Model Specification

ECO 6416

2022-11-01

Contents

Here are all the packages needed to get started.

```
library(readxl) # reading in excel file
library(car) # for vif function
```

```
## Loading required package: carData
```

```
library(plotly) # for interactive visualizations
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] plotly_4.10.0 ggplot2_3.3.6 car_3.1-0     carData_3.0-5 readxl_1.4.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2  xfun_0.31         purrr_0.3.4       colorspace_2.0-3
##  [5] vctrs_0.4.1       generics_0.1.3    htmltools_0.5.2   viridisLite_0.4.0
##  [9] yaml_2.3.5        utf8_1.2.2        rlang_1.0.3       pillar_1.7.0
## [13] glue_1.6.2        withr_2.5.0       DBI_1.1.3         lifecycle_1.0.1
## [17] stringr_1.4.0     munsell_0.5.0     gtable_0.3.0      cellranger_1.1.0
## [21] htmlwidgets_1.5.4 evaluate_0.15     knitr_1.39        fastmap_1.1.0
## [25] fansi_1.0.3       scales_1.2.0      jsonlite_1.8.0    abind_1.4-5
## [29] digest_0.6.29     stringi_1.7.8     dplyr_1.0.9       grid_4.2.1
## [33] cli_3.3.0         tools_4.2.1       magrittr_2.0.3    lazyeval_0.2.2
## [37] tibble_3.1.7      crayon_1.5.1      tidyr_1.2.0       pkgconfig_2.0.3
## [41] ellipsis_0.3.2    data.table_1.14.2 assertthat_0.2.1  rmarkdown_2.14
## [45] httr_1.4.3        rstudioapi_0.14   R6_2.5.1          compiler_4.2.1
```

1 Datasets

To demonstrate much of this, let's use some of the datasets we've already been using:

```
mail <- read_xlsx("../Data/First Class Mail.xlsx")
covid <- read_xlsx("../Data/Covid.xlsx")
```

Let's check some model assumptions.

2 Multicollinearity

We can check how variables are related pairwise:

```
cor(mail)
```

```
##                   Yr FirstClVol    PopUSA     Price      Time
## Yr         1.0000000  0.6297689 0.9981437 0.9947384 1.0000000
## FirstClVol 0.6297689  1.0000000 0.6112215 0.6080891 0.6297689
## PopUSA     0.9981437  0.6112215 1.0000000 0.9935780 0.9981437
## Price      0.9947384  0.6080891 0.9935780 1.0000000 0.9947384
## Time       1.0000000  0.6297689 0.9981437 0.9947384 1.0000000
```

We may have some strongly correlated variables in this situation. Year, Price, and Population are all directly related. Additionally, we have perfect correlation on year and Time.

If we were to ignore these signs and continue forward:

```
mail_model <- lm(FirstClVol ~. , data = mail)

summary(mail_model)
```

```
##
## Call:
## lm(formula = FirstClVol ~ ., data = mail)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.721  -9.379  -2.381  13.465  29.161
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.596e+04  4.239e+03  -3.765 0.000401 ***
## Yr           8.367e+00  2.222e+00   3.765 0.000401 ***
## PopUSA      -2.210e+00  7.812e-01  -2.829 0.006472 **
## Price       -2.136e+00  1.357e+00  -1.575 0.120894
## Time                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.34 on 56 degrees of freedom
## Multiple R-squared:  0.5002, Adjusted R-squared:  0.4734
## F-statistic: 18.68 on 3 and 56 DF,  p-value: 1.605e-08
```

The model catches the perfectly correlated variables. Let's drop them and check the group-wise comparisons.

```
mail_model <- lm(FirstClVol ~. , data = mail[,-1])

vif(mail_model)
```

```
##    PopUSA     Price      Time
## 272.90951  96.44439 332.90106
```

Those numbers are really high, so our model may be impacted by multicollinearity. Let's see:

```
summary(mail_model)
```

```
##
## Call:
## lm(formula = FirstClVol ~ ., data = mail[, -1])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.721  -9.379  -2.381  13.465  29.161
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 430.2370   137.2978   3.134 0.002747 **
## PopUSA       -2.2099     0.7812  -2.829 0.006472 **
## Price        -2.1365     1.3565  -1.575 0.120894
## Time          8.3667     2.2221   3.765 0.000401 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 16.34 on 56 degrees of freedom
## Multiple R-squared:  0.5002, Adjusted R-squared:  0.4734
## F-statistic: 18.68 on 3 and 56 DF,  p-value: 1.605e-08
```

Since year and population test significant, we see that it isn't impacting those variables, but it is most likely impacting the price coefficient.

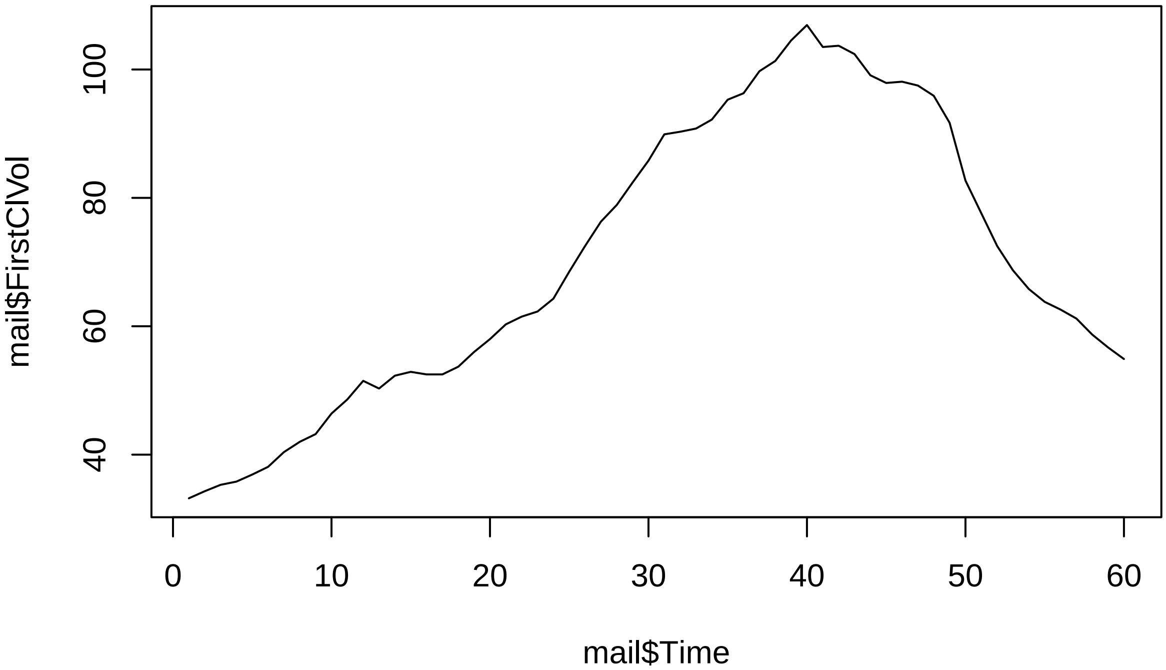
3 Functional Form

Assuming something is linear, when in fact, it isn't can cause issues as well.

3.1 Mail Volume

We can see that there was a peak in mail volume

```
plot(mail$Time,mail$FirstClVol, type = "l")
```



so a line is not best fit. Let's generate a different model

```
mail$TimeSq <- mail$Time^2

mail_model <- lm(FirstClVol ~. , data = mail[,-1])
summary(mail_model)
```

```
##
## Call:
## lm(formula = FirstClVol ~ ., data = mail[, -1])
```

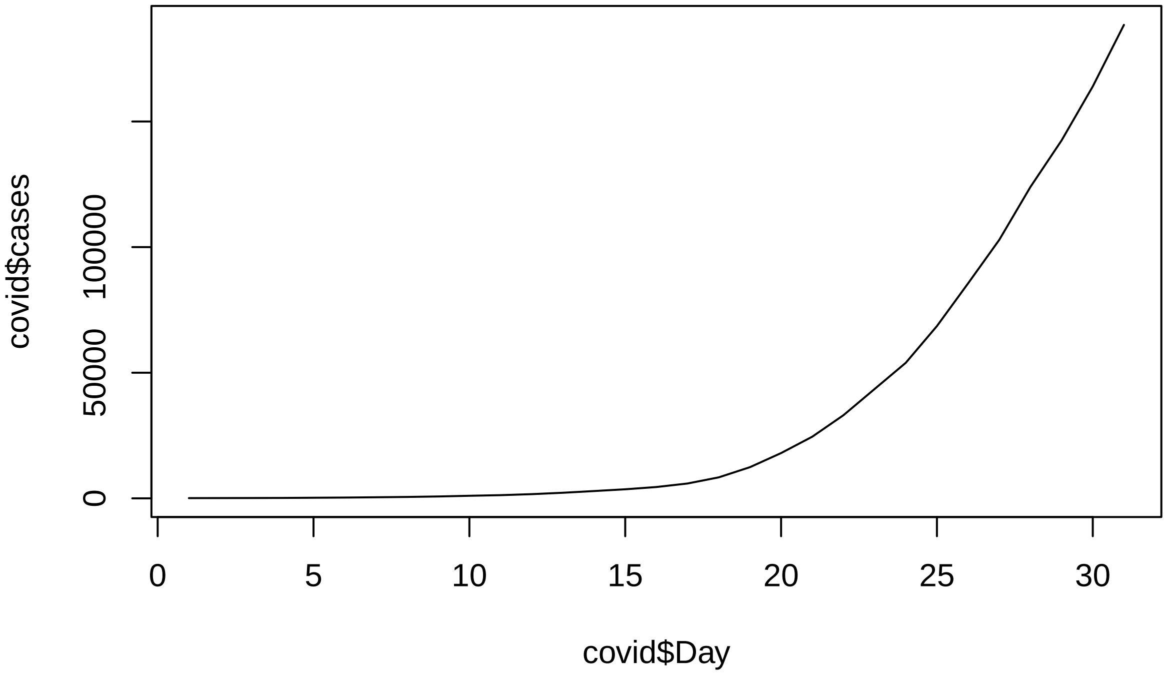
```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.242  -5.352   0.683   6.045  15.160
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.957e+02  9.968e+01  -5.976 1.76e-07 ***
## PopUSA       3.373e+00  5.534e-01   6.095 1.13e-07 ***
## Price        3.102e+00  7.548e-01   4.109 0.000133 ***
## Time        -5.406e+00  1.465e+00  -3.691 0.000515 ***
## TimeSq      -8.482e-02  6.183e-03 -13.717  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.841 on 55 degrees of freedom
## Multiple R-squared:  0.8869, Adjusted R-squared:  0.8787
## F-statistic: 107.9 on 4 and 55 DF,  p-value: < 2.2e-16
```

We now have everything testing significant and our overall fit his increased, but we still have issues with the price coefficient. It test significant, but in the wrong direction. We still have issues here that need further investigation.

3.2 Covid Cases

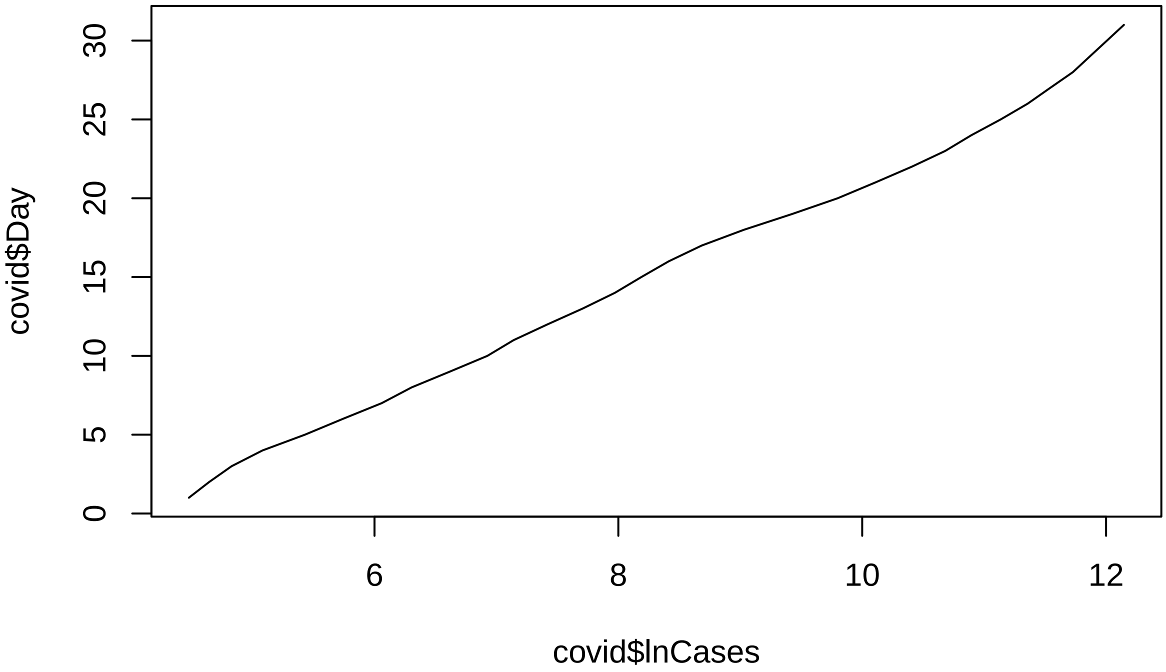
We know that covid cases were also non-linear.

```
plot(covid$Day,covid$cases,type = "l")
```



so we can transform the dependent variable and get something linear:

```
covid$lnCases <- log(covid$cases)

plot(covid$lnCases, covid$Day, type = "l")
```



```
covid_model <- lm(lnCases~Day, data = covid)
summary(covid_model)
```

```
##
## Call:
## lm(formula = lnCases ~ Day, data = covid)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45434 -0.06623  0.00238  0.07979  0.26237
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.142903   0.059430   69.71   <2e-16 ***
## Day         0.272842   0.003242   84.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1615 on 29 degrees of freedom
## Multiple R-squared:  0.9959, Adjusted R-squared:  0.9958
## F-statistic:  7082 on 1 and 29 DF,  p-value: < 2.2e-16
```

In March 2020, the number of COVID cases were increasing by 27.28% each day give or take 0.64%.

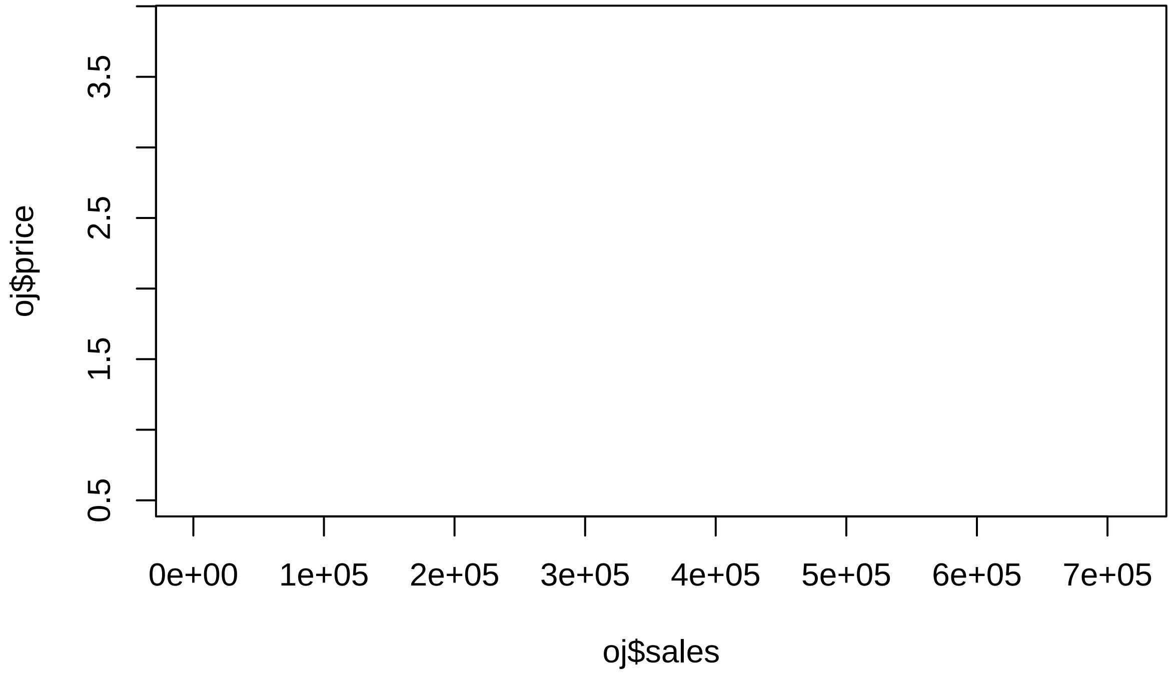
4 Orange Juice Demand

The following data is from a Chicago-area chain called Dominick's. This data was collected in the 1990s and the data include weekly prices and sales (in number of cartons "moved") for 3 orange juice brands — Tropicana, Minute Maid, Dominick's — at 83 Chicagoland Stores, as well as an indicator, `feat`, showing whether each brand was advertised (in store or flyer) that week.

```
oj <- read.csv("../Data/oj.csv")
```

4.1 Visualization

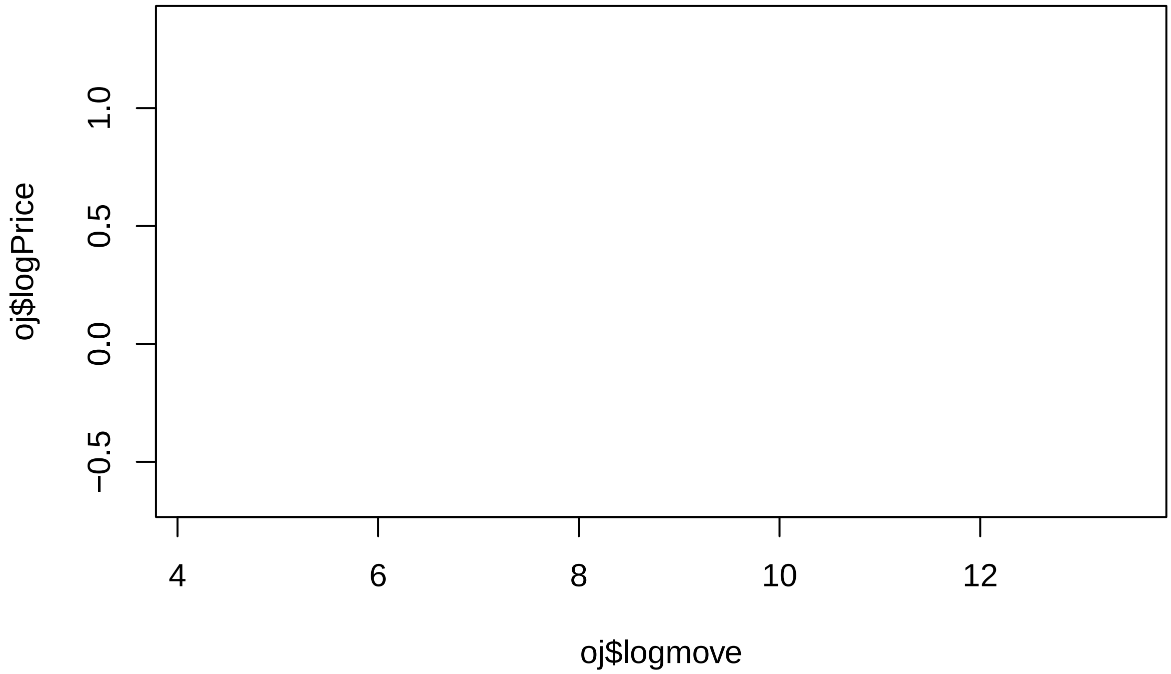
```
brandcol <- c("green","red","gold")

plot(oj$sales,oj$price, col=brandcol[oj$brand])
```



4.2 Functional Forms

You can see that the relationships between price and quantity demanded are nonlinear. We need to transform them first:

```
oj$logPrice <- log(oj$price)
oj$logmove <- log(oj$sales)

plot(oj$logmove,oj$logPrice, col=brandcol[oj$brand])
```



4.3 Omitted Variable Bias

Suppose we ran this model instead:

```
reg <- lm(log(sales) ~ log(price), data=oj)

summary(reg)
```

```
##
## Call:
## lm(formula = log(sales) ~ log(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(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
```

This states that a 1% increase in price constitutes a 1.6% decrease in quantity demanded. This is elastic.

Ignoring the fact that different brands have different demands will cause issues.

```
reg <- lm(log(sales) ~ log(price) + brand, data=oj)

summary(reg)
```

```
##
## Call:
## lm(formula = log(sales) ~ log(price) + brand, data = oj)
##
## 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(price)        -3.13869    0.02293 -136.89   <2e-16 ***
## brandminute.maid   0.87017    0.01293   67.32   <2e-16 ***
## brandtropicana     1.52994    0.01631   93.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7935 on 28943 degrees of freedom
## Multiple R-squared:  0.3941, Adjusted R-squared:  0.394
## F-statistic:  6275 on 3 and 28943 DF,  p-value: < 2.2e-16
```

This states that a 1% increase in price constitutes a 3.3% decrease in quantity demanded. These are wildly different results.

There is a flaw in both these models though. It assumes that price elasticity is the same regardless of brand. Let's fix that.

4.4 Interaction Terms

```
reg_interact <-  lm(log(sales) ~ log(price)*brand, data=oj)

summary(reg_interact)
```

```
##
## Call:
## lm(formula = log(sales) ~ log(price) * brand, data = oj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4434 -0.5232 -0.0494  0.4884  3.4901
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  10.95468    0.02070 529.136   <2e-16 ***
## log(price)                   -3.37753    0.03619 -93.322   <2e-16 ***
## brandminute.maid              0.88825    0.04155  21.376   <2e-16 ***
## brandtropicana                0.96239    0.04645  20.719   <2e-16 ***
## log(price):brandminute.maid   0.05679    0.05729   0.991    0.322
## log(price):brandtropicana    0.66576    0.05352  12.439   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7911 on 28941 degrees of freedom
## Multiple R-squared:  0.3978, Adjusted R-squared:  0.3977
## F-statistic:  3823 on 5 and 28941 DF,  p-value: < 2.2e-16
```

We can essentially see 3 different demand equations, one for each brand. What if we assume that demand for the good changes if it is advertised?

4.4.1 More Interactions

```
full_reg <- lm(log(sales) ~ log(price)*brand*feat, data=oj)

summary(full_reg)
```

```
##
## Call:
## lm(formula = log(sales) ~ log(price) * brand * feat, data = oj)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8893 -0.4290 -0.0091  0.4125  3.2368
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      10.40658    0.02335 445.668  < 2e-16 ***
## log(price)                       -2.77415    0.03883 -71.445  < 2e-16 ***
## brandminute.maid                  0.04720    0.04663   1.012    0.311
## brandtropicana                    0.70794    0.05080  13.937  < 2e-16 ***
## feat                              1.09441    0.03810  28.721  < 2e-16 ***
## log(price):brandminute.maid       0.78293    0.06140  12.750  < 2e-16 ***
## log(price):brandtropicana         0.73579    0.05684  12.946  < 2e-16 ***
## log(price):feat                  -0.47055    0.07409  -6.351 2.17e-10 ***
## brandminute.maid:feat             1.17294    0.08196  14.312  < 2e-16 ***
## brandtropicana:feat               0.78525    0.09875   7.952 1.90e-15 ***
## log(price):brandminute.maid:feat -1.10922    0.12225  -9.074  < 2e-16 ***
## log(price):brandtropicana:feat   -0.98614    0.12411  -7.946 2.00e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.695 on 28935 degrees of freedom
## Multiple R-squared:  0.5354, Adjusted R-squared:  0.5352
## F-statistic:  3031 on 11 and 28935 DF,  p-value: < 2.2e-16
```

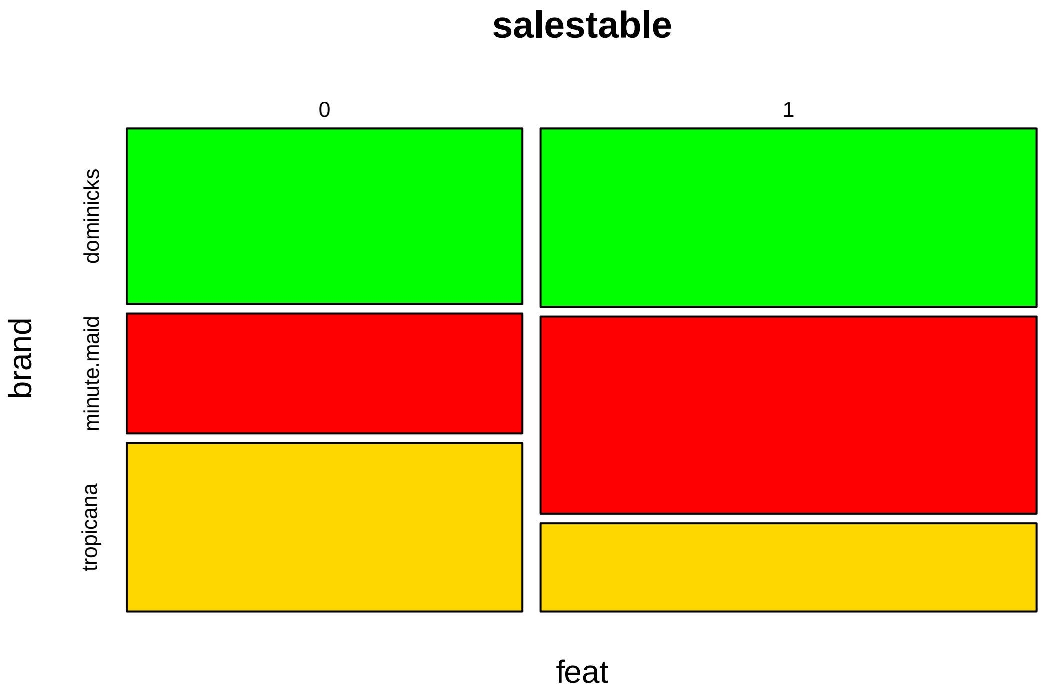
Now we essentially have 6 demand equations; two for each brand based on whether or not it was advertised. We can also view this as a table:

```
b <- coef(full_reg)

Ad <- c("Not Featured","Featured")
Dominicks <- c(b["log(price)"],
               b["log(price)"] + b["log(price):feat"])
MinuteMaid <- c(b["log(price)"] + b["log(price):brandminute.maid"],
                b["log(price)"] + b["log(price):brandminute.maid"] + b["log(price):feat"] + b["log(pric
                )
Tropicana <- c(b["log(price)"] + b["log(price):brandtropicana"],
               b["log(price)"] + b["log(price):brandtropicana"] + b["log(price):feat"] + b["log(price):


data.frame("Advertising" = Ad,
           "Dominicks" = round(Dominicks,digits = 1),
           "MinuteMaid" = round(MinuteMaid,digits = 1),
           "Tropicana" = round(Tropicana,digits = 1))
```

```
##    Advertising Dominicks MinuteMaid Tropicana
## 1 Not Featured      -2.8       -2.0      -2.0
## 2     Featured      -3.2       -3.6      -3.5
```

4.5 Table explaining why ads confounded our brand elasticity estimates

```
salestable <- tapply(exp(oj$logmove), oj[,c("feat","brand")], sum)
mosaicplot(salestable,col=c("green","red","gold"))
```



Minute maid was featured more often than Tropicana. Since being featured leads to more price sensitivity, it lead to Minute maid appearing more price sensitive.