MSBX-5310: Assignment 2 - Solution

due: 2/4/19

Overview & instructions

For homework, you will analyze a different panel dataset from a different product category.

Instructions

1. Due date: Feb 4th 2019, 8:00 AM.
2. Your task is to fill in all R code blocks that currently contain "#TBD" comments. Similarly, insert text responses wherever you see *TBD* in the markdown file.
3. PLEASE UNCOMMENT LINE 2 AND ADD YOUR NAME

In-class consulting project description

Your task is to construct a linear model of demand for a common packaged good (a laundry detergent). The effect of interest is the average effect of price on demand (units sold).

You have access to scanner data across a set of stores of a retail chain in the Chicago metro region. The data are in the file `detergent_data.csv` (the file is available on Canvas). The variables in the data set are:

`store`	Store id number
`week`	Week number
`promoflag`	= 1 if any product in the category was on promotion
`sales`	Tide 128oz laundry detergent: unit sales
`price`	Tide 128oz laundry detergent: price ($)

Homework tasks

1. Data description
    1. Load data into R
    2. Sumarize the data using `describe()`, etc.
    3. Scatterplot of `sales` vs. `price`
2. Regresion models of DV = `sales`
    1. `model1L` – regressors: `price`, `promoflag`
    2. `model2L` – regressors: `price`, `promoflag`, linear time (week) trend
    3. `model3L` – regressors: `price`, `promoflag`, time (week) fixed effects
    4. `model1P` – regressors: `price`, `promoflag`, store fixed effects
    5. `model2P` – regressors: `price`, `promoflag`, linear time (week) trend, store fixed effects
    6. `model3P` – regressors: `price`, `promoflag`, time (week) fixed effects, store fixed effects
    7. Build a results table
        1. Discussion questions
3. Prediction exercises
    1. `model3L.yhat1` – using `model3L`, predict baseline expected sales
        1. Compute aggregate (total) expected sales
    2. `model3L.yhat2` – using `model3L`, predict expected sales with prices increased by 1%
        1. Compute aggregate (total) expected sales

2. Compute percentage change in sales from 1% price increase
3. `model3P.yhat1` – using `model3P`, predict baseline expected sales

1) Data Description

1.1) Read in the data

Read in the data from the csv file, and store to dataframe `DF1`.

Also, report (print):

1) the number of unique stores in the dataset, and
2) number of unique time periods in the dataset

Hint: the functions `unique()` and `length()` can be useful in calculating the number of unique observations

```
DF1 = read.csv("detergent_data.csv")
N_store = length(unique(DF1$store))
N_week = length(unique(DF1$week))
print(sprintf("# stores = %d, # weeks = %d",N_store,N_week))
```

```
[1] "# stores = 86, # weeks = 224"
```

1.2) Use `describe()` to sumarize the data

```
library(psych)
describe(DF1)
```

```
          vars     n  mean     sd median trimmed   mad  min     max
store        1 14744 80.98  35.80  86.00   83.38 40.03 2.00  139.00
week         2 14744 99.12  53.94 101.00   99.75 66.72 1.00  300.00
sales        3 14744 81.22 134.14  48.00   55.67 31.13 1.00 2224.00
price        4 14744  8.36   0.76   8.48    8.40  0.67 4.88   10.51
promoflag    5 14744  0.82   0.39   1.00    0.90  0.00 0.00    1.00
            range  skew kurtosis   se
store      137.00 -0.46    -0.70 0.29
week       299.00 -0.02    -0.91 0.44
sales     2223.00  7.43    77.56 1.10
price        5.63 -0.47     0.18 0.01
promoflag    1.00 -1.65     0.73 0.00
```

Discussion questions

1) What is the total number of observations?

The dataset has 14,744 observations

2) Is the data a balanced or unbalanced panel? Why?

The data is a unbalanced panel. The number of observations is not equivalent to the cross-sectional units multiplied by time periods (19,264).

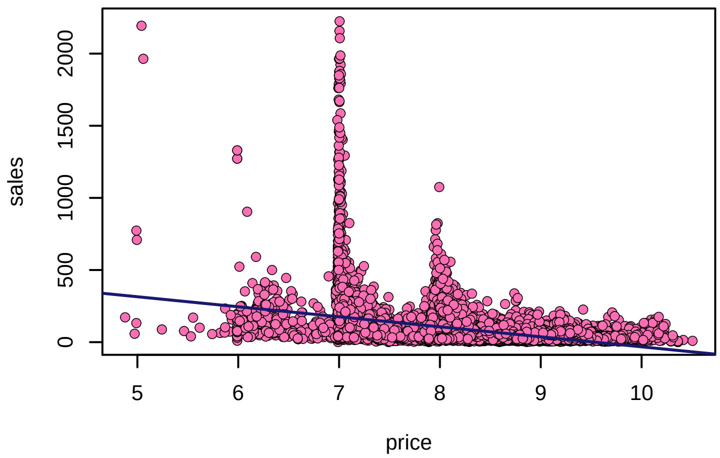
3) Interpret the mean value of the variable `promoflag`

The mean value of promoflag is 0.82. The promoflag variable is binary where 1 = promotion. The mean value indicates that 82% of store/week observations were during promotions.

1.3) Scatterplot of `sales` vs. `price`

Generate a simple scatterplot of sales vs. price. Add a fitted linear regression line.

```
par(cex = 0.65)
plot(DF1$price, DF1$sales,
     pch = 21, lwd = 0.4, bg = "hotpink1",
     xlab = "price", ylab = "sales")
lm_fit = lm(sales~price, data=DF1)
abline(lm_fit,lwd = 1.5, col = "midnightblue")
```



Discussion questions

1) Comment on the distribution of sales and prices. What patterns do you notice?

According to the fitted linear regression line, sales, in general, decrease as price increases, which makes economic sense. The data show most stores have a high proportion of prices at near even dollar increments (price = $7 shows the most related sales followed by when price = $8).

2) Is the (sign of the) fitted regression line slope as expected? Why or why not?

The slope of the fitted regression line is expected. Traditional economics support that sales decrease as price increase. The line shows this trend.

3) Compared to the data in Workshop 2, does this suggest we should be more or less concerned about omitted variable bias?

Compared to the data in Workshop 2, we should be somewhat less concerned about omitted variable bias to the extent the effect of price on demand appears consistent with traditional economic theory.

2) Regresion models of DV = sales

In sections 2.1 - 2.6, you will run a series of regressions. In secton 2.7 you will summarize key results and answer discussion questions.

2.1) `model1L` – regressors: `price`, `promoflag`

Use `lm()` to estimate a model of sales with regressors: `price`, `promoflag`. Name the model `model1L`. Use `summary()` to summarize the results.

```
model1L = lm(sales~price + promoflag, data=DF1)
summary(model1L)
```

```

Call:
lm(formula = sales ~ price + promoflag, data = DF1)

Residuals:
    Min      1Q  Median      3Q     Max 
-259.94  -47.25  -15.51   19.36 2045.53 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  636.795     11.573  55.024   <2e-16 ***
price        -68.622      1.336 -51.347   <2e-16 ***
promoflag     22.379      2.635   8.492   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 123 on 14741 degrees of freedom
Multiple R-squared:  0.1588,	Adjusted R-squared:  0.1587 
F-statistic:  1391 on 2 and 14741 DF,  p-value: < 2.2e-16
```

2.2) `model2L` – regressors: `price`, `promoflag`, linear time (week) trend

Use `lm()` to estimate a model of sales with regressors: `price`, `promoflag` and a linear time (week) trend. Name the model `model2L`. Use `summary()` to summarize the results.

```
model2L = lm(sales~price + promoflag + week, data=DF1)
summary(model2L)
```

```

Call:
lm(formula = sales ~ price + promoflag + week, data = DF1)

Residuals:
    Min      1Q  Median      3Q     Max 
-262.39  -47.39  -15.66   19.46 2044.02 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 633.94646   11.73102  54.040  < 2e-16 ***
price       -68.54153    1.33747 -51.247  < 2e-16 ***
promoflag    21.59894    2.68730   8.037 9.87e-16 ***
week          0.02843    0.01918   1.482    0.138    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 123 on 14740 degrees of freedom
```

```
Multiple R-squared:  0.1589,	Adjusted R-squared:  0.1587 
F-statistic: 928.2 on 3 and 14740 DF,  p-value: < 2.2e-16
```

2.3) `model3L` – regressors: `price`, `promoflag`, time (week) fixed effects

Use `lm()` to estimate a model of sales with regressors: `price`, `promoflag` and time (week) fixed effects. Name the model `model3L`. Use `summary()` to summarize the results.

```r
model3L = lm(sales~price + promoflag + factor(week), data=DF1)
summary(model3L)
```

```
Call:
lm(formula = sales ~ price + promoflag + factor(week), data = DF1)

Residuals:
     Min       1Q   Median       3Q      Max 
-1191.94   -18.41    -3.96    12.65  1878.07 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      179.2997    12.7714  14.039  < 2e-16 ***
price            -17.3808     1.2545 -13.854  < 2e-16 ***
promoflag         -7.2057     3.0677  -2.349 0.018845 *  
factor(week)2     -4.2591    10.8031  -0.394 0.693406    
factor(week)3      7.1346    10.7676   0.663 0.507599    
factor(week)4    105.6222    10.6652   9.903  < 2e-16 ***
factor(week)5     67.2293    10.2984   6.528 6.88e-11 ***
factor(week)6     21.1420    10.3973   2.033 0.042030 *  
factor(week)7     30.3143    10.4385   2.904 0.003689 ** 
factor(week)8     31.5211    10.2880   3.064 0.002189 ** 
factor(week)9     22.2312    10.1885   2.182 0.029127 *  
factor(week)10    24.1935    10.7717   2.246 0.024718 *  
factor(week)11     5.3013    10.9161   0.486 0.627227    
factor(week)12    23.7954    10.6671   2.231 0.025715 *  
factor(week)13    33.6070    10.4728   3.209 0.001335 ** 
factor(week)14    52.7934    10.2992   5.126 3.00e-07 ***
factor(week)15    48.3027    10.2661   4.705 2.56e-06 ***
factor(week)16    63.8106    10.3656   6.156 7.66e-10 ***
factor(week)17   368.5436    10.2499  35.956  < 2e-16 ***
factor(week)18    21.7282    10.7363   2.024 0.043010 *  
factor(week)19    48.4430    10.7443   4.509 6.57e-06 ***
factor(week)20   333.7572    10.5487  31.640  < 2e-16 ***
factor(week)21    44.5825    10.8911   4.093 4.27e-05 ***
factor(week)22    18.5942    10.8013   1.721 0.085185 .  
factor(week)23    22.5426    10.8458   2.078 0.037685 *  
factor(week)24    16.7553    11.1806   1.499 0.134000    
factor(week)25    20.5327    11.3084   1.816 0.069437 .  
factor(week)26     7.4717    11.3090   0.661 0.508823    
factor(week)27    13.3017    11.6735   1.139 0.254522    
factor(week)28   126.9142    11.3049  11.226  < 2e-16 ***
factor(week)29    57.6310    11.5619   4.985 6.28e-07 ***
factor(week)30    25.4194    11.8501   2.145 0.031962 *  
factor(week)31    27.3902    11.5488   2.372 0.017720 *  
```

```
factor(week)32    32.7177    11.3413   2.885 0.003922 **
factor(week)33    21.5342    11.1378   1.933 0.053201 .
factor(week)34    19.9710    11.2802   1.770 0.076673 .
factor(week)35    23.5177    11.2427   2.092 0.036473 *
factor(week)36    42.7294    11.0479   3.868 0.000110 ***
factor(week)37    44.6859    10.2889   4.343 1.41e-05 ***
factor(week)38    44.1526    10.2205   4.320 1.57e-05 ***
factor(week)39    35.9987    10.2099   3.526 0.000423 ***
factor(week)40    40.3781    10.3262   3.910 9.26e-05 ***
factor(week)41    37.9819    10.2181   3.717 0.000202 ***
factor(week)42    36.8931    10.1085   3.650 0.000263 ***
factor(week)43    31.6829    10.0770   3.144 0.001669 **
factor(week)44    12.7436    10.5854   1.204 0.228653
factor(week)45    21.4324    10.1278   2.116 0.034346 *
factor(week)46    20.7015    10.1558   2.038 0.041529 *
factor(week)47    20.5786    10.2146   2.015 0.043962 *
factor(week)48     5.5088    10.6186   0.519 0.603914
factor(week)49     8.8089    10.6693   0.826 0.409030
factor(week)50    14.7092    10.0570   1.463 0.143602
factor(week)51    19.0112    10.0468   1.892 0.058476 .
factor(week)52    14.0450    10.0528   1.397 0.162398
factor(week)53    24.2244    10.0297   2.415 0.015736 *
factor(week)54   188.1124    10.0515  18.715  < 2e-16 ***
factor(week)55    38.3523    10.0777   3.806 0.000142 ***
factor(week)56    -1.1565    10.0017  -0.116 0.907944
factor(week)57    -4.7928    10.4569  -0.458 0.646718
factor(week)58     2.4062    10.1506   0.237 0.812619
factor(week)59     3.9848    10.3925   0.383 0.701408
factor(week)60    23.2274    10.0277   2.316 0.020554 *
factor(week)61   121.1052    10.0801  12.014  < 2e-16 ***
factor(week)62    44.5783    10.0692   4.427 9.62e-06 ***
factor(week)63     5.3834     9.9838   0.539 0.589745
factor(week)64     5.8605    10.4604   0.560 0.575315
factor(week)65     1.2656     9.9762   0.127 0.899050
factor(week)66    13.1524     9.9120   1.327 0.184556
factor(week)67    31.8729     9.9468   3.204 0.001357 **
factor(week)68    44.3713    10.0841   4.400 1.09e-05 ***
factor(week)69    45.7043    10.0262   4.559 5.19e-06 ***
factor(week)70    41.1927    10.0887   4.083 4.47e-05 ***
factor(week)71    41.4639     9.9849   4.153 3.30e-05 ***
factor(week)72    58.9130    10.0208   5.879 4.22e-09 ***
factor(week)73   303.8825    10.0267  30.307  < 2e-16 ***
factor(week)74    40.2134     9.9938   4.024 5.76e-05 ***
factor(week)75    33.8421    10.0237   3.376 0.000737 ***
factor(week)76    38.3706    10.0230   3.828 0.000130 ***
factor(week)77    35.4360    10.0218   3.536 0.000408 ***
factor(week)78    54.3113    10.0846   5.386 7.33e-08 ***
factor(week)79   171.1303    10.1337  16.887  < 2e-16 ***
factor(week)80    49.6456    10.1263   4.903 9.56e-07 ***
factor(week)81    55.1670    10.0225   5.504 3.77e-08 ***
factor(week)82    50.9635    10.0985   5.047 4.55e-07 ***
factor(week)83    48.9224    10.0128   4.886 1.04e-06 ***
factor(week)84   293.9316    10.0249  29.320  < 2e-16 ***
factor(week)85    46.1738    10.0083   4.614 3.99e-06 ***
```

```
factor(week)86    53.7184     9.9987   5.373 7.88e-08 ***
factor(week)87    62.5244    10.0427   6.226 4.92e-10 ***
factor(week)88   103.7413    10.0139  10.360  < 2e-16 ***
factor(week)89    48.6748    10.1971   4.773 1.83e-06 ***
factor(week)90    56.5608    10.0537   5.626 1.88e-08 ***
factor(week)91    31.4695    10.0184   3.141 0.001686 **
factor(week)92    37.5773     9.9300   3.784 0.000155 ***
factor(week)93    34.4123     9.9792   3.448 0.000566 ***
factor(week)94    36.4961     9.9851   3.655 0.000258 ***
factor(week)95    24.3166    10.0267   2.425 0.015312 *
factor(week)96    20.3438    10.0312   2.028 0.042574 *
factor(week)97    20.4977    10.0303   2.044 0.041014 *
factor(week)98    15.7109    10.4708   1.500 0.133519
factor(week)99    15.9032    10.4724   1.519 0.128890
factor(week)100   16.7480     9.9992   1.675 0.093968 .
factor(week)101   15.4032    10.5299   1.463 0.143542
factor(week)102   21.1013    10.0264   2.105 0.035345 *
factor(week)103   17.9604    10.5010   1.710 0.087222 .
factor(week)104   23.0040    10.0535   2.288 0.022143 *
factor(week)105  236.1183     9.9571  23.714  < 2e-16 ***
factor(week)106  163.0363     9.9672  16.357  < 2e-16 ***
factor(week)107   32.9097     9.8963   3.325 0.000885 ***
factor(week)108   47.8498    10.2052   4.689 2.77e-06 ***
factor(week)109   23.7042     9.9779   2.376 0.017530 *
factor(week)110   24.2984    10.3804   2.341 0.019257 *
factor(week)111   39.7323     9.9097   4.009 6.12e-05 ***
factor(week)112   25.5935     9.9247   2.579 0.009925 **
factor(week)113  221.0864     9.9092  22.311  < 2e-16 ***
factor(week)114  119.2756     9.9143  12.031  < 2e-16 ***
factor(week)115   39.7831     9.9662   3.992 6.59e-05 ***
factor(week)116   24.4513     9.9390   2.460 0.013900 *
factor(week)117   26.6132     9.9392   2.678 0.007424 **
factor(week)118   26.5014     9.9095   2.674 0.007496 **
factor(week)119   27.9500     9.9323   2.814 0.004899 **
factor(week)120   36.6042     9.9109   3.693 0.000222 ***
factor(week)121   35.0496     9.9317   3.529 0.000418 ***
factor(week)122  125.9602    10.0031  12.592  < 2e-16 ***
factor(week)123   25.5963     9.9056   2.584 0.009775 **
factor(week)124   17.5425     9.9489   1.763 0.077876 .
factor(week)125   20.1312     9.9533   2.023 0.043137 *
factor(week)126   27.9608     9.9188   2.819 0.004824 **
factor(week)127  201.9281    10.0769  20.039  < 2e-16 ***
factor(week)128   13.9993     9.9048   1.413 0.157563
factor(week)129   28.6200     9.9021   2.890 0.003855 **
factor(week)130   16.1986     9.9103   1.635 0.102170
factor(week)131   18.3100     9.9058   1.848 0.064565 .
factor(week)132   17.8786     9.9652   1.794 0.072816 .
factor(week)133   20.0554     9.9378   2.018 0.043599 *
factor(week)134   19.4797     9.9176   1.964 0.049532 *
factor(week)135   26.7934     9.9058   2.705 0.006842 **
factor(week)136   94.8138     9.8966   9.580  < 2e-16 ***
factor(week)137   17.2997     9.8579   1.755 0.079296 .
factor(week)138   21.0836     9.8960   2.131 0.033145 *
factor(week)139   42.8556     9.9132   4.323 1.55e-05 ***
```

```
factor(week)140   32.7314     9.8574   3.320 0.000901 ***
factor(week)141   23.3636     9.8689   2.367 0.017927 *
factor(week)142   28.2204    10.0108   2.819 0.004824 **
factor(week)143   95.4180    10.0084   9.534  < 2e-16 ***
factor(week)144   17.5950     9.9636   1.766 0.077430 .
factor(week)145   20.7300    10.0328   2.066 0.038825 *
factor(week)146   20.4586     9.9451   2.057 0.039689 *
factor(week)147   25.1984     9.9487   2.533 0.011324 *
factor(week)148   88.0258     9.9367   8.859  < 2e-16 ***
factor(week)149   32.6844     9.8952   3.303 0.000959 ***
factor(week)150   13.7384    10.0233   1.371 0.170505
factor(week)151   18.8133     9.9274   1.895 0.058101 .
factor(week)152   19.8200     9.8978   2.002 0.045253 *
factor(week)153   11.9356    10.4270   1.145 0.252362
factor(week)154   57.4455     9.8579   5.827 5.75e-09 ***
factor(week)155   18.8106    10.0070   1.880 0.060163 .
factor(week)156   16.2379     9.9254   1.636 0.101862
factor(week)157  230.4539     9.8825  23.319  < 2e-16 ***
factor(week)158  120.2192     9.8559  12.198  < 2e-16 ***
factor(week)159   17.9834     9.9079   1.815 0.069537 .
factor(week)160   18.8139     9.8843   1.903 0.057008 .
factor(week)161 1174.6969     9.9166 118.458  < 2e-16 ***
factor(week)162   26.6810     9.8638   2.705 0.006840 **
factor(week)163   10.3208     9.9157   1.041 0.297961
factor(week)164   13.5380     9.8832   1.370 0.170772
factor(week)165    8.2425    10.0769   0.818 0.413395
factor(week)166    9.2891     9.9148   0.937 0.348828
factor(week)167   13.9235     9.9152   1.404 0.160262
factor(week)168    0.3635     9.8640   0.037 0.970605
factor(week)169   21.5975     9.8607   2.190 0.028521 *
factor(week)170    2.1990     9.8916   0.222 0.824079
factor(week)171    2.6709     9.9760   0.268 0.788912
factor(week)172    1.2012     9.8916   0.121 0.903349
factor(week)173    6.4766     9.8905   0.655 0.512590
factor(week)174   81.0736    10.2697   7.894 3.12e-15 ***
factor(week)175   36.1773    10.2702   3.523 0.000429 ***
factor(week)176   -1.6349     9.8931  -0.165 0.868743
factor(week)177   -0.2837     9.8906  -0.029 0.977117
factor(week)178  611.0807     9.9709  61.286  < 2e-16 ***
factor(week)179   92.3112     9.9782   9.251  < 2e-16 ***
factor(week)180  -10.2435    10.3341  -0.991 0.321591
factor(week)181   -2.6761     9.9183  -0.270 0.787308
factor(week)182   -3.9632     9.8910  -0.401 0.688654
factor(week)183   -3.3458     9.8892  -0.338 0.735117
factor(week)184   -6.3216    10.0039  -0.632 0.527457
factor(week)185   -7.9301     9.9212  -0.799 0.424124
factor(week)186   39.1561    10.1801   3.846 0.000120 ***
factor(week)187   -1.7573     9.9199  -0.177 0.859392
factor(week)188  -14.6410    11.9202  -1.228 0.219373
factor(week)189  -11.0933    24.3997  -0.455 0.649367
factor(week)190  579.7828    31.4204  18.452  < 2e-16 ***
factor(week)191  -20.0315    43.7986  -0.457 0.647423
factor(week)192  -13.1714    43.7910  -0.301 0.763588
factor(week)193  -21.7162    61.5765  -0.353 0.724341
```

```
factor(week)194  -28.7832    61.4996  -0.468 0.639776
factor(week)195  -34.5086    61.4989  -0.561 0.574721
factor(week)196  -32.7168    61.5009  -0.532 0.594753
factor(week)197  -12.6022    61.5125  -0.205 0.837675
factor(week)198  -29.9069    61.5114  -0.486 0.626832
factor(week)199  -35.8126    61.5777  -0.582 0.560855
factor(week)200  -27.6635    61.6200  -0.449 0.653484
factor(week)204   31.9950    61.5148   0.520 0.602989
factor(week)205   -3.1236    61.5444  -0.051 0.959523
factor(week)207   18.8311    61.5445   0.306 0.759628
factor(week)210  -10.9277    61.5496  -0.178 0.859084
factor(week)212  -30.2741    61.5208  -0.492 0.622659
factor(week)229   24.3212    61.5159   0.395 0.692580
factor(week)255  -42.7601    61.5990  -0.694 0.487587
factor(week)256  -38.7601    61.5990  -0.629 0.529206
factor(week)277   16.2798    43.8825   0.371 0.710653
factor(week)278 1232.5170    43.8680  28.096  < 2e-16 ***
factor(week)279  -28.5465    43.8297  -0.651 0.514860
factor(week)280   27.8266    43.9098   0.634 0.526272
factor(week)281  -19.5654    36.1053  -0.542 0.587897
factor(week)282 1232.5170    43.8680  28.096  < 2e-16 ***
factor(week)283  -28.5465    43.8297  -0.651 0.514860
factor(week)284  -17.7229    43.8128  -0.405 0.685840
factor(week)285  -11.2128    43.8128  -0.256 0.798011
factor(week)286    8.1879    43.8130   0.187 0.851755
factor(week)287  655.6741    43.9602  14.915  < 2e-16 ***
factor(week)288  -30.8935    43.8234  -0.705 0.480850
factor(week)289  -18.6022    61.5125  -0.302 0.762342
factor(week)298  -36.9830    61.5528  -0.601 0.547960
factor(week)299   -4.9830    36.0965  -0.138 0.890206
factor(week)300  -18.9830    61.5528  -0.308 0.757781
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61.06 on 14518 degrees of freedom
Multiple R-squared:  0.7959,    Adjusted R-squared:  0.7928
F-statistic: 251.7 on 225 and 14518 DF,  p-value: < 2.2e-16
```

2.4) `model1P` – regressors: `price`, `promoflag`, store fixed effects

Use `plm()` to estimate a store fixed-effects model of sales with regressors: `price`, `promoflag`. Name the model `model1P`. Use `summary()` to summarize the results.

```r
library(plm)
model1P = plm(sales~price + promoflag, data=DF1,
              index=c("store","week"), model="within", effect="individual")
summary(model1P)
```

```
Oneway (individual) effect Within Model

Call:
plm(formula = sales ~ price + promoflag, data = DF1, effect = "individual",
    model = "within", index = c("store", "week"))
```

```
Unbalanced Panel: n = 86, T = 32-193, N = 14744

Residuals:
     Min.   1st Qu.    Median   3rd Qu.      Max.
-237.9065  -43.8488   -9.6513   19.8351 2011.5750

Coefficients:
          Estimate Std. Error t-value  Pr(>|t|)
price     -76.4248     1.3877 -55.072 < 2.2e-16 ***
promoflag  21.8063     2.5694   8.487 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    253460000
Residual Sum of Squares: 207950000
R-Squared:      0.17956
Adj. R-Squared: 0.17469
F-statistic: 1603.84 on 2 and 14656 DF, p-value: < 2.22e-16
```

2.5) `model2P` – regressors: `price`, `promoflag`, linear time (week) trend, store fixed effects

Use `plm()` to estimate a store fixed-effects model of sales with regressors: `price`, `promoflag`, and a linear time (week) trend. Name the model `model2P`. Use `summary()` to summarize the results.

```r
model2P = plm(sales~price + promoflag + as.numeric(week) , data=DF1,
              index=c("store","week"), model="within", effect="individual")
summary(model2P)
```

```
Oneway (individual) effect Within Model

Call:
plm(formula = sales ~ price + promoflag + as.numeric(week), data = DF1,
    effect = "individual", model = "within", index = c("store",
        "week"))

Unbalanced Panel: n = 86, T = 32-193, N = 14744

Residuals:
     Min.   1st Qu.    Median   3rd Qu.      Max.
-239.3171  -43.7496   -9.7696   20.0042 2010.5522

Coefficients:
                   Estimate Std. Error  t-value  Pr(>|t|)
price            -76.393044   1.388175 -55.0313 < 2.2e-16 ***
promoflag         21.339086   2.621159   8.1411 4.231e-16 ***
as.numeric(week)   0.017080   0.018947   0.9015    0.3674
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    253460000
Residual Sum of Squares: 207930000
R-Squared:      0.17961
Adj. R-Squared: 0.17468
```

```
F-statistic: 1069.49 on 3 and 14655 DF, p-value: < 2.22e-16
```

2.6) `model3P` – regressors: `price`, `promoflag`, time (week) fixed effects, store fixed effects

Use `plm()` to estimate a store fixed-effects model of sales with regressors: `price`, `promoflag`, and time (week) fixed effects. Name the model `model3P`. Use `summary()` to summarize the results.

```
model3P = plm(sales~price + promoflag + factor(week), data=DF1,
              index=c("store","week"), model="within", effect="individual")
summary(model3P)
```

```
Oneway (individual) effect Within Model

Call:
plm(formula = sales ~ price + promoflag + factor(week), data = DF1,
    effect = "individual", model = "within", index = c("store",
        "week"))

Unbalanced Panel: n = 86, T = 32-193, N = 14744

Residuals:
       Min.     1st Qu.      Median     3rd Qu.        Max.
-1.1609e+03 -1.4138e+01  3.2946e-02  1.2812e+01  1.7795e+03

Coefficients:
                  Estimate Std. Error  t-value  Pr(>|t|)
price            -22.49912    1.49199 -15.0799 < 2.2e-16 ***
promoflag          0.21908    2.81087   0.0779 0.9378763
factor(week)2      4.20899    9.67362   0.4351 0.6634964
factor(week)3     16.47626    9.64064   1.7090 0.0874647 .
factor(week)4    114.76769    9.54864  12.0193 < 2.2e-16 ***
factor(week)5     64.72069    9.25411   6.9937 2.794e-12 ***
factor(week)6     26.04145    9.29189   2.8026 0.0050760 **
factor(week)7     31.67467    9.32950   3.3951 0.0006879 ***
factor(week)8     33.75462    9.19479   3.6711 0.0002424 ***
factor(week)9     25.55041    9.10635   2.8058 0.0050262 **
factor(week)10    33.82367    9.64416   3.5072 0.0004543 ***
factor(week)11    17.78190    9.78827   1.8167 0.0692906 .
factor(week)12    36.43330    9.56941   3.8073 0.0001411 ***
factor(week)13    41.59171    9.43447   4.4085 1.048e-05 ***
factor(week)14    59.58188    9.25915   6.4349 1.274e-10 ***
factor(week)15    54.94134    9.20901   5.9660 2.488e-09 ***
factor(week)16    71.15448    9.29370   7.6562 2.036e-14 ***
factor(week)17   370.63440    9.15946  40.4647 < 2.2e-16 ***
factor(week)18    34.03894    9.62836   3.5353 0.0004086 ***
factor(week)19    59.23457    9.62504   6.1542 7.744e-10 ***
factor(week)20   330.93556    9.46997  34.9458 < 2.2e-16 ***
factor(week)21    40.09550    9.76431   4.1063 4.042e-05 ***
factor(week)22    23.89519    9.65807   2.4741 0.0133680 *
factor(week)23    24.12954    9.69632   2.4885 0.0128386 *
factor(week)24    20.02059    9.99653   2.0028 0.0452223 *
factor(week)25    24.22654   10.11107   2.3960 0.0165859 *
factor(week)26    12.54585   10.11187   1.2407 0.2147351
```

```
factor(week)27    27.49209   10.45868   2.6286 0.0085818 **
factor(week)28   125.38418   10.15165  12.3511 < 2.2e-16 ***
factor(week)29    57.12550   10.37193   5.5077 3.698e-08 ***
factor(week)30    34.20462   10.60737   3.2246 0.0012643 **
factor(week)31    37.04989   10.34032   3.5830 0.0003407 ***
factor(week)32    44.40028   10.16068   4.3698 1.252e-05 ***
factor(week)33    32.13470    9.97096   3.2228 0.0012721 **
factor(week)34    29.50139   10.09971   2.9210 0.0034943 **
factor(week)35    28.20441   10.05991   2.8036 0.0050596 **
factor(week)36    48.46299    9.90936   4.8906 1.016e-06 ***
factor(week)37    52.81652    9.26492   5.7007 1.216e-08 ***
factor(week)38    51.93790    9.20446   5.6427 1.706e-08 ***
factor(week)39    46.83828    9.19074   5.0962 3.508e-07 ***
factor(week)40    51.03958    9.30608   5.4845 4.215e-08 ***
factor(week)41    47.14142    9.20142   5.1233 3.041e-07 ***
factor(week)42    43.14087    9.06953   4.7567 1.987e-06 ***
factor(week)43    38.40246    9.04116   4.2475 2.175e-05 ***
factor(week)44    27.02222    9.51108   2.8411 0.0045017 **
factor(week)45    28.33412    9.09946   3.1138 0.0018504 **
factor(week)46    27.28464    9.12205   2.9911 0.0027848 **
factor(week)47    28.73281    9.19527   3.1247 0.0017832 **
factor(week)48    20.65888    9.56109   2.1607 0.0307330 *
factor(week)49    23.79146    9.60070   2.4781 0.0132199 *
factor(week)50    21.68925    9.02936   2.4021 0.0163147 *
factor(week)51    25.29520    9.01335   2.8064 0.0050163 **
factor(week)52    20.20118    9.00371   2.2437 0.0248700 *
factor(week)53    28.05128    8.96702   3.1283 0.0017619 **
factor(week)54   185.09250    9.04104  20.4725 < 2.2e-16 ***
factor(week)55    36.99893    9.04200   4.0919 4.302e-05 ***
factor(week)56    -0.37116    8.94231  -0.0415 0.9668932
factor(week)57     3.99230    9.37157   0.4260 0.6701135
factor(week)58     7.49633    9.08686   0.8250 0.4094061
factor(week)59    11.87236    9.31160   1.2750 0.2023271
factor(week)60    25.00328    8.96382   2.7894 0.0052883 **
factor(week)61   117.31123    9.06765  12.9373 < 2.2e-16 ***
factor(week)62    41.16009    9.04916   4.5485 5.447e-06 ***
factor(week)63     4.32767    8.93463   0.4844 0.6281308
factor(week)64    13.07163    9.37874   1.3938 0.1634144
factor(week)65     1.02729    8.92284   0.1151 0.9083434
factor(week)66    14.12622    8.85922   1.5945 0.1108411
factor(week)67    35.07992    8.89593   3.9434 8.072e-05 ***
factor(week)68    50.56276    9.03373   5.5971 2.219e-08 ***
factor(week)69    49.69946    8.96227   5.5454 2.984e-08 ***
factor(week)70    47.91716    9.05780   5.2902 1.240e-07 ***
factor(week)71    47.49397    8.95517   5.3035 1.153e-07 ***
factor(week)72    65.06917    8.99237   7.2360 4.852e-13 ***
factor(week)73   300.16729    9.02206  33.2704 < 2.2e-16 ***
factor(week)74    36.85546    8.98859   4.1003 4.150e-05 ***
factor(week)75    39.85583    8.99688   4.4300 9.494e-06 ***
factor(week)76    44.57146    8.99553   4.9548 7.321e-07 ***
factor(week)77    41.59068    8.99363   4.6245 3.788e-06 ***
factor(week)78    63.37598    9.11119   6.9558 3.655e-12 ***
factor(week)79   168.72881    9.09176  18.5584 < 2.2e-16 ***
factor(week)80    49.87806    9.05595   5.5078 3.696e-08 ***
```

```
factor(week)81    62.38482    9.01480   6.9203 4.698e-12 ***
factor(week)82    57.33180    9.05617   6.3307 2.513e-10 ***
factor(week)83    54.63943    8.98116   6.0838 1.203e-09 ***
factor(week)84   290.39988    9.01868  32.1998 < 2.2e-16 ***
factor(week)85    53.84539    9.01071   5.9757 2.345e-09 ***
factor(week)86    58.72325    8.95699   6.5561 5.709e-11 ***
factor(week)87    61.61626    8.98809   6.8553 7.404e-12 ***
factor(week)88   101.16725    8.98283  11.2623 < 2.2e-16 ***
factor(week)89    52.64338    9.12688   5.7680 8.188e-09 ***
factor(week)90    58.27182    8.98525   6.4853 9.145e-11 ***
factor(week)91    34.99342    8.97131   3.9006 9.639e-05 ***
factor(week)92    42.14262    8.88760   4.7417 2.139e-06 ***
factor(week)93    40.36810    8.93123   4.5199 6.237e-06 ***
factor(week)94    42.38299    8.94094   4.7403 2.154e-06 ***
factor(week)95    30.51347    8.98660   3.3954 0.0006870 ***
factor(week)96    27.14244    8.99493   3.0175 0.0025529 **
factor(week)97    27.68195    8.99337   3.0780 0.0020876 **
factor(week)98    28.53414    9.39496   3.0372 0.0023923 **
factor(week)99    28.24572    9.39738   3.0057 0.0026542 **
factor(week)100   23.32679    8.96286   2.6026 0.0092613 **
factor(week)101   27.94724    9.45054   2.9572 0.0031094 **
factor(week)102   27.68686    8.98736   3.0806 0.0020694 **
factor(week)103   30.89753    9.42204   3.2793 0.0010432 **
factor(week)104   29.56884    9.00883   3.2822 0.0010324 **
factor(week)105  239.28473    8.89824  26.8912 < 2.2e-16 ***
factor(week)106  167.66521    8.90902  18.8197 < 2.2e-16 ***
factor(week)107   37.22737    8.85316   4.2050 2.627e-05 ***
factor(week)108   57.75112    9.15108   6.3108 2.856e-10 ***
factor(week)109   30.25659    8.96327   3.3756 0.0007384 ***
factor(week)110   37.82819    9.32319   4.0574 4.988e-05 ***
factor(week)111   41.67637    8.85585   4.7061 2.549e-06 ***
factor(week)112   31.42926    8.89832   3.5320 0.0004136 ***
factor(week)113  223.54211    8.85526  25.2440 < 2.2e-16 ***
factor(week)114  123.12823    8.86361  13.8914 < 2.2e-16 ***
factor(week)115   44.20147    8.92619   4.9519 7.433e-07 ***
factor(week)116   29.27809    8.90227   3.2888 0.0010084 **
factor(week)117   31.45213    8.90263   3.5329 0.0004123 ***
factor(week)118   31.66645    8.87402   3.5684 0.0003603 ***
factor(week)119   32.25805    8.89284   3.6274 0.0002873 ***
factor(week)120   41.77354    8.87655   4.7061 2.549e-06 ***
factor(week)121   39.71540    8.89170   4.4666 8.009e-06 ***
factor(week)122  123.13426    8.98229  13.7086 < 2.2e-16 ***
factor(week)123   24.91228    8.86771   2.8093 0.0049712 **
factor(week)124   24.54204    8.93516   2.7467 0.0060276 **
factor(week)125   25.60667    8.92477   2.8692 0.0041215 **
factor(week)126   32.25592    8.86947   3.6367 0.0002771 ***
factor(week)127  195.09929    9.15345  21.3143 < 2.2e-16 ***
factor(week)128   11.71947    8.88452   1.3191 0.1871601
factor(week)129   34.26257    8.88155   3.8577 0.0001149 ***
factor(week)130   22.58648    8.89035   2.5406 0.0110779 *
factor(week)131   24.47277    8.88421   2.7546 0.0058831 **
factor(week)132   24.62718    8.93846   2.7552 0.0058731 **
factor(week)133   26.82574    8.91714   3.0083 0.0026314 **
factor(week)134   25.87728    8.90535   2.9058 0.0036686 **
```

```
factor(week)135   31.25686    8.86933   3.5242 0.0004262 ***
factor(week)136   92.88466    8.87168  10.4698 < 2.2e-16 ***
factor(week)137   18.09671    8.81099   2.0539 0.0400055 *
factor(week)138   26.37970    8.87211   2.9733 0.0029508 **
factor(week)139   43.33539    8.86209   4.8900 1.019e-06 ***
factor(week)140   35.14181    8.81054   3.9886 6.679e-05 ***
factor(week)141   27.10346    8.82901   3.0698 0.0021459 **
factor(week)142   31.22742    8.95921   3.4855 0.0004927 ***
factor(week)143   92.20464    8.97567  10.2727 < 2.2e-16 ***
factor(week)144   23.88048    8.94484   2.6698 0.0075993 **
factor(week)145   24.85283    8.97476   2.7692 0.0056267 **
factor(week)146   23.25723    8.89431   2.6148 0.0089361 **
factor(week)147   31.30339    8.91967   3.5095 0.0004504 ***
factor(week)148   89.66232    8.88068  10.0963 < 2.2e-16 ***
factor(week)149   36.77774    8.85248   4.1545 3.279e-05 ***
factor(week)150   22.68361    8.98850   2.5236 0.0116258 *
factor(week)151   24.12579    8.90368   2.7096 0.0067435 **
factor(week)152   25.19789    8.87501   2.8392 0.0045291 **
factor(week)153   24.40240    9.35919   2.6073 0.0091348 **
factor(week)154   59.96576    8.81142   6.8055 1.047e-11 ***
factor(week)155   27.27122    8.97823   3.0375 0.0023899 **
factor(week)156   21.94291    8.90153   2.4651 0.0137101 *
factor(week)157  231.22806    8.83289  26.1781 < 2.2e-16 ***
factor(week)158  120.76911    8.80875  13.7101 < 2.2e-16 ***
factor(week)159   22.78552    8.89141   2.5626 0.0103979 *
factor(week)160   23.72362    8.87301   2.6737 0.0075108 **
factor(week)161 1170.03242    8.92015 131.1673 < 2.2e-16 ***
factor(week)162   30.70248    8.84039   3.4730 0.0005162 ***
factor(week)163   15.33552    8.90492   1.7221 0.0850653 .
factor(week)164   18.40416    8.87124   2.0746 0.0380424 *
factor(week)165   17.82447    9.04576   1.9705 0.0488027 *
factor(week)166   14.18347    8.90337   1.5930 0.1111718
factor(week)167   18.81132    8.90408   2.1127 0.0346468 *
factor(week)168    2.48109    8.82237   0.2812 0.7785407
factor(week)169   20.77333    8.81612   2.3563 0.0184719 *
factor(week)170    4.27142    8.84803   0.4828 0.6292781
factor(week)171    5.04931    8.92491   0.5658 0.5715692
factor(week)172    3.27428    8.84804   0.3701 0.7113451
factor(week)173    8.45202    8.84638   0.9554 0.3393807
factor(week)174   87.62103    9.19246   9.5318 < 2.2e-16 ***
factor(week)175   42.78728    9.19261   4.6545 3.276e-06 ***
factor(week)176    0.57412    8.85051   0.0649 0.9482798
factor(week)177    1.69571    8.84645   0.1917 0.8479939
factor(week)178  605.97367    8.96911  67.5623 < 2.2e-16 ***
factor(week)179   88.15770    8.96297   9.8358 < 2.2e-16 ***
factor(week)180   -1.35180    9.25474  -0.1461 0.8838715
factor(week)181   -0.17639    8.87241  -0.0199 0.9841389
factor(week)182   -1.94552    8.84709  -0.2199 0.8259480
factor(week)183   -1.51093    8.84415  -0.1708 0.8643527
factor(week)184   -5.43075    8.94902  -0.6069 0.5439571
factor(week)185  -11.21032    8.89186  -1.2607 0.2074230
factor(week)186   29.82725    9.27664   3.2153 0.0013059 **
factor(week)187    0.36542    8.87323   0.0412 0.9671514
factor(week)188   -4.54101   10.68021  -0.4252 0.6707119
```

```
factor(week)189    3.54384   21.84817   0.1622 0.8711483
factor(week)190  595.46273   28.16308  21.1434 < 2.2e-16 ***
factor(week)191   15.22467   39.26442   0.3877 0.6982089
factor(week)192   26.31356   39.25993   0.6702 0.5027157
factor(week)193   30.64547   55.17763   0.5554 0.5786320
factor(week)194   12.24523   55.09894   0.2222 0.8241296
factor(week)195    6.89516   55.09818   0.1251 0.9004120
factor(week)196    7.74220   55.10050   0.1405 0.8882586
factor(week)197   24.94582   55.11632   0.4526 0.6508413
factor(week)198    7.84579   55.11478   0.1424 0.8868027
factor(week)199   11.51450   55.17848   0.2087 0.8347030
factor(week)200   -0.44033   55.27752  -0.0080 0.9936444
factor(week)204   33.72785   55.11134   0.6120 0.5405509
factor(week)205   -5.25381   55.15489  -0.0953 0.9241131
factor(week)207   16.68744   55.15508   0.3026 0.7622334
factor(week)210  -13.58927   55.16268  -0.2463 0.8054156
factor(week)212   -1.55629   55.11374  -0.0282 0.9774729
factor(week)229   33.20245   55.13745   0.6022 0.5470664
factor(week)255  -51.24975   55.18573  -0.9287 0.3530717
factor(week)256  -47.24975   55.18573  -0.8562 0.3919041
factor(week)277  -23.63995   39.33382  -0.6010 0.5478439
factor(week)278 1193.55051   39.31404  30.3594 < 2.2e-16 ***
factor(week)279  -64.43973   39.26357  -1.6412 0.1007757
factor(week)280  -13.69905   39.37157  -0.3479 0.7278883
factor(week)281  -38.89045   32.35657  -1.2019 0.2294089
factor(week)282 1193.55051   39.31404  30.3594 < 2.2e-16 ***
factor(week)283  -64.43973   39.26357  -1.6412 0.1007757
factor(week)284  -51.75391   39.24291  -1.3188 0.1872538
factor(week)285  -45.24087   39.24288  -1.1528 0.2489940
factor(week)286  -25.86934   39.24315  -0.6592 0.5097736
factor(week)287  611.60041   39.44279  15.5060 < 2.2e-16 ***
factor(week)288  -66.15268   39.25569  -1.6852 0.0919766 .
factor(week)289  -59.03423   55.09241  -1.0715 0.2839405
factor(week)298  -37.33413   55.17262  -0.6767 0.4986207
factor(week)299  -17.04501   32.39153  -0.5262 0.5987448
factor(week)300   -7.41867   55.15846  -0.1345 0.8930111
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Total Sum of Squares:    253460000
Residual Sum of Squares: 42950000
R-Squared:      0.83055
Adj. R-Squared: 0.82691
F-statistic: 314.403 on 225 and 14433 DF, p-value: < 2.22e-16
```

2.7) Build a regression results table

Summarize your model estimates in a table. For now, we are only interested in the estimates for `price` and `promoflag`.

When displayed, the table should have 5 rows and 7 columns (including row/column labels), as follows:

	model1L	model2L	model3L	model1P	model2P	model3P

	model1L	model2L	model3L	model1P	model2P	model3P
price	coef1L	coef2L	coef3L	coef1P	coef2P	coef3P
	sd1L	sd2L	sd3L	sd1P	sd2P	sd3P
promoflag	coef1L	coef2L	coef3L	coef1P	coef2P	coef3P
	sd1L	sd2L	sd3L	sd1P	sd2P	sd3P

Where, for example, coef1L in the price row would be replace with the price parameter from model1L. Similarly, sd1L would contain the standard error for the price coefficient in model1L. And so on…

There are many ways to accomplish this task. A simple approach is to build (and print) a dataframe with the structure above. We will use the `kable()` function in the `knit` package to print the dataframe in R markdown. Below I provide a function to construct a dataframe populated with selected results.

The function `results_table1(model_list, variable_list)` takes two arguments:

`model_list` = list of regression models (output of `lm()` or `plm()`) to include

`variable_list` = list of regressor names to include

```
# results_table1(model_list, variable_list)
#     function to extract key regression results from an estimated model
#     model_list = set of regression object (e.g., output of lm()) names
#                  e.g. model_list = c("model1L","model2L")
#                  Note use of quotes!
#     variable_list = list of variable/parameter names to extract from results
#                  e.g. variable_list = c("price","promoflag")
#  returns lists of parameters, std errs, etc.
results_table1 = function(model_list, vars) {
  # build leftmost column of results table
  outrec = c()
  for (j in 1:length(vars)) {
    outrec = c(outrec,sprintf("%s",vars[j]))
    outrec = c(outrec,"")
  }
  outrec = c(outrec,"R^2")
  outrec = c(outrec,"Observations")
  outdf = as.data.frame(outrec)
  # process each model
  for (i in 1:length(model_list)) {
    # extract estimates for this model
    mod = eval(parse(text=model_list[i]))
    estimates = summary(mod)$coefficients[vars,"Estimate"]
    ses = summary(mod)$coefficients[vars,"Std. Error"]
    pvals = summary(mod)$coefficients[vars,"Pr(>|t|)"]
    # process each parameter of interest
    outrec = c()
    for (j in 1:length(vars)) {
      # set significance stars
      star = ""
      if (pvals[j] <= .05) {star = "*"}
      if (pvals[j] <= .01) {star = "**"}
      if (pvals[j] <= .001) {star = "***"}
```

```r
      # output estimate and std err
      outrec = c(outrec,sprintf("%.4f%s",estimates[j],star))
      outrec = c(outrec,sprintf("(%.4f)",ses[j]))
    }
    # add R^2, # of observations to output
    outrec = c(outrec,sprintf("%.4f",summary(mod)$r.squared[1]))
    outrec = c(outrec,sprintf("%d",nobs(mod)))
    outdf = cbind(outdf,outrec)
  }
  # set column names to model names
  names(outdf) = c("",model_list)
  outdf
}
```

Now here is an example of how to build a sample results table for models 1L and 2L, including only the `price` coefficient:

```r
# uncomment the lines below when you finish the tasks above
model_list = c("model1L", "model2L")
vars = c("price")
outdf = results_table1(model_list, vars)
library(knitr)
```

```
Warning: package 'knitr' was built under R version 3.4.3
```

```r
kable(outdf,align='c')
```

	model1L	model2L
price	-68.6215***	-68.5415***
	(1.3364)	(1.3375)
R^2	0.1588	0.1589
Observations	14744	14744

Using the example above, generate a results table for all six models (1/2/3L, 1/2/3P) and for both the `price` and `promoflag` parameters.

```r
model_list = c("model1L", "model2L", "model3L","model1P","model2P","model3P")
vars = c("price","promoflag")
outdf = results_table1(model_list, vars)
library(knitr)
kable(outdf,align='c')
```

	model1L	model2L	model3L	model1P	model2P	model3P
price	-68.6215***	-68.5415***	-17.3808***	-76.4248***	-76.3930***	-22.4991***
	(1.3364)	(1.3375)	(1.2545)	(1.3877)	(1.3882)	(1.4920)
promoflag	22.3790***	21.5989***	-7.2057*	21.8063***	21.3391***	0.2191
	(2.6354)	(2.6873)	(3.0677)	(2.5694)	(2.6212)	(2.8109)
R^2	0.1588	0.1589	0.7959	0.1796	0.1796	0.8305
Observations	14744	14744	14744	14744	14744	14744

Discussion questions

1) Examine your results table. What patterns do you notice? Comparing to the baseline model, `model1L`, adding which controls (time trends, time fixed effects, store fixed effects) leads to the greatest change in the price and promotion parameter estimates?

Models 1 & 2 across the lm/plm models are consistent in their coefficient predictions for price and promo (not exact but reasonable). However, the two methods each diverge in model 3 compared to the other models present. Compared to the baseline model adding the time fixed effects leads to the greatest change in the price and promotion parameter estimates.

2) What does this suggest about sources of omitted variable bias? E.g., are omitted factors more likely associated with cross-sectional units (stores) or time periods?

This suggessts that the source of omitted variable bias is more likely associated with time periods as the inclusion of the time fixed effects lead to the greatest change in the price and promotion parameter estimates.

3) Which estimate would you report as your "best" estimate of the demand response to price? Why?

Model 3P employs the strongest controls, and is therefore the most conservative estimate. One could also make an argument for model 3L on the basis of parsimony (and the fact that the 95% confidence intervals of price for models 3L and 3P overlaps). In the end, the important point is the necessity to control for time fixed effects in this context. Otherwise, it appears we overestimate the effect of price on sales. [Accept 3L or 3P]

3) Prediction tasks

3.1) Predict baseline expected sales using `model3L`

Using the `predict()` function, predict baseline expected sales using `model3L`. That is, use `model3L` estimates to predict expected sales for each in-sample observation. Name the result `model3L.yhat1`.

```
model3L.yhat1 = predict(model3L)
```

3.1.1) Compute (and print out) aggregate (total) expected sales

```
print(sum(model3L.yhat1))
```

```
[1] 1197463
```

3.1.2) Predict baseline expected sales using `model3L` using matrix algebra.

Using the matrix algebra, predict baseline expected sales using `model3L`. That is, use `model3L` estimates to predict expected sales for each in-sample observation. Name the result `model3L.yhat1b`. Demonstrate the predicted values are the same as in 3.1 above, using the function call:

`all.equal(model3L.yhat1,model3L.yhat1b,check.names=FALSE)`

```
# note two ways to get the right X matrix
X1 = model.matrix(~price + promoflag + factor(week), data=DF1)
X2 = model.matrix(model3L)
all.equal(X1,X2)
```

```
[1] TRUE
```

```r
model3L.yhat1b = as.numeric(X1 %*% model3L$coefficients)
all.equal(model3L.yhat1,model3L.yhat1b,check.names=FALSE)
```

```
[1] TRUE
```

3.2) Predict baseline expected sales using `model3L`, with prices increased by 1%

Use `model3L` estimates to predict expected sales for each in-sample observation, assuming prices have increased by 1%. Name the result `model3L.yhat2`.

```r
# method 1: using predict
DF2 = DF1
DF2$price = 1.01*DF2$price
model3L.yhat2 = predict(model3L,newdata=DF2)

# method 2: matrix algebra
X = model.matrix(model3L)
X[,"price"] = 1.01*X[,"price"]
model3L.yhat2b = as.numeric(X %*% model3L$coefficients)

all.equal(model3L.yhat2,model3L.yhat2b,check.names=FALSE)
```

```
[1] TRUE
```

3.2.1) Compute (and print out) aggregate (total) expected sales.

Next, compute (and print out) the percentage change in aggregate sales in response to a 1% increase in price

```r
t1 = sum(model3L.yhat1)
t2 = sum(model3L.yhat2)
print(t2)
```

```
[1] 1176031
```

```r
100*(t2-t1)/t1
```

```
[1] -1.789764
```

3.3) Predict baseline expected sales using `model3P`

NOTE:This is "bonus" material You do NOT need to answer any questions - read for your interest only

When applied to a `plm()` regression model, the `predict()` function does not work as expected. For technical reasons that I will gloss over, the output of `predict()` on a `plm()` object does *not* include the cross-sectional fixed effect. In most circumstances, we want productions that include the individual fixed effect.

Below I demonstrate how to do prediction from `plm()` that includes the estimated individual fixed effects.

```r
# NOTE: uncomment lines beginning with ## below to see results when you get to this point

# extract individual fixed effects from plm, put in dataframe
# first column contains individual id's - named store to merge with DF1
fe_DF = data.frame(store = names(fixef(model3P)), fe = fixef(model3P))

# match fixed effect (by store id)
```

```r
merged_DF = merge(DF1,fe_DF, by="store")

# full prediction using predict()
model3P.yhat1 = as.numeric(predict(model3P) + merged_DF$fe)
head(model3P.yhat1)
```

```
[1] 160.2977 164.9925 176.1394 274.0202 263.5316 187.3538
```

```r
# full prediction using matrix algebra
X = model.matrix(model3P)
model3P.yhat2 = as.numeric(merged_DF$fe + X %*% model3P$coefficients)
head(model3P.yhat2)
```

```
[1] 160.2977 164.9925 176.1394 274.0202 263.5316 187.3538
```

```r
# show the two methods are equivalent
all.equal(model3P.yhat1,model3P.yhat2,check.names=FALSE)
```

```
[1] TRUE
```