Project 3

```
data <- read_csv("OTC_Data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   store = col_double(),
##   week = col_double(),
##   brand_name = col_character(),
##   size = col_double(),
##   sales = col_double(),
##   count = col_double(),
##   price = col_double(),
##   cost = col_double()
## )
```

```
nrow(distinct(data,store))
```

```
## [1] 73
```

```
#73 distinct stores

nrow(distinct(data,week))
```

```
## [1] 48
```

```
# 48 distinct weeks

nrow(distinct(data,brand_name))*nrow(distinct(data,size))
```

```
## [1] 9
```

```
# 9 distinct products

nrow(distinct(data,brand_name))
```

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

```
# 3 distinct brands
```

calculate total sales and total expenditure of each store-week as well as market share and expenditure share (wi). Stone index and 18 variables for price and cost.

```
data2 <- data
data2 = data2[FALSE,]
options(digits=3)
for (i in unique(data$week)){
  for (j in unique(data$store)){
    test <- filter(data, week == i, store == j)
    test <- test %>% mutate(sales_total = sum(test$sales), mkt_share = (sales/sales_total))

    ty <- filter(test, brand_name == "Tylenol")
    ad <- filter(test, brand_name == "Advil")
    ba <- filter(test, brand_name == "Bayer")
```

```
    ty_total = sum(ty$sales*ty$price)
    ad_total = sum(ad$sales*ad$price)
    ba_total = sum(ba$sales*ba$price)

    test <- test %>% mutate(exp_total = ifelse(brand_name == "Tylenol", ty_total, ifelse(brand_name ==

    test$exp_total = as.numeric(as.character(test$exp_total))

    test <- test %>% mutate(exp_share = (price*sales)/exp_total)

    ty <- filter(test, brand_name == "Tylenol")
    ad <- filter(test, brand_name == "Advil")
    ba <- filter(test, brand_name == "Bayer")

    ty <- ty %>% mutate(lprice = log(price))
    ad <- ad %>% mutate(lprice = log(price))
    ba <- ba %>% mutate(lprice = log(price))

    ty_stone = sum(ty$exp_share*ty$lprice)
    ad_stone = sum(ad$exp_share*ad$lprice)
    ba_stone = sum(ba$exp_share*ba$lprice)

    test <- test %>% mutate(stone = ifelse(brand_name == "Tylenol", ty_stone, ifelse(brand_name == "Adv

    test$stone = as.numeric(as.character(test$stone))

    test <- test %>% mutate(ty25p = price[1],ty50p = price[2], ty100p = price[3], ad25p = price[4], ad5

    test <- test %>% mutate(ty25c = cost[1],ty50c = cost[2], ty100c = cost[3], ad25c = cost[4], ad50c =

    data2 <- rbind(data2, test)
    }}
```

summary statistics

```
options(scipen=100)
options(digits=3)

## Tylenol
ty25 <- filter(data2, brand_name == "Tylenol", size == "25")
stat.desc(ty25, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median          1752.500   78.000  24.500         NA   25 14.000
## mean            1752.500   74.767  24.500         NA   25 15.108
## SE.mean           17.090    0.563   0.234         NA    0  0.109
## CI.mean.0.95      33.508    1.105   0.459         NA    0  0.214
## var          1023460.000 1112.195 191.971         NA    0 41.724
## std.dev         1011.662   33.350  13.855         NA    0  6.459
## coef.var           0.577    0.446   0.566         NA    0  0.428
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 3.50000 2.23000      80.000  0.175439   206.200
## mean            19609.354 3.42047 2.18227      86.417  0.178107   228.198
```

```
## SE.mean            89.894 0.00458 0.00303       0.530  0.000959     1.578
## CI.mean.0.95      176.249 0.00899 0.00594       1.039  0.001880     3.094
## var          28315331.915 0.07363 0.03215     983.457  0.003221  8724.123
## std.dev          5321.215 0.27136 0.17930      31.360  0.056756    93.403
## coef.var            0.271 0.07933 0.08216       0.363  0.318665     0.409
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.22823 1.63649 3.50000 4.94000 7.04000 2.97000 5.29000
## mean           0.23437 1.63647 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00131 0.00119 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00257 0.00232 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.00604 0.00492 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.07774 0.07016 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.33169 0.04287 0.07933 0.05300 0.07452 0.06715 0.05701
##               ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```
# price: mean 3.42, sd 0.271
# wholesale price: mean 2.182, sd 0.179
# market share: mean 0.178, sd 0.057

ty50 <- filter(data2, brand_name == "Tylenol", size == "50")
stat.desc(ty50, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median          5256.500   78.000  24.500         NA   50 17.000
## mean            5256.500   74.767  24.500         NA   50 18.719
## SE.mean           17.090    0.563   0.234         NA    0  0.167
## CI.mean.0.95      33.508    1.105   0.459         NA    0  0.328
## var          1023460.000 1112.195 191.971         NA    0 97.874
## std.dev         1011.662   33.350  13.855         NA    0  9.893
## coef.var           0.192    0.446   0.566         NA    0  0.528
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 4.94000 3.73000      80.000   0.20690   206.200
## mean            19609.354 4.94202 3.67193      86.417   0.21277   228.198
## SE.mean            89.894 0.00442 0.00308       0.530   0.00111     1.578
## CI.mean.0.95      176.249 0.00867 0.00604       1.039   0.00218     3.094
## var          28315331.915 0.06859 0.03321     983.457   0.00431  8724.123
## std.dev          5321.215 0.26191 0.18224      31.360   0.06567    93.403
## coef.var            0.271 0.05300 0.04963       0.363   0.30866     0.409
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.39263 1.63649 3.50000 4.94000 7.04000 2.97000 5.29000
```

```
## mean           0.39906 1.63647 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00173 0.00119 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00340 0.00232 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.01054 0.00492 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.10265 0.07016 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.25722 0.04287 0.07933 0.05300 0.07452 0.06715 0.05701
##                ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```r
# price: mean 4.942, sd 0.262
# wholesale price: mean 3.672, sd 0.182
# market share: mean 0.213, sd 0.0657

ty100 <- filter(data2, brand_name == "Tylenol", size == "100")
stat.desc(ty100, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median          8760.500   78.000  24.500         NA  100 11.000
## mean            8760.500   74.767  24.500         NA  100 12.270
## SE.mean           17.090    0.563   0.234         NA    0  0.119
## CI.mean.0.95      33.508    1.105   0.459         NA    0  0.234
## var          1023460.000 1112.195 191.971         NA    0 49.931
## std.dev         1011.662   33.350  13.855         NA    0  7.066
## coef.var           0.115    0.446   0.566         NA    0  0.576
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 7.04000 5.71000      80.000   0.13576   206.200
## mean            19609.354 7.01607 5.75482      86.417   0.13961   228.198
## SE.mean            89.894 0.00883 0.00517       0.530   0.00090     1.578
## CI.mean.0.95      176.249 0.01732 0.01013       1.039   0.00177     3.094
## var          28315331.915 0.27333 0.09356     983.457   0.00284  8724.123
## std.dev          5321.215 0.52281 0.30587      31.360   0.05330    93.403
## coef.var            0.271 0.07452 0.05315       0.363   0.38181     0.409
##                exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median           0.36822 1.63649 3.50000 4.94000 7.04000 2.97000 5.29000
## mean             0.36658 1.63647 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean          0.00187 0.00119 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95     0.00367 0.00232 0.00899 0.00867 0.01732 0.00659 0.00971
## var              0.01225 0.00492 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev          0.11066 0.07016 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var         0.30187 0.04287 0.07933 0.05300 0.07452 0.06715 0.05701
##                ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
```

```
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```r
# price: mean 4.942, sd 0.262
# wholesale price: mean 3.672, sd 0.182
# market share: mean 0.140, sd 0.053

## Advil
ad25 <- filter(data2, brand_name == "Advil", size == "25")
stat.desc(ad25, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median         12264.5000   78.000  24.500         NA   25 12.000
## mean           12264.5000   74.767  24.500         NA   25 12.331
## SE.mean           17.0904    0.563   0.234         NA    0  0.101
## CI.mean.0.95      33.5082    1.105   0.459         NA    0  0.198
## var          1023460.0000 1112.195 191.971         NA    0 35.637
## std.dev         1011.6620   33.350  13.855         NA    0  5.970
## coef.var           0.0825    0.446   0.566         NA    0  0.484
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 2.97000 2.02000      80.000  0.144231    97.925
## mean            19609.354 2.96365 2.03000      86.417  0.144054   106.968
## SE.mean            89.894 0.00336 0.00120       0.530  0.000885     0.842
## CI.mean.0.95      176.249 0.00659 0.00235       1.039  0.001735     1.651
## var          28315331.915 0.03961 0.00505     983.457  0.002743  2485.866
## std.dev          5321.215 0.19901 0.07105      31.360  0.052371    49.858
## coef.var            0.271 0.06715 0.03500       0.363  0.363549     0.466
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.35733 1.55025 3.50000 4.94000 7.04000 2.97000 5.29000
## mean           0.36035 1.55621 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00233 0.00204 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00457 0.00400 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.01901 0.01461 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.13788 0.12088 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.38263 0.07768 0.07933 0.05300 0.07452 0.06715 0.05701
##                ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
```

```
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```
# price: mean 2.964, sd 0.199
# wholesale price: mean 2.03, sd 0.071
# market share: mean 0.144, sd 0.052

ad50 <- filter(data2, brand_name == "Advil", size == "50")
stat.desc(ad50, basic = F)
```

```
##                       X1    store    week brand_name size  sales
## median         15768.5000   78.000  24.500         NA   50  7.000
## mean           15768.5000   74.767  24.500         NA   50  8.143
## SE.mean           17.0904    0.563   0.234         NA    0  0.112
## CI.mean.0.95      33.5082    1.105   0.459         NA    0  0.219
## var          1023460.0000 1112.195 191.971         NA    0 43.823
## std.dev         1011.6620   33.350  13.855         NA    0  6.620
## coef.var           0.0642    0.446   0.566         NA    0  0.813
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 5.29000 3.63000      80.000  0.084337    97.925
## mean            19609.354 5.14502 3.62258      86.417  0.092132   106.968
## SE.mean            89.894 0.00496 0.00239       0.530  0.000931     0.842
## CI.mean.0.95      176.249 0.00971 0.00469       1.039  0.001825     1.651
## var          28315331.915 0.08603 0.02006     983.457  0.003037  2485.866
## std.dev          5321.215 0.29331 0.14164      31.360  0.055108    49.858
## coef.var            0.271 0.05701 0.03910       0.363  0.598134     0.466
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.36149 1.55025 3.50000 4.94000 7.04000 2.97000 5.29000
## mean           0.37098 1.55621 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00252 0.00204 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00494 0.00400 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.02228 0.01461 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.14928 0.12088 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.40239 0.07768 0.07933 0.05300 0.07452 0.06715 0.05701
##               ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
```

```
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```
# price: mean 5.145, sd 0.293
# wholesale price: mean 3.623, sd 0.142
# market share: mean 0.092, sd 0.055

ad100 <- filter(data2, brand_name == "Advil", size == "100")
stat.desc(ad100, basic = F)
```

```
##                        X1    store    week brand_name size  sales
## median         19272.5000   78.000  24.500         NA  100 3.0000
## mean           19272.5000   74.767  24.500         NA  100 3.6935
## SE.mean           17.0904    0.563   0.234         NA    0 0.0503
## CI.mean.0.95      33.5082    1.105   0.459         NA    0 0.0986
## var          1023460.0000 1112.195 191.971         NA    0 8.8646
## std.dev         1011.6620   33.350  13.855         NA    0 2.9774
## coef.var           0.0525    0.446   0.566         NA    0 0.8061
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 8.29000 6.09000      80.000  0.035714    97.925
## mean            19609.354 8.15974 6.09102      86.417  0.043085   106.968
## SE.mean            89.894 0.00581 0.00381       0.530  0.000508     0.842
## CI.mean.0.95      176.249 0.01140 0.00747       1.039  0.000996     1.651
## var          28315331.915 0.11843 0.05092     983.457  0.000905  2485.866
## std.dev          5321.215 0.34414 0.22566      31.360  0.030082    49.858
## coef.var            0.271 0.04217 0.03705       0.363  0.698191     0.466
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.25176 1.55025 3.50000 4.94000 7.04000 2.97000 5.29000
## mean           0.26867 1.55621 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00228 0.00204 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00447 0.00400 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.01824 0.01461 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.13505 0.12088 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.50268 0.07768 0.07933 0.05300 0.07452 0.06715 0.05701
##               ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```
# price: mean 8.160, sd 0.344
# wholesale price: mean 6.091, sd 0.226
# market share: mean 0.043, sd 0.030
```

```r
## Bayer
ba25 <- filter(data2, brand_name == "Bayer", size == "25")
stat.desc(ba25, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median         22776.5000   78.000  24.500         NA   25 4.0000
## mean           22776.5000   74.767  24.500         NA   25 4.2300
## SE.mean           17.0904    0.563   0.234         NA    0 0.0415
## CI.mean.0.95      33.5082    1.105   0.459         NA    0 0.0814
## var          1023460.0000 1112.195 191.971         NA    0 6.0356
## std.dev         1011.6620   33.350  13.855         NA    0 2.4567
## coef.var           0.0444    0.446   0.566         NA    0 0.5808
##                     count   price    cost sales_total mkt_share exp_total
## median          18492.000 2.62000 1.84000      80.000  0.046875    53.320
## mean            19609.354 2.67305 1.84650      86.417  0.051516    57.046
## SE.mean            89.894 0.00544 0.00262       0.530  0.000495     0.422
## CI.mean.0.95      176.249 0.01066 0.00514       1.039  0.000971     0.828
## var          28315331.915 0.10364 0.02407     983.457  0.000859   625.228
## std.dev          5321.215 0.32193 0.15513      31.360  0.029304    25.005
## coef.var            0.271 0.12044 0.08402       0.363  0.568834     0.438
##              exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median         0.19523 1.27204 3.50000 4.94000 7.04000 2.97000 5.29000
## mean           0.21273 1.27285 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean        0.00197 0.00115 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95   0.00387 0.00226 0.00899 0.00867 0.01732 0.00659 0.00971
## var            0.01362 0.00465 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev        0.11673 0.06822 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var       0.54870 0.05359 0.07933 0.05300 0.07452 0.06715 0.05701
##               ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```r
# price: mean 2.673, sd 0.322
# wholesale price: mean 1.847, sd 0.155
# market share: mean 0.051, sd 0.029

ba50 <- filter(data2, brand_name == "Bayer", size == "50")
stat.desc(ba50, basic = F)
```

```
##                      X1    store    week brand_name size  sales
## median         26280.5000   78.000  24.500         NA   50 3.0000
```

```
## mean           26280.5000   74.767  24.500         NA   50 3.5865
## SE.mean           17.0904    0.563   0.234         NA    0 0.0473
## CI.mean.0.95      33.5082    1.105   0.459         NA    0 0.0927
## var          1023460.0000 1112.195 191.971         NA    0 7.8332
## std.dev         1011.6620   33.350  13.855         NA    0 2.7988
## coef.var           0.0385    0.446   0.566         NA    0 0.7804
##                     count   price   cost sales_total mkt_share exp_total
## median          18492.000 3.49000 2.3600      80.000  0.035149    53.320
## mean            19609.354 3.60720 2.4221      86.417  0.041477    57.046
## SE.mean            89.894 0.00636 0.0059       0.530  0.000462     0.422
## CI.mean.0.95      176.249 0.01247 0.0116       1.039  0.000906     0.828
## var          28315331.915 0.14177 0.1220     983.457  0.000748   625.228
## std.dev          5321.215 0.37652 0.3492      31.360  0.027356    25.005
## coef.var            0.271 0.10438 0.1442       0.363  0.659533     0.438
##               exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median          0.20056 1.27204 3.50000 4.94000 7.04000 2.97000 5.29000
## mean            0.22537 1.27285 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean         0.00220 0.00115 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95    0.00432 0.00226 0.00899 0.00867 0.01732 0.00659 0.00971
## var             0.01699 0.00465 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev         0.13033 0.06822 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var        0.57829 0.05359 0.07933 0.05300 0.07452 0.06715 0.05701
##                ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median       8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean         8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean      0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var          0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev      0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var     0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median       2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean         2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean      0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var          0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev      0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var     0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```r
# price: mean 3.607, sd 0.377
# wholesale price: mean 2.422, sd 0.349
# market share: mean 0.041, sd 0.027

ba100 <- filter(data2, brand_name == "Bayer", size == "100")
stat.desc(ba100, basic = F)
```

```
##                     X1    store    week brand_name size   sales
## median         29784.500   78.000  24.500         NA  100  8.0000
## mean           29784.500   74.767  24.500         NA  100  8.3356
## SE.mean           17.090    0.563   0.234         NA    0  0.0827
## CI.mean.0.95      33.508    1.105   0.459         NA    0  0.1622
## var          1023460.000 1112.195 191.971         NA    0 23.9713
## std.dev         1011.662   33.350  13.855         NA    0  4.8960
## coef.var           0.034    0.446   0.566         NA    0  0.5874
##                     count   price    cost sales_total mkt_share exp_total
```

```
## median          18492.000 3.97000 3.71000      80.000  0.092105    53.320
## mean            19609.354 3.96664 3.71152      86.417  0.097254    57.046
## SE.mean            89.894 0.00312 0.00197       0.530  0.000766     0.422
## CI.mean.0.95      176.249 0.00612 0.00387       1.039  0.001502     0.828
## var          28315331.915 0.03412 0.01366     983.457  0.002057   625.228
## std.dev          5321.215 0.18472 0.11687      31.360  0.045354    25.005
## coef.var            0.271 0.04657 0.03149       0.363  0.466348     0.438
##               exp_share   stone   ty25p   ty50p  ty100p   ad25p   ad50p
## median          0.57039 1.27204 3.50000 4.94000 7.04000 2.97000 5.29000
## mean            0.56189 1.27285 3.42047 4.94202 7.01607 2.96365 5.14502
## SE.mean         0.00261 0.00115 0.00458 0.00442 0.00883 0.00336 0.00496
## CI.mean.0.95    0.00512 0.00226 0.00899 0.00867 0.01732 0.00659 0.00971
## var             0.02389 0.00465 0.07363 0.06859 0.27333 0.03961 0.08603
## std.dev         0.15455 0.06822 0.27136 0.26191 0.52281 0.19901 0.29331
## coef.var        0.27505 0.05359 0.07933 0.05300 0.07452 0.06715 0.05701
##                ad100p   ba25p   ba50p  ba100p   ty25c   ty50c  ty100c
## median        8.29000 2.62000 3.49000 3.97000 2.23000 3.73000 5.71000
## mean          8.15974 2.67305 3.60720 3.96664 2.18227 3.67193 5.75482
## SE.mean       0.00581 0.00544 0.00636 0.00312 0.00303 0.00308 0.00517
## CI.mean.0.95 0.01140 0.01066 0.01247 0.00612 0.00594 0.00604 0.01013
## var           0.11843 0.10364 0.14177 0.03412 0.03215 0.03321 0.09356
## std.dev       0.34414 0.32193 0.37652 0.18472 0.17930 0.18224 0.30587
## coef.var      0.04217 0.12044 0.10438 0.04657 0.08216 0.04963 0.05315
##                 ad25c   ad50c  ad100c   ba25c  ba50c  ba100c
## median        2.02000 3.63000 6.09000 1.84000 2.3600 3.71000
## mean          2.03000 3.62258 6.09102 1.84650 2.4221 3.71152
## SE.mean       0.00120 0.00239 0.00381 0.00262 0.0059 0.00197
## CI.mean.0.95 0.00235 0.00469 0.00747 0.00514 0.0116 0.00387
## var           0.00505 0.02006 0.05092 0.02407 0.1220 0.01366
## std.dev       0.07105 0.14164 0.22566 0.15513 0.3492 0.11687
## coef.var      0.03500 0.03910 0.03705 0.08402 0.1442 0.03149
```

```
# price: mean 3.967, sd 0.185
# wholesale price: mean 3.712, sd 0.117
# market share: mean 0.097, sd 0.045
```

```
options(scipen = 1)
fit1 <- lm(log(sales) ~ log(price), data2)
summary(fit1)
```

```
##
## Call:
## lm(formula = log(sales) ~ log(price), data = data2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.004 -0.559  0.134  0.678  2.715
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1144     0.0212    99.9   <2e-16 ***
## log(price)   -0.1406     0.0140   -10.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.914 on 31534 degrees of freedom
## Multiple R-squared:  0.0032, Adjusted R-squared:  0.00316
## F-statistic:  101 on 1 and 31534 DF,  p-value: <2e-16
```

```
# price elasticity of demand is -0.1406
```

```
# create 9 new dummy variables
data2 <- data2 %>% mutate(ty25 = ifelse(brand_name == "Tylenol" & size == 25,1,0 ),
                          ty50 = ifelse(brand_name == "Tylenol" & size == 50,1,0 ),
                          ty100 = ifelse(brand_name == "Tylenol" & size == 100,1,0),
                          ad25 = ifelse(brand_name == "Advil" & size == 25,1,0),
                          ad50 = ifelse(brand_name == "Advil" & size == 50,1,0),
                          ad100 = ifelse(brand_name == "Advil" & size == 100,1,0),
                          ba25 = ifelse(brand_name == "Bayer" & size == 25,1,0),
                          ba50 = ifelse(brand_name == "Bayer" & size == 50,1,0),
                          ba100 = ifelse(brand_name == "Bayer" & size == 100,1,0))
```

```
# 27 interaction variables between log(price) and product dummies
n = 42
for (i in seq(33,35)){
  for (j in  seq(15,17)){
  data2[,n] = data2[,i]*log(data2[,j])
  n <- n + 1
  }
}

for (i in seq(36,38)){
  for (j in  seq(18,20)){
  data2[,n] = data2[,i]*log(data2[,j])
  n <- n + 1
  }
}

for (i in seq(39,41)){
  for (j in  seq(21,23)){
  data2[,n] = data2[,i]*log(data2[,j])
  n <- n + 1
  }
}

# 27 interaction variables between cost and product dummies for the IV model
for (i in seq(33,35)){
  for (j in  seq(24,26)){
  data2[,n] = data2[,i]*data2[,j]
  n <- n + 1
  }
}

for (i in seq(36,38)){
  for (j in  seq(27,29)){
  data2[,n] = data2[,i]*data2[,j]
  n <- n + 1
  }
}
```

```
for (i in seq(39,41)){
  for (j in  seq(30,32)){
  data2[,n] = data2[,i]*data2[,j]
  n <- n + 1
  }
}
```

Lowest level stone price index

Estimate the lowest level of the model using the Stone price index and without assuming symmetry of the γ.

```
data2 <- data2 %>% mutate(lxoverp = log(exp_total)-stone)

options(digits = 3)
n = 97
for (i in seq(33,41)){
  data2[,n] = data2[,i]*data2[,96]
  n = n + 1
}

# create a new data frame and keep only the columns in the regression.
data3 <- data2
data3[,c(1:12,14:32,69:96)]<- NULL

fit2 <- lm(exp_share ~ ., data3)
summary(fit2)
```

```
##
## Call:
## lm(formula = exp_share ~ ., data = data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4770 -0.0809 -0.0064  0.0730  0.6322
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.67886    0.06856   24.49  < 2e-16 ***
## ty25        -1.62677    0.11037  -14.74  < 2e-16 ***
## ty50        -1.19531    0.11037  -10.83  < 2e-16 ***
## ty100       -1.21449    0.11037  -11.00  < 2e-16 ***
## ad25        -2.17628    0.12952  -16.80  < 2e-16 ***
## ad50        -1.92859    0.12952  -14.89  < 2e-16 ***
## ad100        0.06830    0.12952    0.53  0.59796
## ba25        -1.46858    0.09696  -15.15  < 2e-16 ***
## ba50        -2.56799    0.09696  -26.48  < 2e-16 ***
## ba100             NA         NA      NA       NA
## ty25.1       0.07383    0.01800    4.10  4.1e-05 ***
## ty25.2       0.14377    0.03885    3.70  0.00022 ***
## ty25.3      -0.02171    0.02773   -0.78  0.43364
## ty50.1       0.00653    0.01800    0.36  0.71670
## ty50.2      -0.27470    0.03885   -7.07  1.6e-12 ***
## ty50.3       0.13755    0.02773    4.96  7.1e-07 ***
## ty100.1     -0.08037    0.01800   -4.47  8.0e-06 ***
```

```
## ty100.2      0.13093    0.03885    3.37  0.00075 ***
## ty100.3     -0.11584    0.02773   -4.18  3.0e-05 ***
## ad25.1      -0.32737    0.03257  -10.05  < 2e-16 ***
## ad25.2       0.67386    0.03467   19.44  < 2e-16 ***
## ad25.3       0.07789    0.04895    1.59  0.11154
## ad50.1       0.05503    0.03257    1.69  0.09112 .
## ad50.2      -0.86294    0.03467  -24.89  < 2e-16 ***
## ad50.3       0.88812    0.04895   18.14  < 2e-16 ***
## ad100.1      0.27234    0.03257    8.36  < 2e-16 ***
## ad100.2      0.18907    0.03467    5.45  5.0e-08 ***
## ad100.3     -0.96601    0.04895  -19.74  < 2e-16 ***
## ba25.1       0.16797    0.01777    9.45  < 2e-16 ***
## ba25.2      -0.16499    0.02412   -6.84  8.0e-12 ***
## ba25.3       0.14388    0.05159    2.79  0.00529 **
## ba50.1      -0.09016    0.01777   -5.07  3.9e-07 ***
## ba50.2       0.13653    0.02412    5.66  1.5e-08 ***
## ba50.3       0.73568    0.05159   14.26  < 2e-16 ***
## ba100.1     -0.07780    0.01777   -4.38  1.2e-05 ***
## ba100.2      0.02846    0.02412    1.18  0.23808
## ba100.3     -0.87956    0.05159  -17.05  < 2e-16 ***
## ty25.7      -0.02571    0.00500   -5.14  2.7e-07 ***
## ty50.7       0.02114    0.00500    4.23  2.4e-05 ***
## ty100.7      0.00458    0.00500    0.92  0.36010
## ad25.7      -0.01771    0.00456   -3.89  0.00010 ***
## ad50.7       0.03635    0.00456    7.98  1.5e-15 ***
## ad100.7     -0.01864    0.00456   -4.09  4.3e-05 ***
## ba25.7      -0.05567    0.00456  -12.22  < 2e-16 ***
## ba50.7       0.00567    0.00456    1.25  0.21310
## ba100.7      0.05000    0.00456   10.97  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.12 on 31491 degrees of freedom
## Multiple R-squared:  0.471,  Adjusted R-squared:  0.47
## F-statistic:  637 on 44 and 31491 DF,  p-value: <2e-16
```

Lowest level IV: wholesale price

Estimate the lowest level of the model using IV, with cost as the instrument for price.

```
data4 <- data2
data4[,c(1:12,14:32,96)]<- NULL
fit3 <- ivreg(exp_share ~ ty25 + ty50 + ty100 + ad25 + ad50 + ad100 + ba25 + ba50 + ba100 + ty25.1 + ty
summary(fit3)
```

```
##
## Call:
## ivreg(formula = exp_share ~ ty25 + ty50 + ty100 + ad25 + ad50 +
##     ad100 + ba25 + ba50 + ba100 + ty25.1 + ty25.2 + ty25.3 +
##     ty50.1 + ty50.2 + ty50.3 + ty100.1 + ty100.2 + ty100.3 +
##     ad25.1 + ad25.2 + ad25.3 + ad50.1 + ad50.2 + ad50.3 + ad100.1 +
##     ad100.2 + ad100.3 + ba25.1 + ba25.2 + ba25.3 + ba50.1 + ba50.2 +
##     ba50.3 + ba100.1 + ba100.2 + ba100.3 + ty25.7 + ty50.7 +
##     ty100.7 + ad25.7 + ad50.7 + ad100.7 + ba25.7 + ba50.7 + ba100.7 |
##     ty25 + ty50 + ty100 + ad25 + ad50 + ad100 + ba25 + ba50 +
```

```
##         ba100 + ty25.4 + ty25.5 + ty25.6 + ty50.4 + ty50.5 +
##         ty50.6 + ty100.4 + ty100.5 + ty100.6 + ad25.4 + ad25.5 +
##         ad25.6 + ad50.4 + ad50.5 + ad50.6 + ad100.4 + ad100.5 +
##         ad100.6 + ba25.4 + ba25.5 + ba25.6 + ba50.4 + ba50.5 +
##         ba50.6 + ba100.4 + ba100.5 + ba100.6 + ty25.7 + ty50.7 +
##         ty100.7 + ad25.7 + ad50.7 + ad100.7 + ba25.7 + ba50.7 +
##         ba100.7, data = data4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94790 -0.08986 -0.00485  0.08390  0.70879
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.23618    0.23258   -1.02  0.30988
## ty25        -0.27861    0.31863   -0.87  0.38191
## ty50         0.33871    0.31863    1.06  0.28778
## ty100        1.64843    0.31863    5.17  2.3e-07 ***
## ad25         3.99569    1.09753    3.64  0.00027 ***
## ad50         2.88602    1.09753    2.63  0.00855 **
## ad100       -5.17318    1.09753   -4.71  2.4e-06 ***
## ba25         0.67168    0.32891    2.04  0.04115 *
## ba50         1.03685    0.32891    3.15  0.00162 **
## ty25.1       0.08170    0.02316    3.53  0.00042 ***
## ty25.2       0.62417    0.14220    4.39  1.1e-05 ***
## ty25.3      -0.15305    0.08403   -1.82  0.06854 .
## ty50.1       0.00333    0.02316    0.14  0.88559
## ty50.2      -0.16377    0.14220   -1.15  0.24947
## ty50.3       0.23234    0.08403    2.77  0.00569 **
## ty100.1     -0.08503    0.02316   -3.67  0.00024 ***
## ty100.2     -0.46039    0.14220   -3.24  0.00121 **
## ty100.3     -0.07929    0.08403   -0.94  0.34535
## ad25.1      -0.54849    0.17349   -3.16  0.00157 **
## ad25.2       0.91988    0.24216    3.80  0.00015 ***
## ad25.3      -1.99615    0.51583   -3.87  0.00011 ***
## ad50.1       0.82731    0.17349    4.77  1.9e-06 ***
## ad50.2      -2.38334    0.24216   -9.84  < 2e-16 ***
## ad50.3       0.34861    0.51583    0.68  0.49916
## ad100.1     -0.27882    0.17349   -1.61  0.10803
## ad100.2      1.46346    0.24216    6.04  1.5e-09 ***
## ad100.3      1.64754    0.51583    3.19  0.00140 **
## ba25.1       0.11779    0.02343    5.03  5.0e-07 ***
## ba25.2      -0.35674    0.05664   -6.30  3.1e-10 ***
## ba25.3       0.20549    0.20110    1.02  0.30688
## ba50.1      -0.06499    0.02343   -2.77  0.00554 **
## ba50.2       0.92472    0.05664   16.33  < 2e-16 ***
## ba50.3      -1.24579    0.20110   -6.19  5.9e-10 ***
## ba100.1     -0.05280    0.02343   -2.25  0.02422 *
## ba100.2     -0.56798    0.05664  -10.03  < 2e-16 ***
## ba100.3      1.04030    0.20110    5.17  2.3e-07 ***
## ty25.7      -0.01335    0.00700   -1.91  0.05668 .
## ty50.7       0.02746    0.00700    3.92  8.9e-05 ***
## ty100.7     -0.01411    0.00700   -2.01  0.04403 *
## ad25.7      -0.04024    0.00886   -4.54  5.6e-06 ***
```

```
## ad50.7      -0.00252    0.00886   -0.28  0.77623
## ad100.7      0.04276    0.00886    4.83  1.4e-06 ***
## ba25.7      -0.06166    0.00554  -11.14  < 2e-16 ***
## ba50.7       0.00814    0.00554    1.47  0.14177
## ba100.7      0.05353    0.00554    9.67  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.142 on 31491 degrees of freedom
## Multiple R-Squared: 0.257,   Adjusted R-squared: 0.256
## Wald test:  430 on 44 and 31491 DF,  p-value: <2e-16
```

Lowest level IV: Hausman instruments

Estimate the lowest level of the model using IV, using Hausman instruments for price.

```
data5 <- data2
data5 = data5[FALSE,]
for (i in unique(data2$week)){
  for (j in unique(data2$brand_name)){
    for (k in unique(data2$size)){
    test <- filter(data2, brand_name == j, week == i, size == k)
    test <- test %>% mutate(avgp = (sum(test$price)-price)/(nrow(test)-1))
    data5 <- rbind(data5, test)
    }}}
data6 <- data5
data6 = data6[FALSE,]
for (i in unique(data5$week)){
  for (j in unique(data$store)){
    test <- filter(data5, week == i, store == j)
    test <- test %>% mutate(ty25h = avgp[1],ty50h = avgp[2], ty100h = avgp[3], ad25h = avgp[4], ad50h =
    data6 <- rbind(data6, test)
  }
}

data6[,c(1:12,14:32,69:96,106)]<- NULL
# 27 interaction variables between hausman instrument and product dummies
n = 56
for (i in seq(2,4)){
  for (j in  seq(47,49)){
  data6[,n] = data6[,i]*data6[,j]
  n <- n + 1
  }
}

for (i in seq(5,7)){
  for (j in  seq(50,52)){
  data6[,n] = data6[,i]*data6[,j]
  n <- n + 1
  }
}

for (i in seq(8,10)){
  for (j in  seq(53,55)){
  data6[,n] = data6[,i]*data6[,j]
```

```
  n <- n + 1
  }
}


fit4 <- ivreg(exp_share ~ ty25 + ty50 + ty100 + ad25 + ad50 + ad100 + ba25 + ba50 + ba100 + ty25.1 + ty
summary(fit4)
```

```
##
## Call:
## ivreg(formula = exp_share ~ ty25 + ty50 + ty100 + ad25 + ad50 +
##     ad100 + ba25 + ba50 + ba100 + ty25.1 + ty25.2 + ty25.3 +
##     ty50.1 + ty50.2 + ty50.3 + ty100.1 + ty100.2 + ty100.3 +
##     ad25.1 + ad25.2 + ad25.3 + ad50.1 + ad50.2 + ad50.3 + ad100.1 +
##     ad100.2 + ad100.3 + ba25.1 + ba25.2 + ba25.3 + ba50.1 + ba50.2 +
##     ba50.3 + ba100.1 + ba100.2 + ba100.3 + ty25.7 + ty50.7 +
##     ty100.7 + ad25.7 + ad50.7 + ad100.7 + ba25.7 + ba50.7 + ba100.7 |
##     ty25 + ty50 + ty100 + ad25 + ad50 + ad100 + ba25 + ba50 +
##         ba100 + ty25.4 + ty25.5 + ty25.6 + ty50.4 + ty50.5 +
##         ty50.6 + ty100.4 + ty100.5 + ty100.6 + ad25.4 + ad25.5 +
##         ad25.6 + ad50.4 + ad50.5 + ad50.6 + ad100.4 + ad100.5 +
##         ad100.6 + ba25.4 + ba25.5 + ba25.6 + ba50.4 + ba50.5 +
##         ba50.6 + ba100.4 + ba100.5 + ba100.6 + ty25.7 + ty50.7 +
##         ty100.7 + ad25.7 + ad50.7 + ad100.7 + ba25.7 + ba50.7 +
##         ba100.7, data = data6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52236 -0.08143 -0.00619  0.07388  0.70832
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.15586    0.12366   17.43  < 2e-16 ***
## ty25        -2.49339    0.17809  -14.00  < 2e-16 ***
## ty50        -1.87337    0.17809  -10.52  < 2e-16 ***
## ty100       -1.10080    0.17809   -6.18  6.4e-10 ***
## ad25        -2.63086    0.17023  -15.45  < 2e-16 ***
## ad50        -2.57337    0.17023  -15.12  < 2e-16 ***
## ad100       -0.26334    0.17023   -1.55  0.12188
## ba25        -1.86103    0.17489  -10.64  < 2e-16 ***
## ba50        -3.60654    0.17489  -20.62  < 2e-16 ***
## ty25.1      -0.05071    0.02919   -1.74  0.08230 .
## ty25.2       0.18756    0.05026    3.73  0.00019 ***
## ty25.3       0.20284    0.04667    4.35  1.4e-05 ***
## ty50.1       0.06822    0.02919    2.34  0.01942 *
## ty50.2      -0.43208    0.05026   -8.60  < 2e-16 ***
## ty50.3       0.33304    0.04667    7.14  9.8e-13 ***
## ty100.1     -0.01751    0.02919   -0.60  0.54861
## ty100.2      0.24453    0.05026    4.87  1.1e-06 ***
## ty100.3     -0.53588    0.04667  -11.48  < 2e-16 ***
## ad25.1      -0.41976    0.04879   -8.60  < 2e-16 ***
## ad25.2       0.66277    0.03659   18.12  < 2e-16 ***
## ad25.3       0.12627    0.05493    2.30  0.02151 *
## ad50.1       0.13648    0.04879    2.80  0.00516 **
```

```
## ad50.2      -0.86991    0.03659  -23.78  < 2e-16 ***
## ad50.3       0.92840    0.05493   16.90  < 2e-16 ***
## ad100.1      0.28328    0.04879    5.81  6.5e-09 ***
## ad100.2      0.20715    0.03659    5.66  1.5e-08 ***
## ad100.3     -1.05468    0.05493  -19.20  < 2e-16 ***
## ba25.1       0.24841    0.06356    3.91  9.3e-05 ***
## ba25.2      -0.40734    0.03964  -10.28  < 2e-16 ***
## ba25.3       0.26561    0.07125    3.73  0.00019 ***
## ba50.1      -0.13483    0.06356   -2.12  0.03390 *
## ba50.2       0.53246    0.03964   13.43  < 2e-16 ***
## ba50.3       0.77755    0.07125   10.91  < 2e-16 ***
## ba100.1     -0.11358    0.06356   -1.79  0.07394 .
## ba100.2     -0.12512    0.03964   -3.16  0.00160 **
## ba100.3     -1.04316    0.07125  -14.64  < 2e-16 ***
## ty25.7      -0.01617    0.00536   -3.02  0.00256 **
## ty50.7       0.02018    0.00536    3.76  0.00017 ***
## ty100.7     -0.00401    0.00536   -0.75  0.45470
## ad25.7      -0.01957    0.00466   -4.20  2.7e-05 ***
## ad50.7       0.03847    0.00466    8.25  < 2e-16 ***
## ad100.7     -0.01890    0.00466   -4.05  5.0e-05 ***
## ba25.7      -0.06352    0.00472  -13.47  < 2e-16 ***
## ba50.7       0.02122    0.00472    4.50  6.8e-06 ***
## ba100.7      0.04230    0.00472    8.97  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.122 on 31491 degrees of freedom
## Multiple R-Squared: 0.453,   Adjusted R-squared: 0.452
## Wald test:  629 on 44 and 31491 DF,  p-value: <2e-16
```

Middle level IV

```
# start with a new set of data
data7 <- data2
data7= data7[FALSE,]

for (i in unique(data2$week)){
  for (j in unique(data2$store)){
    test <- filter(data2, week == i, store == j)

    total1 = test$sales[1] + test$sales[2] + test$sales[3]
    total2 =test$sales[4] + test$sales[5] + test$sales[6]
    total3 = test$sales[7] + test$sales[8] + test$sales[9]

    test <- test[-c(2,3,5, 6,8,9),]

    test[,c(15:105)]<- NULL

    test <- test %>% mutate(totalq = ifelse(brand_name == "Tylenol", total1, ifelse(brand_name == "Advi

    test$totalq = as.numeric(as.character(test$totalq))

    test <- test %>% mutate(ty = ifelse(brand_name == "Tylenol", 1, 0),
                            ad = ifelse(brand_name == "Advil",1,0),
```

```
                            ba = ifelse(brand_name == "Bayer",1,0)) %>% mutate(exp_total2 = (sum(exp_to
                            stone_ad = test$stone[2],
                            stone_ba = test$stone[3])


    data7<- rbind(data7,test)

  }}

n = 23
for (i in seq(16,18)){
  for (j in  seq(20,22)){
  data7[,n] = data7[,i]*data7[,j]
  n <- n + 1
  }
}
for (i in seq(16,18)){
  data7[,n] = data7[,i]*log(data7[,19])
  n <- n + 1
}

fit5 <- lm(log(totalq) ~ ty + ad + ba + ty.1 + ty.2 + ty.3 + ad.1 + ad.2 + ad.3 + ba.1 + ba.2 + ba.3 +
summary(fit5)
```

```
##
## Call:
## lm(formula = log(totalq) ~ ty + ad + ba + ty.1 + ty.2 + ty.3 +
##     ad.1 + ad.2 + ad.3 + ba.1 + ba.2 + ba.3 + ty.4 + ad.4 + ba.4,
##     data = data7)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5973 -0.1292  0.0177  0.1440  1.0506
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.5709     0.1361  -11.54  < 2e-16 ***
## ty            1.2331     0.1925    6.41  1.6e-10 ***
## ad           -0.4736     0.1925   -2.46    0.014 *
## ba               NA         NA      NA       NA
## ty.1         -0.7968     0.0624  -12.77  < 2e-16 ***
## ty.2         -0.2054     0.0365   -5.62  1.9e-08 ***
## ty.3         -0.3724     0.0638   -5.84  5.4e-09 ***
## ad.1         -0.1280     0.0624   -2.05    0.040 *
## ad.2         -0.4149     0.0365  -11.36  < 2e-16 ***
## ad.3          0.0661     0.0638    1.04    0.300
## ba.1         -0.5516     0.0624   -8.84  < 2e-16 ***
## ba.2         -0.1888     0.0365   -5.17  2.4e-07 ***
## ba.3          0.4242     0.0638    6.65  3.0e-11 ***
## ty.4          1.0475     0.0113   93.06  < 2e-16 ***
## ad.4          0.9994     0.0113   88.79  < 2e-16 ***
## ba.4          0.8324     0.0113   73.95  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.251 on 10497 degrees of freedom
## Multiple R-squared:  0.838,  Adjusted R-squared:  0.838
## F-statistic: 3.89e+03 on 14 and 10497 DF,  p-value: <2e-16
```

Top level

```
data8 <- data7
data8= data8[FALSE,]
income <- read_csv("OTC_Incomes.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   store = col_double(),
##   week = col_double(),
##   average_income = col_double()
## )
```

```
for (i in unique(data7$week)){
  for (j in unique(data7$store)){
    test <- filter(data7, week == i, store == j)
    income2 <- filter(income,week == i, store == j)

    test <- test %>% mutate(exp_share2 = exp_total/exp_total2)
    test <- test %>% mutate(overallstone = sum(test$exp_share2*test$stone))
    test <- test %>% mutate(income = income2$average_income[1])

    test <- test[-c(2,3),]

    data8<- rbind(data8,test)

  }}
data8[,c(4:9,11:14,16:18,20:34)]<- NULL


data8 <- cbind(data8,income)

fit6 <- lm(log(sales_total)~overallstone+income, data8)
summary(fit6)
```

```
##
## Call:
## lm(formula = log(sales_total) ~ overallstone + income, data = data8)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.186 -0.246 -0.012  0.257  1.109
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.77e+00   1.56e-01   30.52   <2e-16 ***
## overallstone -2.53e-01   1.02e-01   -2.47    0.014 *
```

```
## income        4.64e-07   4.24e-07    1.09    0.274
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.374 on 3501 degrees of freedom
## Multiple R-squared:  0.00179,    Adjusted R-squared:  0.00122
## F-statistic: 3.14 on 2 and 3501 DF,  p-value: 0.0433
```

Compare across the models

The model using the Hausman instrument is the most valid one. This is determined by looking at the sign of the estimated coefficients for the nine different own price elasticities.

For the original OLS model, own price elasticities: 6 out of 9 are negative. For the IV model with wholesale price as instrument, 4 out of 9 are negative. For the IV model with Hausman instrument, 7 out of 9 are negative. Therefore, the third model is the preferred specification. Its estimated coefficients will also be used to calculate the following conditional and unconditional elasticities.

```
# average expenditure share for 9 products
omega_i <- c(mean(ty25$exp_share), mean(ty50$exp_share),mean(ty100$exp_share),mean(ad25$exp_share),mean

data9 <- data7
data9 <- data9 %>% mutate(exp_share2 = exp_total/exp_total2)
ty1 <- filter(data9, brand_name == "Tylenol")
omega_ty = mean(ty1$exp_share2)
ad1 <- filter(data9, brand_name == "Advil")
omega_ad = mean(ad1$exp_share2)
ba1 <- filter(data9, brand_name == "Bayer")
omega_ba = mean(ba1$exp_share2)

# average expenditrue share for 3 segments
omega_g <- c(omega_ty,omega_ad,omega_ba)

coef <- fit4$coefficients
coef <- as.data.frame(coef)
beta_i <- coef$coef[38:46]
gamma_ij <- coef$coef[11:37]
```

conditional elasticities:

Conditional Tyelenol

```
## Tylenol25 with 25, 50, 100
  ty25_25 = (1/omega_i[1])*(gamma_ij[1]+gamma_ij[1]-beta_i[1]*omega_i[1])-1
  ty25_50 = (1/omega_i[1])*(gamma_ij[2]+gamma_ij[4]-beta_i[1]*omega_i[2])
  ty25_100 = (1/omega_i[1])*(gamma_ij[3]+gamma_ij[7]-beta_i[1]*omega_i[3])
## Tylenol50 with 25, 50, 100
  ty50_25 = (1/omega_i[2])*(gamma_ij[2]+gamma_ij[4]-beta_i[2]*omega_i[1])
  ty50_50 = (1/omega_i[2])*(gamma_ij[5]+gamma_ij[5]-beta_i[2]*omega_i[2])-1
  ty50_100 = (1/omega_i[2])*(gamma_ij[6]+gamma_ij[8]-beta_i[2]*omega_i[3])
## Tylenol100 with 25, 50, 100
  ty100_25 = (1/omega_i[3])*(gamma_ij[3]+gamma_ij[7]-beta_i[3]*omega_i[1])
  ty100_50 = (1/omega_i[3])*(gamma_ij[6]+gamma_ij[8]-beta_i[3]*omega_i[2])
  ty100_100 = (1/omega_i[3])*(gamma_ij[9]+gamma_ij[9]-beta_i[3]*omega_i[3])-1
con_ty <- c(ty25_25,ty25_50,ty25_100,ty50_25,ty50_50,ty50_100,ty100_25,ty100_50,ty100_100)
```

```
dim(con_ty) <- c(3,3)
con_ty
```

```
##         [,1]   [,2]   [,3]
## [1,] -1.417  0.629  0.508
## [2,]  1.119 -3.186  1.580
## [3,]  0.816  1.429 -3.920
```

Conditional Advil

```
## Advil25 with 25, 50, 100
  ad25_25 = (1/omega_i[4])*(gamma_ij[10]+gamma_ij[10]-beta_i[4]*omega_i[4])-1
  ad25_50 = (1/omega_i[4])*(gamma_ij[2+9]+gamma_ij[4+9]-beta_i[4]*omega_i[5])
  ad25_100 = (1/omega_i[4])*(gamma_ij[12]+gamma_ij[16]-beta_i[4]*omega_i[6])
## Advil50 with 25, 50, 100
  ad50_25 = (1/omega_i[5])*(gamma_ij[11]+gamma_ij[13]-beta_i[5]*omega_i[4])
  ad50_50 = (1/omega_i[5])*(gamma_ij[14]+gamma_ij[14]-beta_i[5]*omega_i[5])-1
  ad50_100 = (1/omega_i[5])*(gamma_ij[15]+gamma_ij[17]-beta_i[5]*omega_i[6])
## Advil100 with 25, 50, 100
  ad100_25 = (1/omega_i[6])*(gamma_ij[12]+gamma_ij[16]-beta_i[6]*omega_i[4])
  ad100_50 = (1/omega_i[6])*(gamma_ij[15]+gamma_ij[17]-beta_i[6]*omega_i[5])
  ad100_100 = (1/omega_i[6])*(gamma_ij[18]+gamma_ij[18]-beta_i[6]*omega_i[6])-1
con_ad <- c(ad25_25,ad25_50,ad25_100,ad50_25,ad50_50,ad50_100,ad100_25,ad100_50,ad100_100)
dim(con_ad) <- c(3,3)
con_ad
```

```
##       [,1]  [,2]  [,3]
## [1,] -3.31  2.12  1.55
## [2,]  2.24 -5.73  4.25
## [3,]  1.15  3.03 -8.83
```

Conditional Bayer

```
## Bayer25 with 25, 50, 100
  ba25_25 = (1/omega_i[7])*(gamma_ij[19]+gamma_ij[19]-beta_i[7]*omega_i[7])-1
  ba25_50 = (1/omega_i[7])*(gamma_ij[20]+gamma_ij[22]-beta_i[7]*omega_i[8])
  ba25_100 = (1/omega_i[7])*(gamma_ij[21]+gamma_ij[25]-beta_i[7]*omega_i[9])
## Bayer50 with 25, 50, 100
  ba50_25 = (1/omega_i[8])*(gamma_ij[20]+gamma_ij[22]-beta_i[8]*omega_i[7])
  ba50_50 = (1/omega_i[8])*(gamma_ij[23]+gamma_ij[23]-beta_i[8]*omega_i[8])-1
  ba50_100 = (1/omega_i[8])*(gamma_ij[24]+gamma_ij[26]-beta_i[8]*omega_i[9])
## Bayer100 with 25, 50, 100
  ba100_25 = (1/omega_i[9])*(gamma_ij[21]+gamma_ij[25]-beta_i[9]*omega_i[7])
  ba100_50 = (1/omega_i[9])*(gamma_ij[14]+gamma_ij[26]-beta_i[9]*omega_i[8])
  ba100_100 = (1/omega_i[9])*(gamma_ij[27]+gamma_ij[27]-beta_i[9]*omega_i[9])-1
con_ba <- c(ba25_25,ba25_50,ba25_100,ba50_25,ba50_50,ba50_100,ba100_25,ba100_50,ba100_100)
dim(con_ba) <- c(3,3)
con_ba
```

```
##         [,1]  [,2]   [,3]
## [1,]  1.399 -2.43  0.255
## [2,] -2.481  3.70 -1.788
## [3,]  0.882  2.84 -4.755
```

Unconditional elasticities:

Unconditional Tyelenol

```
delta_1 = fit6$coefficients[2]
beta_g = fit5$coefficients[14:16]

## Unconditional Tylenol25 with 25, 50, 100
  ty25_25_Xg_Pj = omega_i[1]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[1])*(1+delta_1)
  ty25_25_un = (1/omega_i[1])*(gamma_ij[1]+gamma_ij[1])+beta_i[1]*(ty25_25_Xg_Pj-omega_i[1])-1+ty25_25_X
  ty25_50_Xg_Pj = omega_i[2]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[2])*(1+delta_1)
  ty25_50_un = (1/omega_i[1])*(gamma_ij[2]+gamma_ij[4])+beta_i[1]*(ty25_50_Xg_Pj-omega_i[2])+ty25_50_Xg
  ty25_100_Xg_Pj = omega_i[3]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[3])*(1+delta_1)
  ty25_100_un = (1/omega_i[1])*(gamma_ij[3]+gamma_ij[7])+beta_i[1]*(ty25_100_Xg_Pj-omega_i[3])+ty25_100


## Unconditional Tylenol50 with 25, 50, 100
  ty50_25_Xg_Pj = omega_i[1]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[1])*(1+delta_1)
  ty50_25_un = (1/omega_i[2])*(gamma_ij[2]+gamma_ij[4])+beta_i[2]*(ty50_25_Xg_Pj-omega_i[1])+ty50_25_Xg
  ty50_50_Xg_Pj = omega_i[2]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[2])*(1+delta_1)
  ty50_50_un = (1/omega_i[2])*(gamma_ij[5]+gamma_ij[5])+beta_i[2]*(ty50_50_Xg_Pj-omega_i[2])+ty50_50_Xg
  ty50_100_Xg_Pj = omega_i[3]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[3])*(1+delta_1)
  ty50_100_un = (1/omega_i[2])*(gamma_ij[6]+gamma_ij[8])+beta_i[2]*(ty50_100_Xg_Pj-omega_i[3])+ty50_100

## Unconditional Tylenol100 with 25, 50, 100
  ty100_25_Xg_Pj = omega_i[1]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[1])*(1+delta_1)
  ty100_25_un = (1/omega_i[3])*(gamma_ij[3]+gamma_ij[7])+beta_i[3]*(ty100_25_Xg_Pj-omega_i[1])+ty100_25
  ty100_50_Xg_Pj = omega_i[2]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[2])*(1+delta_1)
  ty100_50_un = (1/omega_i[3])*(gamma_ij[6]+gamma_ij[8])+beta_i[3]*(ty100_50_Xg_Pj-omega_i[2])+ty100_50
  ty100_100_Xg_Pj = omega_i[3]*(1+omega_g[1])+beta_g[1]*(omega_g[1]*omega_i[3])*(1+delta_1)
  ty100_100_un = (1/omega_i[3])*(gamma_ij[9]+gamma_ij[9])+beta_i[3]*(ty100_100_Xg_Pj-omega_i[3])+ty100_1
uncon_ty <- c(ty25_25_un,ty25_50_un,ty25_100_un,ty50_25_un,ty50_50_un,ty50_100_un,ty100_25_un,ty100_50_u
dim(uncon_ty) <- c(3,3)
uncon_ty
```

```
##         [,1]  [,2]   [,3]
## [1,] -0.961  1.12  0.981
## [2,]  1.895 -2.35  2.384
## [3,]  1.529  2.20 -3.181
```

Unconditional Advil

```
ad_Xg_Pj_25 = omega_i[4]*(1+omega_g[2])+beta_g[2]*(omega_g[2]*omega_i[4])*(1+delta_1)
ad_Xg_Pj_50 = omega_i[5]*(1+omega_g[2])+beta_g[2]*(omega_g[2]*omega_i[5])*(1+delta_1)
ad_Xg_Pj_100 = omega_i[6]*(1+omega_g[2])+beta_g[2]*(omega_g[2]*omega_i[6])*(1+delta_1)

## Unconditional Advil25 with 25, 50, 100
  ad25_25_un = (1/omega_i[4])*(gamma_ij[10]+gamma_ij[10])+beta_i[4]*(ad_Xg_Pj_25-omega_i[4])-1+ad_Xg_Pj
  ad25_50_un = (1/omega_i[4])*(gamma_ij[11]+gamma_ij[13])+beta_i[4]*(ad_Xg_Pj_50-omega_i[5])+ad_Xg_Pj_50
  ad25_100_un = (1/omega_i[4])*(gamma_ij[12]+gamma_ij[16])+beta_i[4]*(ad_Xg_Pj_100-omega_i[6])+ad_Xg_Pj


## Unconditional Advil50 with 25, 50, 100
  ad50_25_un = (1/omega_i[5])*(gamma_ij[11]+gamma_ij[13])+beta_i[5]*(ad_Xg_Pj_25-omega_i[4])+ad_Xg_Pj_25
```

```
  ad50_50_un = (1/omega_i[5])*(gamma_ij[14]+gamma_ij[14])+beta_i[5]*(ad_Xg_Pj_50-omega_i[5])+ad_Xg_Pj_50
  ad50_100_un = (1/omega_i[5])*(gamma_ij[15]+gamma_ij[17])+beta_i[5]*(ad_Xg_Pj_100-omega_i[6])+ad_Xg_Pj

## Unconditional Advil100 with 25, 50, 100
  ad100_25_un = (1/omega_i[6])*(gamma_ij[12]+gamma_ij[16])+beta_i[6]*(ad_Xg_Pj_25-omega_i[4])+ad_Xg_Pj_1
  ad100_50_un = (1/omega_i[6])*(gamma_ij[15]+gamma_ij[17])+beta_i[6]*(ad_Xg_Pj_50-omega_i[5])+ad_Xg_Pj_5
  ad100_100_un = (1/omega_i[6])*(gamma_ij[18]+gamma_ij[18])+beta_i[6]*(ad_Xg_Pj_100-omega_i[6])+ad_Xg_P
uncon_ad<- c(ad25_25_un,ad25_50_un,ad25_100_un,ad50_25_un,ad50_50_un,ad50_100_un,ad100_25_un,ad100_50_un
dim(uncon_ad) <- c(3,3)
uncon_ad
```

```
##       [,1]  [,2]  [,3]
## [1,] -2.80  2.69  2.05
## [2,]  2.76 -5.14  4.77
## [3,]  1.53  3.46 -8.46
```

Unconditional Bayer

```
ba_Xg_Pj_25 = omega_i[7]*(1+omega_g[3])+beta_g[3]*(omega_g[3]*omega_i[7])*(1+delta_1)
ba_Xg_Pj_50 = omega_i[8]*(1+omega_g[3])+beta_g[3]*(omega_g[3]*omega_i[8])*(1+delta_1)
ba_Xg_Pj_100 = omega_i[9]*(1+omega_g[3])+beta_g[3]*(omega_g[3]*omega_i[9])*(1+delta_1)

## Unconditional Advil25 with 25, 50, 100
  ba25_25_un = (1/omega_i[7])*(gamma_ij[19]+gamma_ij[19])+beta_i[7]*(ba_Xg_Pj_25-omega_i[7])-1+ba_Xg_Pj
  ba25_50_un = (1/omega_i[7])*(gamma_ij[20]+gamma_ij[22])+beta_i[7]*(ba_Xg_Pj_50-omega_i[8])+ba_Xg_Pj_50
  ba25_100_un = (1/omega_i[7])*(gamma_ij[21]+gamma_ij[25])+beta_i[7]*(ba_Xg_Pj_100-omega_i[9])+ba_Xg_Pj

## Unconditional Advil50 with 25, 50, 100
  ba50_25_un = (1/omega_i[8])*(gamma_ij[20]+gamma_ij[22])+beta_i[8]*(ba_Xg_Pj_25-omega_i[7])+ba_Xg_Pj_25
  ba50_50_un = (1/omega_i[8])*(gamma_ij[23]+gamma_ij[23])+beta_i[8]*(ba_Xg_Pj_50-omega_i[8])+ba_Xg_Pj_50
  ba50_100_un = (1/omega_i[8])*(gamma_ij[24]+gamma_ij[26])+beta_i[8]*(ba_Xg_Pj_100-omega_i[9])+ba_Xg_Pj

## Unconditional Advil100 with 25, 50, 100
  ba100_25_un = (1/omega_i[9])*(gamma_ij[21]+gamma_ij[25])+beta_i[9]*(ba_Xg_Pj_25-omega_i[7])+ba_Xg_Pj_1
  ba100_50_un = (1/omega_i[9])*(gamma_ij[24]+gamma_ij[26])+beta_i[9]*(ba_Xg_Pj_50-omega_i[8])+ba_Xg_Pj_5
  ba100_100_un = (1/omega_i[9])*(gamma_ij[27]+gamma_ij[27])+beta_i[9]*(ba_Xg_Pj_100-omega_i[9])+ba_Xg_P

uncon_ba<- c(ba25_25_un,ba25_50_un,ba25_100_un,ba50_25_un,ba50_50_un,ba50_100_un,ba100_25_un,ba100_50_un
dim(uncon_ba) <- c(3,3)
uncon_ba
```

```
##       [,1]  [,2]   [,3]
## [1,]  1.60 -2.14  0.537
## [2,] -2.27  4.01  1.443
## [3,]  1.40  3.60 -4.009
```