpoint of the distribution.

2

a)

```
# get unweighted price of each brand
AvgUnwtPrice <- c(mean(data$price1), mean(data$price2), mean(data$price3),
    mean(data$price4), mean(data$price5), mean(data$price6))
# get weighted price for each brand
Wtprice1 <- weighted.mean(data$price1, data$sale1)
Wtprice2 <- weighted.mean(data$price2, data$sale2)
Wtprice3 <- weighted.mean(data$price3, data$sale3)
Wtprice4 <- weighted.mean(data$price4, data$sale4)
Wtprice5 <- weighted.mean(data$price5, data$sale5)
Wtprice6 <- weighted.mean(data$price6, data$sale6)
# combine average weighted price
AvgwtPrice <- c(Wtprice1, Wtprice2, Wtprice3, Wtprice4, Wtprice5,
    Wtprice6)
# get average unweighted unit sales
AvgUnwtSales <- c(mean(data$sale1), mean(data$sale2), mean(data$sale3),
    mean(data$sale4), mean(data$sale5), mean(data$sale6))
# get standard deviation of unit sales
StDevUnitSales <- c(sd(data$sale1), sd(data$sale2), sd(data$sale3),
    sd(data$sale4), sd(data$sale5), sd(data$sale6))

# make data frame for table
PizzaStats2 <- data.frame(AvgUnwtPrice, AvgwtPrice, AvgUnwtSales,
    StDevUnitSales, row.names = brand)
# create latex syntax for the table, digits are how many
# digits for rounding
tab2 <- xtable(PizzaStats2, digits = c(0, 2, 2, 2, 2))
# print(tab2, type='latex')
```

code used to generate table above and below is the table generated

Table 2: summarizing competitive scene in Frozen Pizza space

	AvgUnwtPrice	AvgwtPrice	AvgUnwtSales	StDevUnitSales
DiGiorno	5.72	5.43	83.27	89.12
Tombstone	4.07	3.68	319.81	353.31
Jack's	3.77	3.54	346.85	379.68
Freshchetta	5.67	5.29	41.32	52.23
Red Baron	4.56	4.20	115.80	155.50
Tony's	3.01	2.44	104.92	230.68

b)

Based on the prices I would say that Freshchetta and DiGiorno are competing for the high quality frozen pizza market. Note that they both have fancy italian names. Tombstone and Jack's seem to compete for the mid-tier quality frozen pizza catgeory. Red Baron prices itself between these 2 levels at a medium-high quality price range. Finally, Tony round us off pricing itself into a low-tier quality frozen pizza category and seems to have little to no competition in this space (from our data at least)

c)

From the sales statistics it looks like DiGiorno is winning out in the high-priced frozen pizza category over Freshchetta with double their average unti sales. Also it's important to note that overall the high priced frozen pizza category has the lowest average sales in general. The mid tier frozen pizzas of Tombstone and Jack's look relatively even in their average sales, and this tier sells the most on average. Red Baron and Tony's have similar average sales around 100, each taking a decent chunk of customers due to their unique pricing tiers in this market. However DiGiorno sells almost as much as these 2 brands on average, and they likely have higher margins, so if I was a pizza manufacturer I think I'd like to be either in the mid tier selling a lot for cheaper prices or be DiGiorno and sell few high priced frozen pizzas. Another interesting thing to note is that in general the lower and mid tier priced frozen pizza categories seem to have much more variations in their sales, suggesting that maybe there is more brand loyalty/recognition for the high priced brands with continuous purchases and less so for the lower priced frozen pizza brands.

d)

The weighted average price takes into account the number of sales, as well as the price, instead of solely the price. We see all the weighted averages are lower than the unweighted averages. This makes sense with regard to prices and sales, because according to the Law of Demand, for a normal good: if prices fall, sales are expected to rise. So for instance, if there was a sale at the store, whereby they lowered the price of a frozen pizza brand on a certain week, then the sales that week probably increased.Now, due to the

3

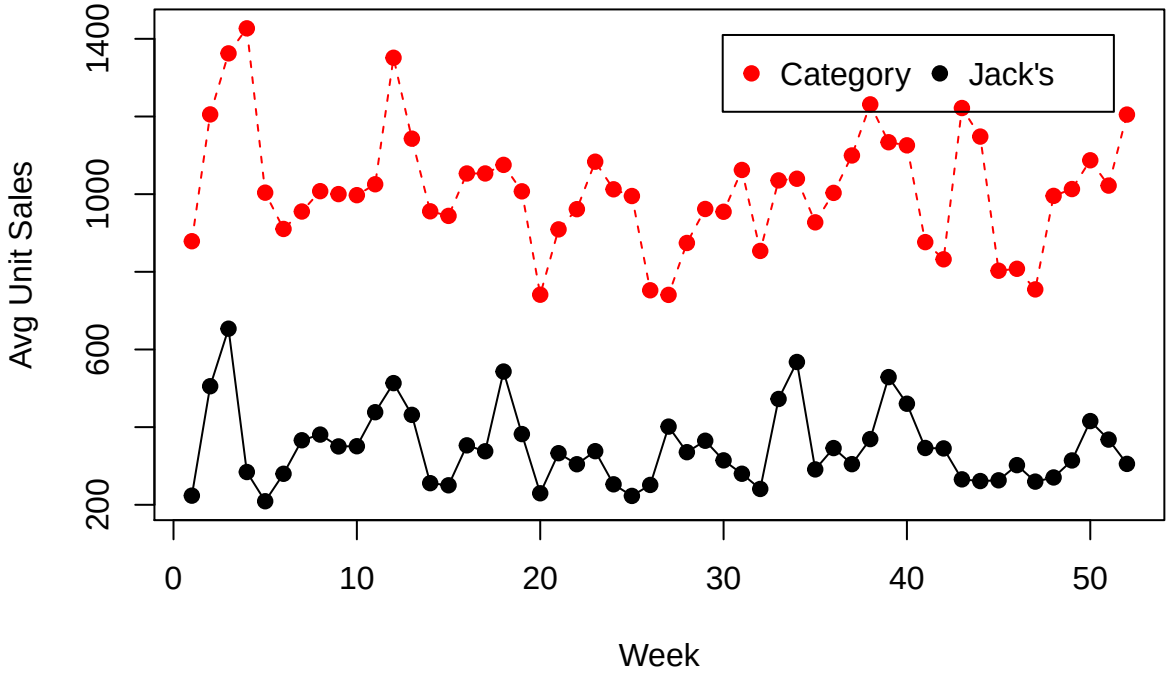
a)

```
# get AvgJackSales by week
z <- aggregate(data$sale3, by = list(data$week), mean)
AvgJacksSales <- z[, 2]
# in case you were dumb like me and want to create your own
# category volume variable because you didnt realize there
# was already one in the dataset
data$TotCatUnitSales <- with(data, rowSums(data[, c("sale1",
    "sale2", "sale3", "sale4", "sale5", "sale6")]))
# aggregate this by week also
zed <- aggregate(data$TotCatUnitSales, by = list(data$week),
    mean)
AvgCatSales <- zed[, 2]

# plot the data
matplot(data$week[1:52], cbind(AvgJacksSales, AvgCatSales), xlab = "Week",
    ylab = "Avg Unit Sales", pch = 19)

matlines(data$week[1:52], cbind(AvgJacksSales, AvgCatSales),
    type = "l", lty = 1:5, lwd = 1, pch = NULL, col = 1:6)

legend("topright", inset = 0.05, legend = c("Category", "Jack's"),
    pch = 19, col = c(2, "black"), horiz = TRUE)
```



b)

There does seem to be some seasonality, with avg unit sales spiking around the holidays (we see spikes starting near December, as well as a little after the beginning of the year which may be explained by National pizza Week being from Jan 8th - Jan 14th)

c)

```
# use which.max to get index of max value of this vector
max <- which.max(z[, 2])
max
```

```
## [1] 3
```

```
# we see the max is at index 3 so now we can go back and find
# this in our variable z from earlier
z[3, ]
```

```
##   Group.1        x
## 3       3 653.5238
```

we see that Week 3 has the max Avg Unit Sales for Jack's at 653.5238

d)

```
# get averages for all jack's relevant variables for all
# stores by quarter. Only want the vector of the numbers for
# our purposes
AvgPricePerQuarter <- aggregate(data$price3, by = list(data$quarter),
    mean)[, 2]
AvgSalesPerQuarter <- aggregate(data$sale3, by = list(data$quarter),
    mean)[, 2]
AvgRevPerQuarter <- aggregate(data$Revenue3, by = list(data$quarter),
    mean)[, 2]
AvgFeatPerQuarter <- aggregate(data$feature3, by = list(data$quarter),
    mean)[, 2]
AvgDispPerQuarter <- aggregate(data$display3, by = list(data$quarter),
    mean)[, 2]

Quarter <- c("1", "2", "3", "4")

JacksAvgStats <- data.frame(AvgPricePerQuarter, AvgSalesPerQuarter,
    AvgRevPerQuarter, AvgFeatPerQuarter, AvgDispPerQuarter, row.names = Quarter)

tab <- xtable(JacksAvgStats, digits = c(0, 2, 2, 2, 2, 2))
# print(tab, type='latex')
```

Table 3: Average Statistics for Jack's frozen pizza

	AvgPricePerQuarter	AvgSalesPerQuarter	AvgRevPerQuarter	AvgFeatPerQuarter	AvgDispPerQuarter
1	3.70	383.66	1336.72	0.42	0.37
2	3.80	311.87	1101.79	0.35	0.34
3	3.77	370.55	1317.48	0.40	0.37
4	3.80	321.31	1159.33	0.35	0.28

e)

As we can see from the table above, Q1 had the lowest average price at $3.70

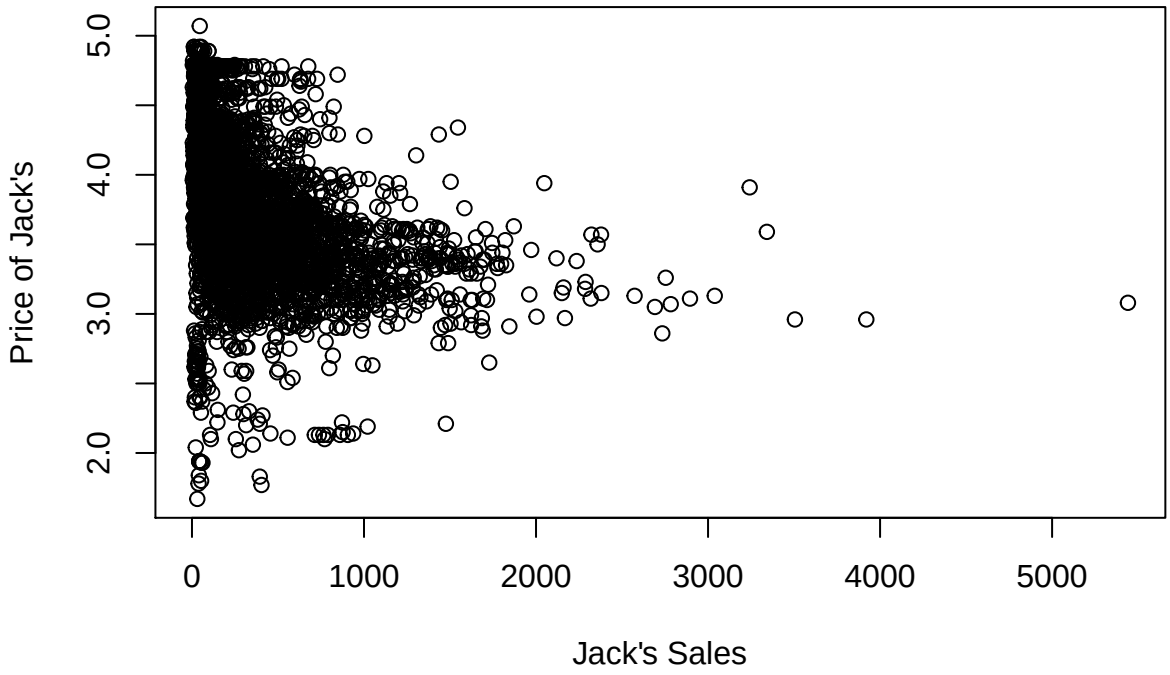
f)

As we saw from our weighted prices earlier, all the weighted average price were lower than their unweighted counterparts. This would hold true on the Quarter level also, with the average weighted price being lower than the unweighted average prices we just found.

4)

a)

```
plot(data$sale3, data$price3, xlab = "Jack's Sales", ylab = "Price of Jack's")
```



Note we see an inverse relationship between price and sales, consisten with what the Law of Demand from economics tell us.

b)

```
# calculate correlation between Jack's sales and price
corrcoeff <- cor(data$sale3, data$price3)
corrcoeff
```

```
## [1] -0.4005702
```

We see the correlation coefficient is negative at -.4005702

c)

Sales is the dependent variable because sales DEPEND on the price

d)

Price is the independent variable because price is independent of what sales is

e)

```
linmod <- lm(data$sale3 ~ data$price3, data = data)
summary(linmod)
```

```
##
## Call:
## lm(formula = data$sale3 ~ data$price3, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -939.3 -189.8  -68.5   92.4 4889.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1466.78      39.13   37.49   <2e-16 ***
## data$price3  -297.29      10.29  -28.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 347.9 on 4366 degrees of freedom
## Multiple R-squared:  0.1605, Adjusted R-squared:  0.1603
## F-statistic: 834.4 on 1 and 4366 DF,  p-value: < 2.2e-16
```

We see that R squared is around .16, so price of Jack's explains about 16% of the variance in the sales of Jack's frozen pizza.

f)

```
# stargazer(linmod, title= 'Linear Model of Jack's Price and
# Sales')
```

Table 1: Linear Model of Jack's Price and Sales

	Dependent variable:
	sale3
price3	-297.288^{***}
	(10.292)
Constant	$1{,}466.776^{***}$
	(39.125)
Observations	4,368
R^2	0.160
Adjusted R^2	0.160
Residual Std. Error	347.931 (df = 4366)
F Statistic	834.445^{***} (df = 1; 4366)
Note:	*p<0.1; **p<0.05; ***p<0.01

as we can see from the above table the coefficients listed next to their respective variables and the standard errors below these coefficients in parentheses.

g)

The parameter can be interpreted as follows: For a 1 unit increase in price of Jack's frozen pizza, we expect on average to see a 297.29 decrease in sales.

---

5)

a)

```
linmod2 <- lm(data$sale3 ~ data$price3 + data$feature3 + data$display3,
    data = data)
summary(linmod2)
```

```
##
## Call:
## lm(formula = data$sale3 ~ data$price3 + data$feature3 + data$display3,
```

```
##     data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -726.5 -173.0  -51.8   80.4 4746.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     978.38      43.08  22.710   <2e-16 ***
## data$price3    -198.11      10.72 -18.482   <2e-16 ***
## data$feature3   104.82      11.41   9.189   <2e-16 ***
## data$display3   221.05      11.23  19.687   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 326.7 on 4364 degrees of freedom
## Multiple R-squared:  0.2603, Adjusted R-squared:  0.2597
## F-statistic: 511.8 on 3 and 4364 DF,  p-value: < 2.2e-16
```

```
# stargazer(linmod2, title= 'Linear Model of Jack's Sales
# regressed on Price, Feature, and Display')
```

b)

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

Table 2: Linear Model of Jack's Sales regressed on Price, Feature, and Display

	Dependent variable:
	sale3
price3	-198.113^{***}
	(10.719)
feature3	104.822^{***}
	(11.408)
display3	221.050^{***}
	(11.228)
Constant	978.382^{***}
	(43.081)
Observations	4,368
R^2	0.260
Adjusted R^2	0.260
Residual Std. Error	326.673 (df = 4364)
F Statistic	511.767^{***} (df = 3; 4364)
Note:	*p<0.1; **p<0.05; ***p<0.01

Note: the standard errors are shown in the table above in parentheses below the coefficient estimates

c)

As we can see by the new R squared value in the table above, about 26% of the variance in sales is explained by this model containing price, feature, and display.

d)

So we can interpret the coefficient of Jack's feature in our model as follows: If Jack's frozen pizza is featured (rather than not featured), then we expect on average that Jack's frozen pizza sales will increase by 104.82, holding display and price of Jack's frozen pizza fixed.

e)

If Jack's increased it's selling price by $1, we'd expect sales to decrease on average by 198.11, holding feature and display fixed.

f)

The effect of price on sales became smaller. This is due to the fact, as we see in our new model, that feature and display are also statistically significant in predicting changes in sales. Therefore, in our original model our estimate for price was biased by these unobserved variables that were not previously in our model. Once we put these variables in our model, we control for the effects of these 2 variables and get a "truer" picture about how much effect price has on sales. In fact, if we had even more data on advertising expenditures and related variables in our moel, the effect of price on sales would likely become yet even smaller.

we can actually check our intuition of this estimator being biased by these ommitted variables via the following criterion:

1. the omitted variable must be a determinant of the dependent variable (i.e., its true regression coefficient is not zero) -this was shown to be satisfied by the statistical significance of the 2 new variables, rejecting Ho that the coefficients are = 0

2. the omitted variable must be correlated with one or more of the included independent variables (i.e. cov(z,x) is not equal to zero). -this we check below and see that in fact both variables were correlated with price

```
cor(data$price3, data$feature3)
```

```
## [1] -0.3964563
```

```
cor(data$price3, data$display3)
```

```
## [1] -0.2920399
```

Following a rule of thumb, since the correlations are negative and the coefficient on price is negative, we know that the coefficient on price (the biased estimator) should increase in the positive direction(become less negative) in the new model, and we do in fact see this.

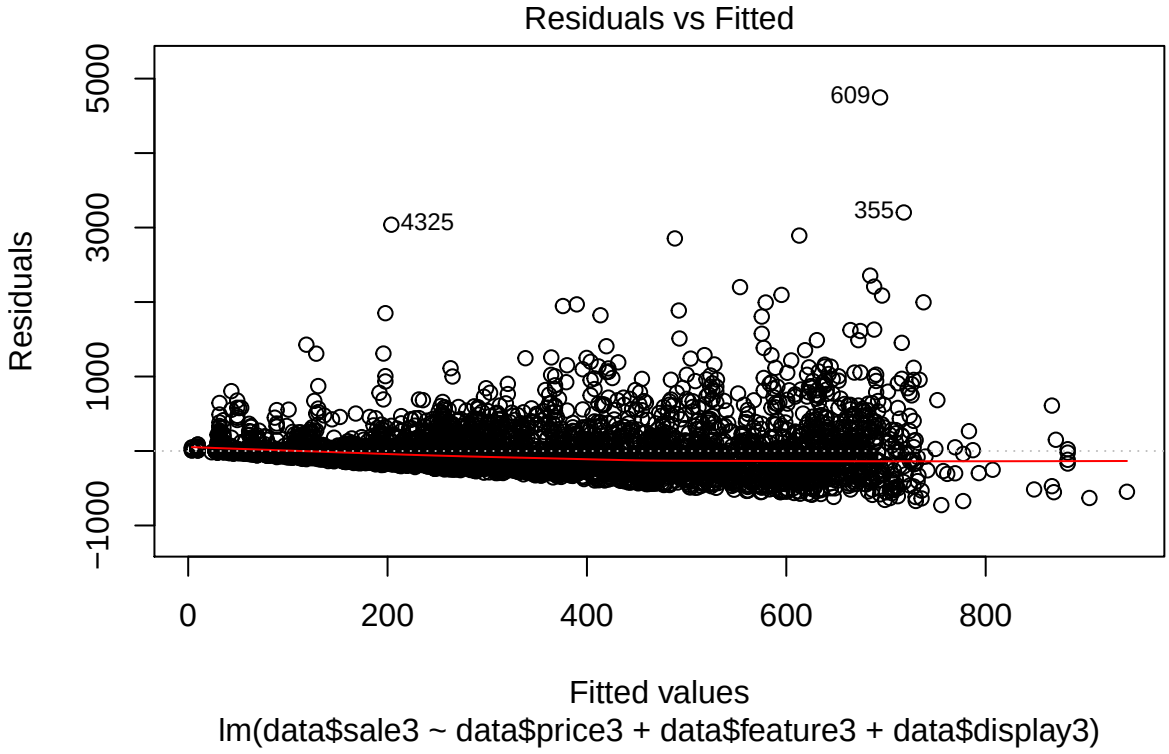
---

6)

a)

We see from the plot below that the assumption of homoskedasticity (i.e. constant variance over all x's) is violated in our model. The variance is not constant over the fitted y values. Generally, you are looking for a trumpet shape in the residuals and a curved red line to determine if there is heteroskedasticity. Clearly, this model fails the test.

```
plot(linmod2, which = 1)
```



We can also test homoskedasticity with the Breusch Pagan Test or the NCV (Non-Constant-Variance) test. We'll use the Breusch Pagan Test here

```
library(lmtest)
# Breushe Pagan Test
lmtest::bptest(linmod2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linmod2
## BP = 120.27, df = 3, p-value < 2.2e-16
```

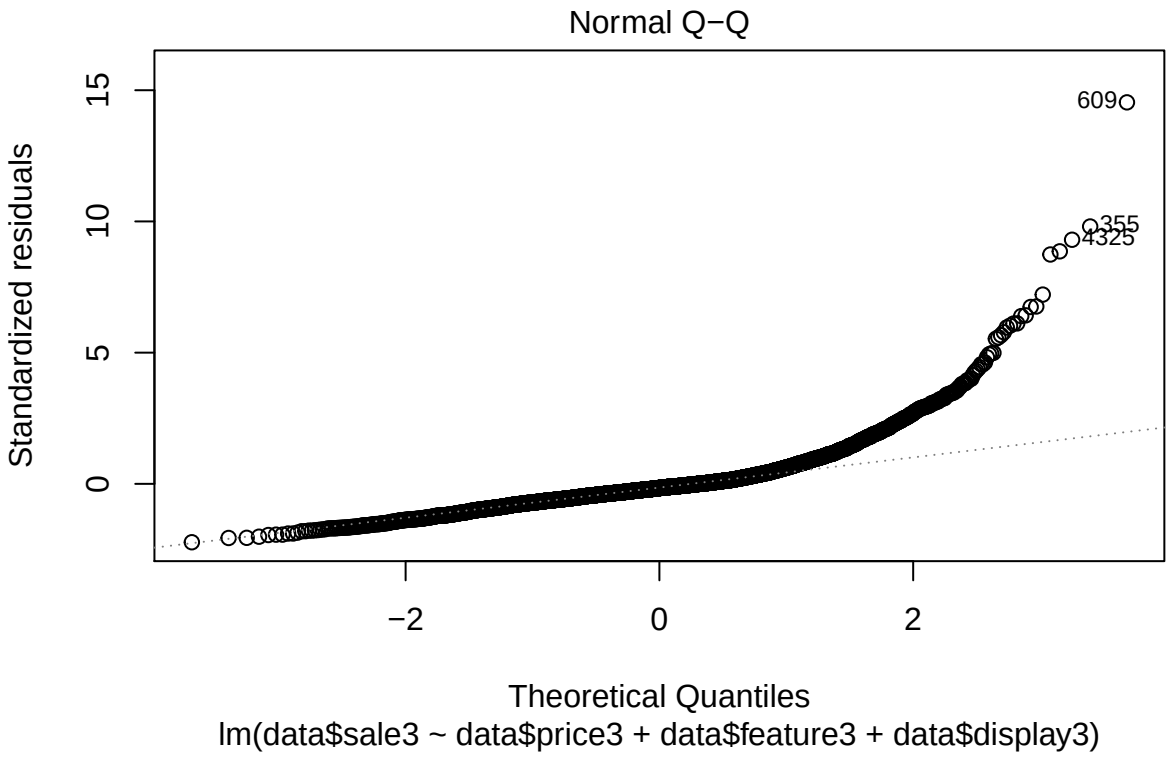
We see that the p-value is $< .05$ threshold, so we can again reject Ho that there is constant variance

Homoskedasticity: Fail

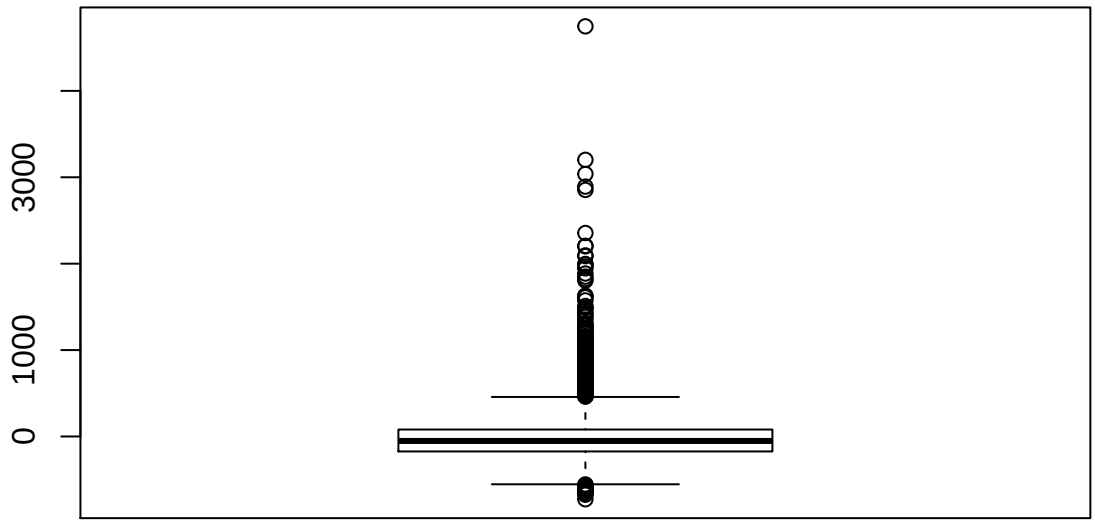
b)

We can check the assumption of normality of residuals with a Q-Q plot or a boxplot. We are expecting the residuals to lie on the dotted line in the Q-Q plot, and in the boxplot, we expect the residuals to be normally distributed around 0. We see below that in both cases, our residuals are clearly not normally distributed, so our model also fails the test for normality of residuals. The test for Normality is essentially checking how our model is making errors, what we really don't want in our model is to have systematic errors (keep overestimating values in certain ranges of the model for example)

```
plot(linmod2, which = 2)
```




```
boxplot(residuals(linmod2))
```




```
shapiro.test(residuals(linmod2))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(linmod2)
## W = 0.79234, p-value < 2.2e-16
```

Above we see we can also test normality of residuals via the Shapiro-Wilk statistical test. The p-value of this test falls below .05 threshold, so we can reject Ho that the residuals are normally distributed

Normality: Fail

c)

No, as stated above our model fails both of these assumptions

d)

We can use transformations in order "to linearize non-linear relationships between independent and dependent variables to produce residuals that are normally distributed with constant variance" and fix our violations of these 2 assumptions. For example, in our data it looks like a log transform of the data may be suitable.

---

7)

a)

```
data$lnprice1 = log(data$price1)
data$lnprice2 = log(data$price2)
data$lnprice3 = log(data$price3)
data$lnprice4 = log(data$price4)
data$lnprice5 = log(data$price5)
data$lnprice6 = log(data$price6)
```

b)

```
data$lnsale3 = log(data$sale3)
```

c)

```
linmod3 = lm(lnsale3 ~ feature3 + display3 + lnprice1 + lnprice2 +
    lnprice3 + lnprice4 + lnprice5 + lnprice6, data = data)
summary(linmod3)
```

```
##
## Call:
## lm(formula = lnsale3 ~ feature3 + display3 + lnprice1 + lnprice2 +
##     lnprice3 + lnprice4 + lnprice5 + lnprice6, data = data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3958 -0.5288  0.0566  0.5894  3.1075
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.60126    0.22730  29.043  < 2e-16 ***
## feature3     0.35573    0.03181  11.181  < 2e-16 ***
## display3     0.66829    0.03090  21.628  < 2e-16 ***
## lnprice1     0.49562    0.12788   3.876 0.000108 ***
## lnprice2     0.68103    0.07476   9.109  < 2e-16 ***
## lnprice3    -2.51905    0.13012 -19.359  < 2e-16 ***
## lnprice4    -0.43316    0.11843  -3.658 0.000258 ***
## lnprice5     0.31066    0.08751   3.550 0.000389 ***
## lnprice6     0.14296    0.06257   2.285 0.022370 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8859 on 4359 degrees of freedom
## Multiple R-squared:  0.3494, Adjusted R-squared:  0.3483
## F-statistic: 292.7 on 8 and 4359 DF,  p-value: < 2.2e-16
```

d)

```
# stargazer(linmod3, title= 'Linear Model of Jack's LN(Sales)
# regressed on LN(Price) of every brand, Jack's Feature, and
# Jack's Display')
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

Note: standard errors shown in parentheses below coefficients

e)

Ommitted Variable Bias

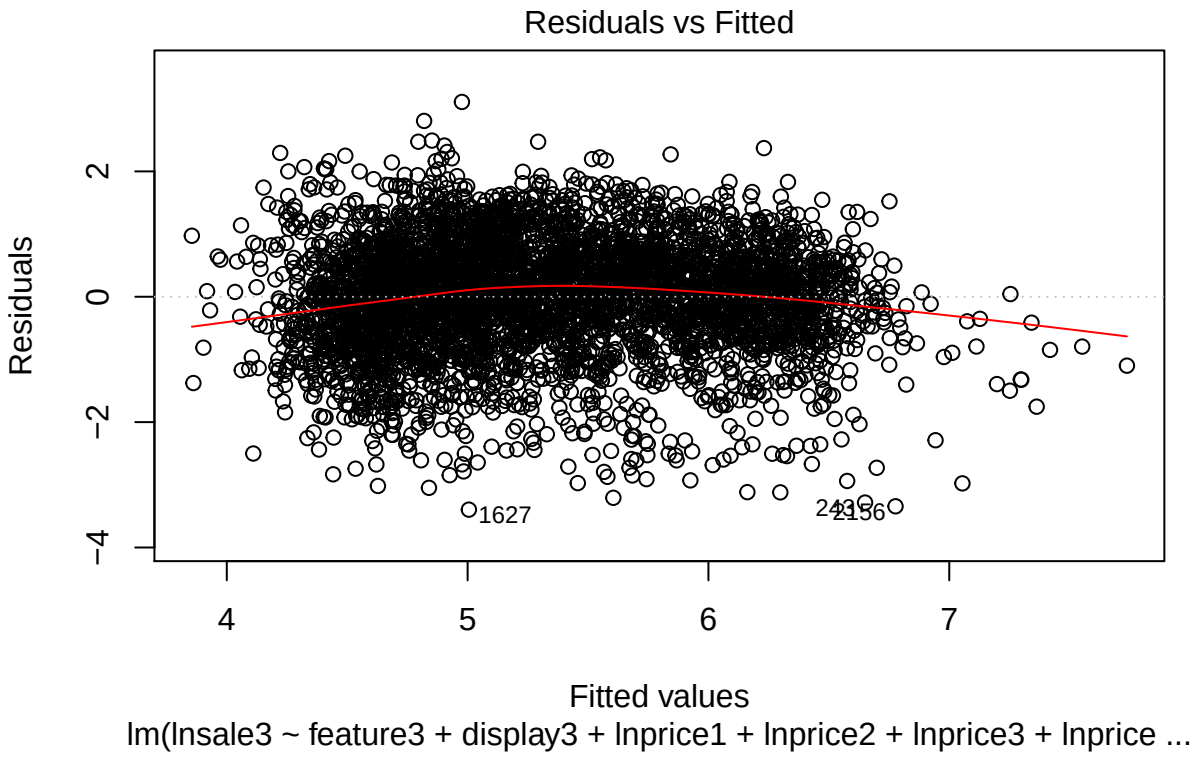
f)

we'll check for heteroskedasticity via a plot of the residuals as well as the Breusch Pagan test for homoskedasticity

```
# we'll check for heteroskedasticity via a plot of the
# residuals as well as the Breusch Pagan test for
# homoskedasticity
plot(linmod3, which = 1)
```

Table 3: Linear Model of Jack's LN(Sales) regressed on LN(Price) of every brand, Jack's Feature, and Jack's Display

	Dependent variable:
	lnsale3
feature3	0.356^{***}
	(0.032)
display3	0.668^{***}
	(0.031)
lnprice1	0.496^{***}
	(0.128)
lnprice2	0.681^{***}
	(0.075)
lnprice3	-2.519^{***}
	(0.130)
lnprice4	-0.433^{***}
	(0.118)
lnprice5	0.311^{***}
	(0.088)
lnprice6	0.143^{**}
	(0.063)
Constant	6.601^{***}
	(0.227)
Observations	4,368
R^2	0.349
Adjusted R^2	0.348
Residual Std. Error	0.886 (df = 4359)
F Statistic	292.678^{***} (df = 8; 4359)
Note:	*p<0.1; **p<0.05; ***p<0.01




```
lmtest::bptest(linmod3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linmod3
## BP = 170.17, df = 8, p-value < 2.2e-16
```

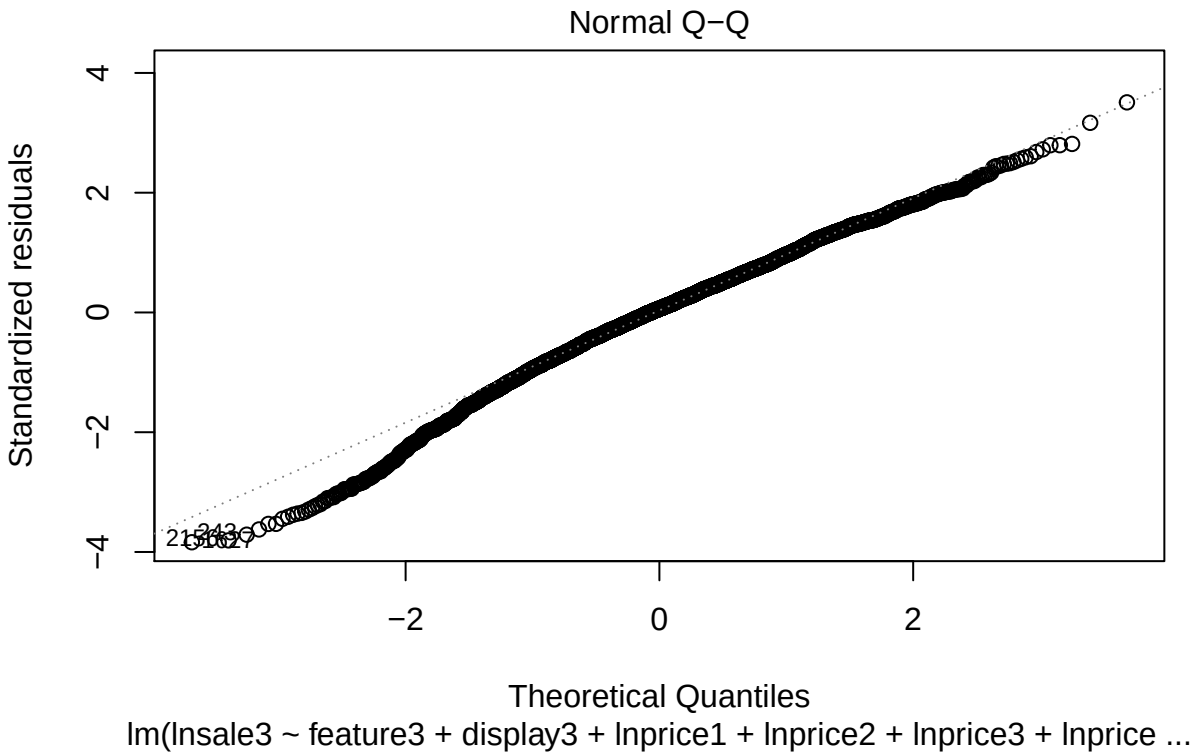
Although the dsitribution of residuals looks more normal than before, the red line is still curved, and according to the Breusch Pagan test, we again reject the Null Hypothesis that there is constant variance and conclude that there is evidence of heteroskedasticity.

Homosekdasticity: Fail

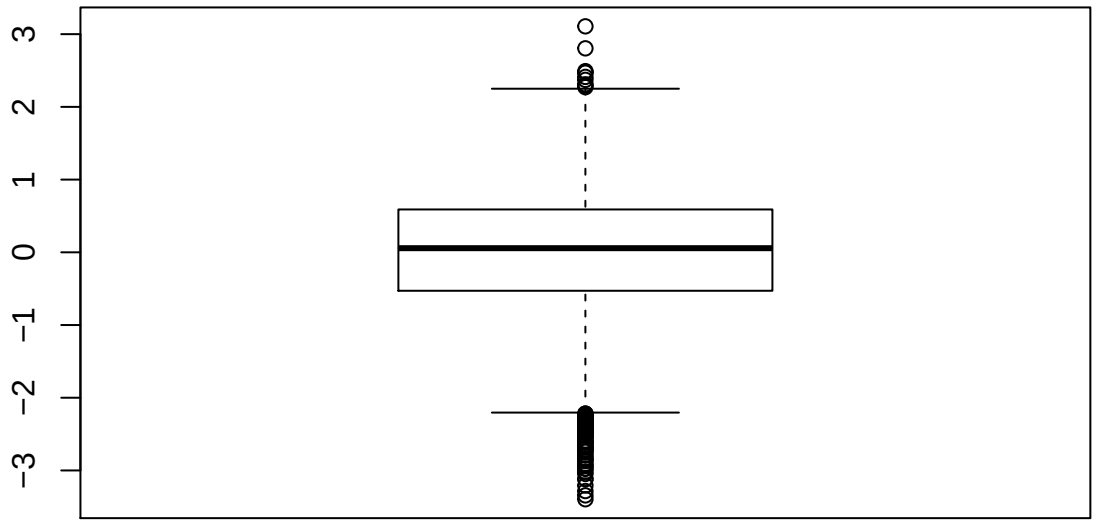
g)

We'll check for normality again using the Q-Q plot, boxplot, and the Shapiro-Wilk test

```
plot(linmod3, which = 2)
```




```
boxplot(residuals(linmod3))
```




```
shapiro.test(residuals(linmod3))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(linmod3)
## W = 0.98953, p-value < 2.2e-16
```

Similar story as our other assumption, we see that the distribution seems more normal than before, but still fails the visual tests of normality with the plots, as well as having a p-value on the Shapiro-Wilk test of less than .05, meaning we reject the Null Hypothesis that our residuals in this model are normally distributed

Normality: Fail

h)

Obviously Jack's has the biggest impact on sales, followed by Tombstone, DiGiorno, Freshchetta, Red Baron, and finally Tony's. This is determined by the absolute value of the magnitude of the coefficients on the price coefficients for each brand

i)

Since sales are also logged, the 1% increase in price of Jack's is expected on average to be associated with a 2.519% decrease in sales of Jack's frozen pizza

j)

Our new model has an R squared of .349 where as in Q5 our model had an R squared of .26. This is an increase of (.349-.26) = .089 or 8.9%. One might worry that this R squared is simply increasing due to the addition of variables, as it will always increase if more variables are added. However, if we check the adjusted R squared values which penalize the value when more variables are added, the values are very similar to the original R squared values so we don't need to worry about this here.

k)

All of the price coefficients of the other brands are statistically significant in terms of having a relationship with the amount of sales of Jack's brand. Interestingly, what we said in number 2 is reflect here. We said that Tombstone seemed to be competing with Jack's which is reflected in the fact that Tombstone's price is the second most important brand for determining sales of Jack's after Jack's own price.It does however, change part of my intuition when I thought Jack's would compete more with Red Baron than the high tiered price frozen pizza brands. In fact, according to our model, Jack's seem to compete more with DiGiorno and Freshchetta than with Red Baron.

Once we know DiGiorno is in the Kraft brand this competitve landscape makes much more sense. We see the 2 most competing brands of Jack's are Tombstone and DiGiorno. This makes sense and follows from the concept of cannibilization that we talked about in the class earlier in the semester. For instance, if Kraft lowers the price of its DiGiorno brand (high tier price), then some customers may be more willing to buy DiGiorno frozen pizza, but those customers may very likely come from Kraft's own customer base that would've bought Tombstone or Jack's instead.This is what we are seeing in our model output.

8

a)

b)

c)

d)

e)

f)

9

a)

We already found this in number 3 part c) where the max week for Jack's sales is week 3 when aavg unit sales are 653.52. We can account for this Demand "shock" by creating a dummy variable for Week 3 shown below:

```
# create new variable for week 3
data$week3 = (data$week == 3) * 1
# create new linear model with week 3 added
linmod4 = lm(lnsale3 ~ feature3 + display3 + lnprice1 + lnprice2 +
    lnprice3 + lnprice4 + lnprice5 + lnprice6 + week3, data = data)
summary(linmod4)
```

```
##
## Call:
## lm(formula = lnsale3 ~ feature3 + display3 + lnprice1 + lnprice2 +
##     lnprice3 + lnprice4 + lnprice5 + lnprice6 + week3, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3886 -0.5276  0.0556  0.5911  3.1128
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.59518    0.22670  29.093  < 2e-16 ***
## feature3     0.35214    0.03174  11.095  < 2e-16 ***
```

```
## display3     0.66313    0.03084  21.505  < 2e-16 ***
## lnprice1     0.47265    0.12763   3.703 0.000215 ***
## lnprice2     0.67671    0.07457   9.075  < 2e-16 ***
## lnprice3    -2.50377    0.12982 -19.287  < 2e-16 ***
## lnprice4    -0.42575    0.11813  -3.604 0.000317 ***
## lnprice5     0.32164    0.08730   3.684 0.000232 ***
## lnprice6     0.13962    0.06241   2.237 0.025313 *
## week3        0.48078    0.09772   4.920 8.99e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8836 on 4358 degrees of freedom
## Multiple R-squared:  0.353,  Adjusted R-squared:  0.3517
## F-statistic: 264.2 on 9 and 4358 DF,  p-value: < 2.2e-16
```

b)

We see that the R squared has increased from .3494 to .353, and increase of .0036 or .36%. So we are essentially explaining .36% more of the variance in the data by including this variable, which isn't that much, but it is probably worthwhile to hold this demand shock fixed in our model regardless.

c)

```
# stargazer(linmod4, title= 'Linear Model of Jack's LN(Sales)
# regressed on LN(Price) of every brand, Jack's Feature,
# Jack's Display, and Week 3')
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

Note: standard errors shown in parentheses below coefficient estimates

d)

We can find the effect of this week on sales by taking exp(.481) - 1 = 1.617691 - 1 = .617691 (percent increase in sales: 61.78%)

e)

You could also use a lagged sales variable to account for this. Essentially this lagged variables captures the effect of the previous week's sales on that of the current week's sales

```
z[order(z$x, z$Group.1), ]
```

Table 4: Linear Model of Jack's LN(Sales) regressed on LN(Price) of every brand, Jack's Feature, Jack's Display, and Week 3

	Dependent variable:
	lnsale3
feature3	0.352^{***}
	(0.032)
display3	0.663^{***}
	(0.031)
lnprice1	0.473^{***}
	(0.128)
lnprice2	0.677^{***}
	(0.075)
lnprice3	-2.504^{***}
	(0.130)
lnprice4	-0.426^{***}
	(0.118)
lnprice5	0.322^{***}
	(0.087)
lnprice6	0.140^{**}
	(0.062)
week3	0.481^{***}
	(0.098)
Constant	6.595^{***}
	(0.227)
Observations	4,368
R^2	0.353
Adjusted R^2	0.352
Residual Std. Error	0.884 (df = 4358)
F Statistic	264.233^{***} (df = 9; 4358)
Note:	*p<0.1; **p<0.05; ***p<0.01

```
##    Group.1        x
## 5        5 209.1786
## 25      25 222.9167
## 1        1 223.5952
## 20      20 229.7143
## 32      32 240.8214
## 15      15 250.2500
## 26      26 251.1548
## 24      24 252.7381
## 14      14 255.7262
## 47      47 259.7262
## 44      44 261.1548
## 45      45 263.0714
## 43      43 265.5119
## 48      48 270.4405
## 6        6 279.8571
## 31      31 279.9405
## 4        4 284.4405
## 35      35 291.1190
## 46      46 302.1667
## 22      22 304.4762
## 37      37 304.6786
## 52      52 305.3929
## 49      49 314.2500
## 30      30 314.4286
## 21      21 332.8810
## 28      28 335.5119
## 17      17 337.8810
## 23      23 338.1548
## 42      42 345.2143
## 36      36 346.0714
## 41      41 346.2857
## 9        9 350.3333
## 10      10 350.8214
## 16      16 353.0833
## 29      29 364.9405
## 7        7 366.2024
## 51      51 367.9048
## 38      38 369.2381
## 8        8 380.8214
## 19      19 382.1190
## 27      27 401.3452
## 50      50 415.5952
## 13      13 431.5833
## 11      11 438.4286
## 40      40 460.2976
## 33      33 472.4881
## 2        2 505.3452
## 12      12 513.4048
## 39      39 528.8095
## 18      18 543.1905
## 34      34 567.7738
## 3        3 653.5238
```

We can see from the order function that weeks 47 and 52 also have sharp spikes in demand

---

10

a) Note: Assuming for this problem that you mean for Jack's brand and not all brand

Below we see first a frequency table and then a relative frequency table

```
labeledata <- data.frame(Feature = data$feature3, Display = data$display3)

# get frequency table
frequencytable <- table(labeledata)
frequencytable
```

```
##         Display
## Feature    0    1
##       0 2112  596
##       1  775  885
```

```
# get proportions table
percentagetable <- prop.table(frequencytable)
percentagetable
```

```
##         Display
## Feature         0         1
##       0 0.4835165 0.1364469
##       1 0.1774267 0.2026099
```

b)

As we can see in the above relative frequency table, Jack's is on feature and display about 20% of the time

c)

```
# display only variable
data$disponly3 = (data$feature3 == 0 & data$display3 == 1) *
    1
# feature only variable
data$featonly3 = (data$feature3 == 1 & data$display3 == 0) *
    1
# feature and display
data$featdisp3 = (data$feature3 == 1 & data$display3 == 1) *
    1
```

d)

```
linmod5 = lm(lnsale3 ~ featonly3 + disponly3 + featdisp3 + lnprice1 +
    lnprice2 + lnprice3 + lnprice4 + lnprice5 + lnprice6 + week3,
    data = data)
# summary(linmod5) stargazer(linmod5, title= 'Linear Model of
# Jack's LN(Sales) regressed on LN(Price) of every brand,
# Jack's Feature only, Jack's Display only, Feature and
# Display, and Week 3')
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

e)

So now the feature only coefficient measures the effect of Jack's being featured on sales of Jack's (when display is fixed at 0). The display only coefficient measures the effect if Jack's being on display on sales of Jack's (when feature is fixed at 0). Finally, featdisp3 measure the effect of Jack's being featured AND on display on sales of Jack's.

f)

There was modest improvment in R squared from .353 to .356 (an increase of .003 or .3%)

11

a)

```
# create lagged LN sales variable for Jack's
data$laglnsale = NA
data$laglnsale[2:nrow(data)] = log(data$sale3[1:(nrow(data) -
    1)])
data$laglnsale[data$week == 1] = NA
# get length of lagged ln sales vector without NA's included
length(data$laglnsale[!is.na(data$laglnsale)])
```

```
## [1] 4284
```

Table 5: Linear Model of Jack's LN(Sales) regressed on LN(Price) of every brand, Jack's Feature only, Jack's Display only, Feature and Display, and Week 3

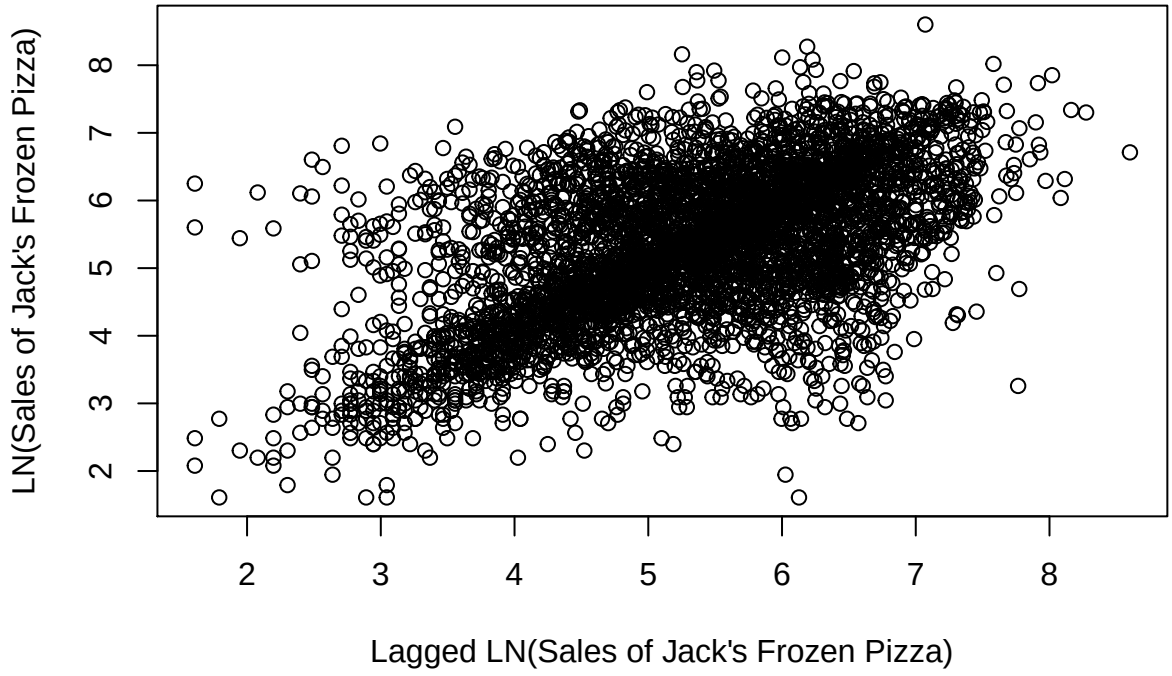
	Dependent variable:
	lnsale3
featonly3	0.460^{***}
	(0.040)
disponly3	0.792^{***}
	(0.042)
featdisp3	0.983^{***}
	(0.040)
lnprice1	0.462^{***}
	(0.127)
lnprice2	0.667^{***}
	(0.074)
lnprice3	-2.467^{***}
	(0.130)
lnprice4	-0.429^{***}
	(0.118)
lnprice5	0.326^{***}
	(0.087)
lnprice6	0.136^{**}
	(0.062)
week3	0.469^{***}
	(0.098)
Constant	6.551^{***}
	(0.226)
Observations	4,368
R^2	0.356
Adjusted R^2	0.355
Residual Std. Error	0.882 (df = 4357)
F Statistic	240.862^{***} (df = 10; 4357)
Note:	*p<0.1; **p<0.05; ***p<0.01

There are 4284 observations of the laggedlnsales variable

b)

Below we plot the scatter plot of logged sales of Jack's frozen pizza and its lagged version and see a positive correlation between the two variables.

```
plot(data$laglnsale, data$lnsale3, xlab = "Lagged LN(Sales of Jack's Frozen Pizza)",
    ylab = "LN(Sales of Jack's Frozen Pizza)")
```



c)

```
linmod6 = lm(lnsale3 ~ featonly3 + disponly3 + featdisp3 + lnprice1 +
    lnprice2 + lnprice3 + lnprice4 + lnprice5 + lnprice6 + week3 +
    laglnsale, data = data)
# summary(linmod6) stargazer(linmod6, title= 'Linear Model of
# Jack's LN(Sales) regressed on LN(Price) of every brand,
# Jack's Feature only, Jack's Display only, Feature and
# Display, Week 3, and lagged LN(Sales)')
```

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu

Table 6: Linear Model of Jack's LN(Sales) regressed on LN(Price) of every brand, Jack's Feature only, Jack's Display only, Feature and Display, Week 3, and lagged LN(Sales)

	Dependent variable:
	lnsale3
featonly3	0.576^{***}
	(0.033)
disponly3	0.545^{***}
	(0.035)
featdisp3	0.945^{***}
	(0.034)
lnprice1	0.220^{**}
	(0.106)
lnprice2	0.646^{***}
	(0.062)
lnprice3	-1.983^{***}
	(0.109)
lnprice4	-0.157
	(0.099)
lnprice5	0.535^{***}
	(0.072)
lnprice6	0.092^{*}
	(0.052)
week3	0.347^{***}
	(0.080)
laglnsale	0.468^{***}
	(0.010)
Constant	3.162^{***}
	(0.202)
Observations	4,284
R^2	0.566
Adjusted R^2	0.565
Residual Std. Error	0.726 (df = 4272)
F Statistic	506.019^{***} (df = 11; 4272)
Note:	*p<0.1; **p<0.05; ***p<0.01

d)

We see that this model improves substantially over the previous model. The R squared has increased from .356 to .5658 (this is an increase of .2098 or 20.98%!) It seems like lagged sales has a very big effect on sales in the current period. This makes a lot of sense intuitively, and is common among most time series data.

4

a) If you looks at his notes, I think you have to make store a factor and include it in the model as each store ID being its own dummy variable