Which Variables help in predicting supermarket revenue? Evidence from Chicago

Introduction

Location choice is critical to retail organization, due to its effect on supermarket success (Clarkson, Clarke-Hill, and Robinson 1996). To choose the outlet location, previous studies have suggested that the location of supermarkets largely depends on the local demographic profile (Baviera-Puig, Buitrago-Vera, and Escriba-Perez 2016). Our study considers 45 demographic variables and tries to answer the research question: what demographic variables are important for predicting the revenue of supermarkets? We examine which variables have the biggest impact through an elastic net. This model is chosen because it has shown to work better than other regularized regressions, such as the least absolute shrinkage method (LASSO), when several variables are highly correlated (Zou and Hastie 2005). This is the case in our dataset. We assess the best parameters for the model using k-fold cross validation to further prevent overfitting (Friedman, Hastie, and Tibshirani 2001). We find that the % of households with children under 9 years old, % unemployed, and % of households with a mortgage matter most.

Data

Our research works with a dataset of 45 demographic variables related to 77 supermarkets located around the Chicago area from the year of 1996. We define demographic data as data that reflects a profile of the customers; examples of such data included in our data include age, sex, income level, race, employment, homeownership, and level of education. The variables are all continuous. A full description of each variable included can be found in table A of the appendix, and table B includes the summary statistics. Past studies have indicated that these variables likely affect the supermarket turnover. For example, the variable of income has a relation to food consumption in supermarkets (Jones 1997); the variable of gender may influence the choice of supermarket (Beynon, Moutinho, and Veloutsou 2010); and the variable of race composition may have an impact on supermarket location(Lamichhane et al. 2013). Our response variable is yearly total turnover ($). The demographic data were derived from the U.S. government's (1990) census for the Chicago metropolitan area. We scale both the independent and dependent variables as follows:

$$\tilde{\mathbf{X}} = \frac{\mathbf{X} - \mu}{\sqrt{\frac{\sum \mathbf{X}^2}{n-1}}}.$$

Where $\tilde{X}$ is the scaled variable, X is the unscaled variable, μ is the mean of $\boldsymbol{X}$, and n denotes the sample size. This scaling is done to ensure a fair interpretation of the model coefficients, as the range of each variable is different. Table C of the appendix shows that several variables are highly correlated. It is to be expected that characteristics such as income, home ownership, age etc. are to a large extent correlated.

Method

In short, our method is using an elastic net, which deals with overfitting by using a weighted average of the penalty terms applied in the LASSO and Ridge regression. Before we explain the elastic net, let's consider the standard linear regression model:

$$\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}.$$

In which $\hat{\boldsymbol{y}}$ is an n x 1 column vector of the predicted response variable, $\mathbf{X}$ is an n x (p+1) matrix with n observations of p predictor variables and the intercept. $\hat{\boldsymbol{\beta}}$ is an (1 + p) x 1 column vector of the estimated coefficients for true parameter $\boldsymbol{\beta}$ of the intercept and p predictor variables. The standard method for finding the optimal $\boldsymbol{\beta}$ is ordinary least squared (OLS that minimizes the sum of squared error between the predicted and observed response variables $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$. Thus the following loss function is minimized:

$$L(\boldsymbol{\beta}) = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}).$$

The problem with using this method is that it is prone to overfitting (Friedman, Hastie, and Tibshirani 2001). This is because when applied to a training set, it often overestimates the effect of certain variables. To solve this problem, regularized regression methods can be used. These apply penalty terms to the above loss function in order to shrink the $\boldsymbol{\beta}$. Elastic net is a regularized regression that combines two penalty terms:

$$L_1(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{p} |\beta_i|, \qquad L_2(\boldsymbol{\beta}) = \lambda \boldsymbol{\beta}^T\boldsymbol{\beta}.$$

In which L_1, L_2 are the two penalty terms, and λ is a constant parameter that determines their size. The L_1 is the sum of the absolute value of the coefficients, and is used in the so-called LASSO method (Tibshirani 1996). When this penalty term is used, coefficients are continuously reduced, or removed completely. But using just this penalty term has several downsides; it performs worse when $p \geq n$, and when several variables are highly correlated (Zou and Hastie 2005) . Zhou and Hastie (2005) show that in these circumstances, the elastic net performs better than the Lasso, by adding a second penalty term, L_2. This is the same penalty used in a Ridge regression (Hoerl and Kennard 1970). Elastic net then determines the weight between the two penalty terms, using the constant parameter α. Given the high correlation between several of the variables in our dataset, it seems appropriate to use the elastic net. This gives the following loss function:

$$L(\boldsymbol{\beta}) = (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\alpha \sum_{i=1}^{p} |\beta_i| + \lambda(1-\alpha)\boldsymbol{\beta}^T\boldsymbol{\beta}.$$

In order to find the $\boldsymbol{\beta}$ at which $L(\boldsymbol{\beta})$ is minimized, we use the majorize-minimization algorithm (MM). We use the following majorization function:

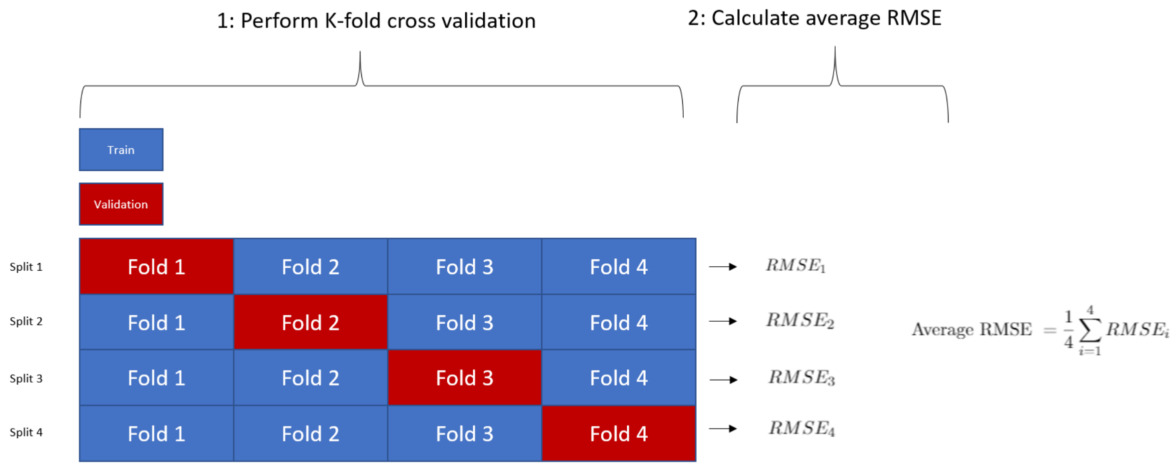
$$L(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T(\mathbf{A})\boldsymbol{\beta} - n^{-1}\boldsymbol{\beta}^T\mathbf{X}^T\boldsymbol{y} + c,$$

$$\mathbf{A} = n^{-1}\mathbf{X}^T\mathbf{X} + \lambda(1+\alpha)I + \lambda\alpha\mathbf{D}, \qquad \mathbf{D} = \begin{bmatrix} \frac{1}{max(\beta_1,\epsilon)} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{1}{max(\beta_p,\epsilon)} \end{bmatrix}, \qquad c = \frac{1}{2n}\boldsymbol{y}^T\boldsymbol{y} + \frac{1}{2}\lambda\alpha\sum_{i=1}^{p}|\beta_i|.$$

We then find the $\boldsymbol{\beta}$ for which this function is minimized through stepwise updating of $\boldsymbol{\beta}$. This happens by solving the following: $\hat{\boldsymbol{\beta}_k} = (\mathbf{A})^{-1}n^{-1}\mathbf{X}^T\boldsymbol{y}$. Here, $\hat{\boldsymbol{\beta}_k}$ is the estimated $\boldsymbol{\beta}$ at step k. This stepwise updating continues until the next set of $\boldsymbol{\beta}$ does not improve the loss function by more than ϵ. To find the λ and α for our elastic net, we use K-fold cross validation. In this method, the dataset is split into K random samples(folds). One of the folds is picked as the validation set, and the $\boldsymbol{\beta}$ are determined based on the remaining K-1 folds. This then happens K times, until each of the folds is used as a validation set. For each iteration, we calculate the root mean squared error (RMSE), and then the average RMSE across the K folds used for validation.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{\boldsymbol{y}} - \boldsymbol{y})^2}, \qquad RM\bar{S}E = \frac{1}{K}\sum_{i=1}^{K} RMSE_i.$$

In which $RMSE_i$ is the RMSE of the validation on sample i. Our metric for picking the λ and α is the lowest average RMSE. The process of finding this metric is visualized below for K = 4.



Using K-fold cross validation to pick the ideal λ, α reduces the likelihood of overfitting, since the parameters are not just based on one random sample, but on multiple, reducing the likelihood that features from one sample dominate when predicting. We fold our sample 10 times - this allows us to use a lot of different folds, without becoming computationally too expensive. We use RMSE to measure the error between our prediction and the actual values. The squaring of these errors ensures that larger errors contribute more to the loss function, which decreases the chance of our model making large mistakes. When searching for λ, α, we evaluate all combinations between two sets. The first set containing all possible α values is defined as follows: $\alpha = \{0.1, 0.2, ..., 1\}$. The set containing the λ values is $\lambda = \{10^{x_1}, 10^{x_2}, ..., 10^{x_{50}}\}$ with x increasing from -2 to 10 in 50 steps.

Results

By minimizing the RMSE of all α and λ combinations, we find that an α of 0.2 and λ of 0.1 produces the best fitted model. This indicates that in our problem more emphasis should be put on the L_2 penalty. This result is consistent with the findings in (Marquardt and Snee 1975), which found that in problems with highly correlated explanatory variables ridge regression performs best.

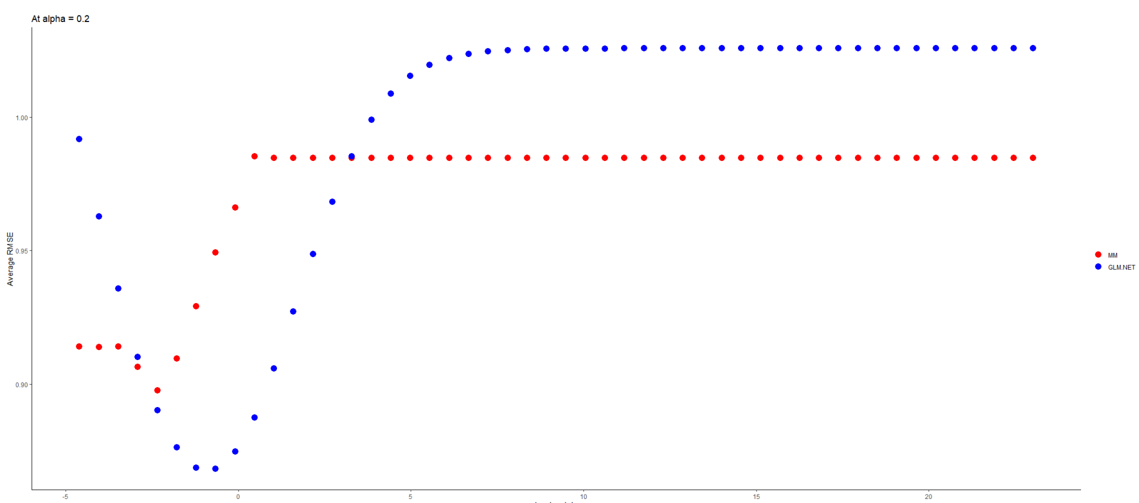
To answer our research question on what variables are most important for the prediction of supermarket turnover, we analyse the estimated coefficients obtained by training on the scaled dataset. These are found in 1. In this table only the coefficients with an absolute value higher than 0.01 are displayed, because any coefficients below this threshold are at least an order of magnitude smaller and thus have extremely little effect on our dependent variable. The coefficients represent change in our standardised dependent variable when the independent variable changes with one standard deviation. Thus the absolute size of the coefficient can be interpreted as the contribution of the variable for the prediction. From table 1 we find that the three most influential variables are the % of households with children under nine years old, % unemployed and percentage of households with a mortgage. Here the first two are positive in their influence on supermarket turnover and the last one is negative.

To verify if our implementation of elastic net regression and hyperparameter search using 10-fold cross-validation is correct we compare our implementation's estimates to the estimates of an established library, called glmnet (Friedman, Hastie, and Tibshirani 2010) The library finds betas that are equal to our implementation when rounded to 2 digits. As seen the betas are very similar to each other. Furthermore for the optimal α (0.2) we plot the RMSE against the $\log(\lambda)$. Here we find the graphs differ in two ways, first the

Table 1: Our Est. Coefficients

Predictor Variable	Est. Coefficient
% With Mortgage	−0.17
% of Women with children under 5	−0.16
% of Working Women	−0.16
Population density	−0.09
% of Households with 5, 6 or 7 people	−0.08
% of Households with more than 5 people	−0.08
% of Avid Shoppers	−0.05
% of White Shoppers	−0.02
% of Shopping Strangers	−0.01
% of population with income under 15, 000$	0.02
% of Households with more than 2 people	0.04
% of Households with Value over 200, 000$	0.07
% of Households with 1 person	0.09
% of Households with 3 or 4 persons	0.12
% of Households with Value over 150, 000	0.14
% of Hurried Shoppers	0.14
% of Unemployed	0.23
% of Population under age 9	0.29

RMSE for our implementation converges to a different value than glmnet. Secondly the rate of convergence is different. Both of these differences are likely due to the fact that, dependent on the initial beta chosen, the MM algorithm is not guaranteed to converge to a global minimum in the loss. This also explains why are minimum RMSE value is slightly higher than that of glmnet.



\

Conclusion

In this report we analysed what variables are important for predicting the turnover of supermarkets. To do so we used an elastic net regression which was optimized using the MM algorithm. We find that the three most important variables for predicting turnover are the percentage of households with children under nine years old, percentage unemployed and percentage of households with a mortgage. Thus we can advice supermarkets to include these demographic variables in their location strategy. \ A limitation of this research is that the data was only gathered from Chicago and is quite old. It is possible that our findings are very location dependent and do not extend to the rest of the United States or world, or that cultural changes caused the effects to change. For further research including more data from a wider geological area or gathering new data might create new insights into the present day influence of demographic variables on supermarket turnover.

References

Baviera-Puig, Amparo, Juan Buitrago-Vera, and Carmen Escriba-Perez. 2016. "Geomarketing Models in Supermarket Location Strategies." *Journal of Business Economics and Management* 17 (6): 1205–21.

Beynon, Malcolm J, Luiz Moutinho, and Cleopatra Veloutsou. 2010. "Gender Differences in Supermarket Choice." *European Journal of Marketing*.

Clarkson, Richard M, Colin M Clarke-Hill, and Terry Robinson. 1996. "UK Supermarket Location Assessment." *International Journal of Retail & Distribution Management*.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. 10. Springer series in statistics New York.

———. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. http://www.jstatsoft.org/v33/i01/.

Hoerl, Arthur E, and Robert W Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems." *Technometrics* 12 (1): 55–67.

Jones, Eugene. 1997. "An Analysis of Consumer Food Shopping Behavior Using Supermarket Scanner Data: Differences by Income and Location." *American Journal of Agricultural Economics* 79 (5): 1437–43.

Lamichhane, Archana P, Joshua Warren, Robin Puett, Dwayne E Porter, Matteo Bottai, Elizabeth J Mayer-Davis, and Angela D Liese. 2013. "Spatial Patterning of Supermarkets and Fast Food Outlets with Respect to Neighborhood Characteristics." *Health & Place* 23: 157–64.

Marquardt, Donald W, and Ronald D Snee. 1975. "Ridge Regression in Practice." *The American Statistician* 29 (1): 3–20.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267–88.

Zou, Hui, and Trevor Hastie. 2005. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (2): 301–20.

Appendix

Table A: Description of variables

Variable Name	Description
unemp	% of Unemployed
wrkch5	% of working women with children under 5
wrkch17	% of working women with children 6 - 17
nwrkch5	% of non-working women with children under 5
nwrkch17	% of non-working women with children 6 - 17
wrkch	% of working women with children
nwrkch	% of non-working women with children
wrkwch	% of working women with children under 5
wrkwnch	% of working women with no children
telephn	% of households with telephones
mortgage	% of households with mortgages
nwhite	% of population that is non-white
poverty	% of population with income under $15,000
shopcons	% of Constrained Shoppers
shophurr	% of Hurried Shoppers
shopavid	% of Avid Shoppers
shopstr	% of Shopping Stranges
shopunft	% of Unfettered Shoppers
shopbird	% of Shopper Birds
shopindx	Ability to Shop (Car and Single Family House)
shpindx	Ability to Shop (Car and Single Family House)
store	Store identification number
city	City of supermarket
Zip	zip code
grocery__sum	Total turnover in one year of groceries(dollar)
groccoup__sum	Total of redeemed grocery coupos ()
age9	% population under age 9
age60	% population over age 60
ethnic	% Blacks and Hispanics
educ	% College Graduates
nocar	% With No Vehicles
income	Log of median income
incsigma	Standard deviation of income distribution(approximated)
hsizeavg	Average Household Size
hsize1	% of households with 1 person
hsize2	% of households with 2 persons
hsize34	% of households with 3 or 4 persons
hsize567	% of households with 5 ore more persons
hh3plus	% of households with 3 or more persons
hh4plus	% of households with 4 or more persons
hhsingle	% Detached Houses
hhlarge	% of households with 5 or more persons
workwom	% Working Women with full-time jobs
sinhouse	% of households with 1 person
density	Trading Area in Sq Miles per Capita
hval150	% of Households with Value over $150,000
hval200	% of Households with Value over $200,000
hvalmean	Mean Household Value(Approximated)
single	% of Singles
retired	% of Retired

Table B: Summary Statistics

Predictor variable	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
GROCERY_sum	77	7,341,015.000	2,341,073.000	1,423,582.000	5,778,987.000	9,022,599.000	13,165,586.000
AGE9	77	0.138	0.025	0.046	0.121	0.151	0.193
AGE60	77	0.173	0.063	0.058	0.122	0.214	0.307
ETHNIC	77	0.160	0.193	0.024	0.044	0.188	0.996
EDUC	77	0.227	0.114	0.050	0.146	0.284	0.528
NOCAR	77	0.113	0.132	0.012	0.025	0.144	0.551
INCOME	77	10.616	0.293	9.867	10.414	10.797	11.236
INCSIGMA	77	24,840.880	2,295.236	20,359.560	23,488.270	26,458.280	30,276.640
HSIZEAVG	77	2.665	0.263	1.554	2.543	2.790	3.309
HSIZE1	77	0.245	0.083	0.122	0.200	0.269	0.614
HSIZE2	77	0.309	0.031	0.219	0.290	0.333	0.369
HSIZE34	77	0.330	0.060	0.092	0.306	0.367	0.446
HSIZE567	77	0.116	0.031	0.014	0.098	0.132	0.216
HH3PLUS	77	0.446	0.083	0.106	0.405	0.490	0.650
HH4PLUS	77	0.274	0.063	0.041	0.241	0.305	0.443
HHSINGLE	77	0.245	0.083	0.122	0.200	0.269	0.614
HHLARGE	77	0.116	0.031	0.014	0.098	0.132	0.216
WORKWOM	77	0.358	0.053	0.244	0.312	0.402	0.472
SINHOUSE	77	0.548	0.216	0.017	0.517	0.706	0.822
DENSITY	77	0.001	0.001	0.0001	0.0004	0.001	0.005
HVAL150	77	0.349	0.246	0.003	0.123	0.534	0.917
HVAL200	77	0.186	0.186	0.001	0.043	0.268	0.781
HVALMEAN	77	147.907	47.534	64.348	108.924	179.072	267.390
SINGLE	77	0.280	0.068	0.203	0.242	0.286	0.593
RETIRED	77	0.150	0.051	0.056	0.109	0.188	0.236
UNEMP	77	0.182	0.023	0.142	0.166	0.195	0.245
WRKCH5	77	0.056	0.020	0.024	0.041	0.070	0.118
WRKCH17	77	0.124	0.029	0.041	0.103	0.144	0.198
NWRKCH5	77	0.084	0.028	0.030	0.064	0.101	0.169
NWRKCH17	77	0.070	0.021	0.018	0.059	0.082	0.122
WRKCH	77	0.180	0.044	0.071	0.149	0.214	0.293
NWRKCH	77	0.154	0.043	0.048	0.123	0.183	0.250
WRKWCH	77	0.055	0.020	0.024	0.041	0.069	0.115
WRKWNCH	77	0.258	0.044	0.157	0.227	0.282	0.460
TELEPHN	77	0.977	0.029	0.839	0.976	0.993	0.998
MORTGAGE	77	0.710	0.147	0.443	0.617	0.826	0.960
NWHITE	77	0.204	0.194	0.035	0.091	0.205	0.995
POVERTY	77	0.058	0.045	0.014	0.027	0.076	0.213
SHPCONS	77	0.082	0.062	0.019	0.037	0.115	0.279
SHPHURR	77	0.153	0.059	0.026	0.110	0.191	0.286
SHPAVID	77	0.189	0.043	0.061	0.161	0.220	0.310
SHPKSTR	77	0.284	0.066	0.184	0.232	0.330	0.558
SHPUNFT	77	0.246	0.055	0.145	0.197	0.291	0.391
SHPBIRD	77	0.046	0.025	0.004	0.025	0.064	0.105
SHOPINDX	77	0.736	0.246	0.00000	0.730	0.890	0.986

Table C: Variables that have an absolute correlation of more than 0.7

	Predictor Variable 1	Predictor Variable 2	Correlation
1	HSIZE1	HSIZE34	−0.96
2	HSIZEAVG	HSIZE1	−0.91
3	HH4PLUS	HHSINGLE	−0.89
4	MORTGAGE	SHPBIRD	−0.87
5	SINHOUSE	SINGLE	−0.86
6	WORKWOM	RETIRED	−0.86
7	NOCAR	INCOME	−0.84
8	SHPCONS	SHPHURR	−0.81
9	HHLARGE	WRKWNCH	−0.80
10	HHSINGLE	SINHOUSE	−0.80
11	HSIZE2	UNEMP	−0.78
12	TELEPHN	NWHITE	−0.76
13	RETIRED	WRKCH17	−0.73
14	SINGLE	TELEPHN	−0.73
15	UNEMP	TELEPHN	−0.73
16	ETHNIC	INCOME	−0.72
17	AGE9	AGE60	−0.70
18	EDUC	INCSIGMA	0.72
19	WRKCH5	NWRKCH5	0.75
20	WRKCH17	NWRKCH5	0.75
21	WRKCH	NWRKCH	0.75
22	INCSIGMA	HVAL150	0.78
23	SHPHURR	SHOPINDX	0.78
24	NWRKCH	SHPHURR	0.79
25	INCOME	INCSIGMA	0.80
26	HSIZE567	HH3PLUS	0.84
27	NWRKCH17	NWRKCH	0.84
28	NWHITE	POVERTY	0.84
29	NWRKCH5	WRKCH	0.84
30	AGE60	RETIRED	0.88
31	HVAL150	HVAL200	0.93
32	HVAL200	HVALMEAN	0.94
33	HSIZE34	HH3PLUS	0.96
34	HH3PLUS	HH4PLUS	0.99
35	POVERTY	SHPCONS	0.99

Code

```r
# first, we define several functions that we later use for the analysis

# Calculates diagonal matrix D
calc_mD = function(mBeta, p, epsilon){

  # Create the diagonal vector that will be filled with diagonal elements of D
  to_diagonalise <- vector(length=p)

  # Get diagonal elements of D (max of mBeta_i, epsilon)
  for (i in 1:p) {

    to_diagonalise[i] <- 1 / max(abs(mBeta[i]), epsilon)
  }
  # Multiply the vector containing diagonals with Identity matrix to get D
  return(to_diagonalise * diag(p))
}


# Calculates the loss function for the elastic
elasticLoss = function(mBeta, mA, mXtY, mYtY, alpha, lambda, n){

  # Calculate transposed matrix of Beta's
  transposed_mBeta = t(mBeta)

  # Compute constant
  constant <- (1/2*n) %*% mYtY + (1/2) * alpha * lambda *sum(abs(mBeta))

  # Return loss function
  return(1/2 * (transposed_mBeta%*%mA%*%mBeta)-(1/n)*(transposed_mBeta%*%mXtY) + constant)
}

# Calculates a often, used variable in subsequent computations
# This indicates the division between the lasso (a = 1) and ridge method (a = 0)
calc_typeNet = function(lambda, alpha, mD,p){
  lambda *(1 - alpha)*diag(p) + (lambda * alpha  * mD)
  }

# Calculates the root mean squared error (RMSE)
calcRMSE = function(X, y, est_beta, n){
  error <- y - X %*% est_beta
  rsme <- sqrt((1/n) * (t(error)%*%error))
}

# calculates estimate for the elastic net, given lambda and alpha, using MM algorithm
ElasticNetEst = function(mX, mY, beta_init, lambda, alpha, tolerance, epsilon, max_iter = 100000){

  # Set iterations and improvement
  k <- 1
  improvement <- 0

  # Define number of predictor variables and datapoints
```

```r
  n <- nrow(mX)
  p <- ncol(mX)

  # Pre-compute constants
  mXtX <- crossprod(mX,mX)
  mXtY <- crossprod(mX,mY)
  mYtY <- crossprod(mY,mY)
  scaled_I <- lambda * (1-alpha) * diag(p)

  # get initial values for mD, mA, and Beta's
  mD <- calc_mD(beta_init, p, epsilon)
  typeNetInit <- calc_typeNet(lambda, alpha, mD, p)
  mA <- 1/n * mXtX + typeNetInit
  Beta_prev <- beta_init

  # start stepwise improvement of Beta's
  while (k == 1 | k < max_iter && (improvement > tolerance)) {

    # Increase number steps k
    k <- k + 1

    # calculate mD, MA
    mD <- calc_mD(Beta_prev, p, epsilon)
    typeNet <- calc_typeNet(lambda, alpha, mD, p)
    mA <- ((1/n) * mXtX) + typeNet

    # get new set of Beta's
    Beta_current <- solve(mA, 1/n * mXtY)

    # calculate loss function for previous, current Beta's - and the improvement
    loss_current <- elasticLoss(Beta_current,mA, mXtY, mYtY, alpha, lambda, n)
    loss_prev <- elasticLoss(Beta_prev, mA, mXtY, mYtY, alpha, lambda, n)
    improvement <- (loss_prev - loss_current)/loss_prev

    # set the previous beta's to current beta's for next step
    Beta_prev <- Beta_current

  }

  # return est. Beta's
  return(Beta_current)
}

# k-fold crossvalidation of the elastic net
crossValidation = function (df, k, beta_init, lambda, alpha, tolerance, folds) {

  # initial value for total rmse, min and max
  total_rmse <- 0
  min_rsme <- Inf
  max_rsme <- 0

  # save the n of observations
  n <- nrow(df)
```

```r
  #Perform k fold cross validation
  for(i in 1:length(folds)){

    #Split the data according to the folds
    test = df[folds[[i]],]
    train = df[-folds[[i]],]

    # define train and test set for y and x
    y_train <- as.matrix(train[,1])
    X_train <- as.matrix(train[,-1])
    y_test <- as.matrix(test[,1])
    X_test <- as.matrix(test[,-1])

    # get est. Beta's from the elastic net
    Beta_est <- ElasticNetEst(X_train, y_train, beta_init, lambda, alpha, tolerance, epsilon)

    # define rmse for this set of lambda, alpha
    rmse <- calcRMSE(X_test, y_test, as.matrix(Beta_est), nrow(X_test))

    # add current rms to total, to tlater take average
    total_rmse <- total_rmse + rmse

    # save min and max of rmse
    if(rmse > max_rsme){
      max_rsme = rmse
    }else if (rmse < min_rsme){
      min_rsme = rmse
    }

  }
  # calculate the avg. rmse across the folds
  avg_rmse <- total_rmse / length(folds)

  # returns results
  result = list(alpha = alpha,
                lambda = lambda,
                avg_rmse = avg_rmse,
                min_rsme = min_rsme,
                max_rsme = max_rsme
  )

  return(result)

}

# search the  hyperparameters lambda, alpha that minimize rmse in k-fold
HyperSearch = function(df, k, grid, beta_init, tolerance){

  # scale both the dependent and independent
  df <- scale(df)

  # empty dataframe
  results <- data.frame(Lambda= numeric(),
```

```
                         Alpha= numeric(),
                         avg_rmse = numeric(),
                         min_rsme = numeric(),
                         max_rsme = numeric())

  # create k equally size folds
  folds = createFolds(y, k = k, list = TRUE, returnTrain = FALSE)

  # iterate over the grid
  for(i in 1:nrow(grid)){

    # get current lambda & alpha
    lambda <- as.numeric(grid[i,][1])
    alpha <- as.numeric(grid[i,][2])

    # get result of cross validation for lambda, alpha
    result_cv <- crossValidation(df, k, beta_init, lambda, alpha, tolerance, folds)

    # define row to add to dataframe with results
    result_row <- c(lambda,
                    alpha,
                    result_cv$avg_rmse,
                    result_cv$min_rsme,
                    result_cv$max_rsme)
    results[i,] <- result_row

  }

  return(results)


}

# Here, we perform the analysis used in the report, using the functions above

# load libraries
library(MASS)
library(matlib)
library(caret)
library(glmnet)
library(tidyverse)
library(reshape2)
library(stargazer)

# set seed to ensure stability of results
set.seed(0)

# load the data
load("supermarket1996.RData")

# create dataframe with dependent and independent variables
df = subset(supermarket1996, select = -c(STORE, CITY, ZIP, GROCCOUP_sum, SHPINDX) )
y = scale(as.matrix(df[,1]))
```

```r
X = scale(as.matrix(df[,-1]))

# define the initial set of beta's
Beta_init = as.matrix(runif(ncol(df)-1, min=-1, max=1))

# define the parameters
tolerance = 0.0000000000001
epsilon = 0.0000000000001

# create grid of lambda and alpha combinations
listLambda <- 10^seq(-2, 10, length.out = 50)
listAlpha <- seq(0.0,1,0.1)
paramGrid <- expand.grid(listLambda, listAlpha)

# find the results of gridsearch
search_result <- HyperSearch(df, 10, paramGrid, Beta_init, tolerance)

# set best set of parameters
best_param <- search_result[search_result$avg_rmse==min(search_result$avg_rmse),]
best_lambda <- round(best_param$Lambda,2)
best_alpha <- best_param$Alpha

# find Beta's for our estimate
BetaEst <- ElasticNetEst(X, y, Beta_init, lambda = best_lambda, alpha = best_alpha, tolerance, epsilon)

# select the top beta's in terms of absolute value
top_Beta <- data.frame(BetaEst) %>%
  filter(abs(BetaEst) > 0.01)%>%
  arrange(BetaEst)

# create overview table
stargazer(top_Beta,
          digits=2,
          summary = FALSE)

# Beta's for glm.net estimate, same param as from our grid search
result.cv.ideal <- glmnet(scale(X), scale(y), alpha = best_alpha,
                              lambda =best_lambda, nfolds = 10)
glm.net_Beta <- as.matrix(result.cv.ideal$beta)
colnames(glm.net_Beta) <- c("BetaEst")

# find top beta's for glm.net in terms of absolute value
glm.net_top_Beta <- data.frame(glm.net_Beta) %>%
  filter(abs(BetaEst) >= 0.01)%>%
  arrange(BetaEst)

# create overview table
stargazer(glm.net_top_Beta,
          digits=2,
          summary = FALSE)

# Beta's for glm.net estimate - for all values of lambda evaluated
result.cv.lambda <- cv.glmnet(scale(X), scale(y), alpha = 0,
```

```r
                       lambda =listLambda, nfolds = 10)

# compare convergence of the two methods
# set variables for our method
paramGrid_compare <- expand.grid(listLambda, best_alpha)
search_compare <- search_result[search_result$Alpha == best_alpha,]

# create dataframe for the visualization
df_compare = data.frame(lambda = log(listLambda),
                        MM = search_compare$avg_rmse,
                        GLM.NET = rev(result.cv.lambda$cvm))
df_compare = melt(df_compare, id.vars = 'lambda', variable.name = 'series')

# plot to compare convergence to glm.net
ggplot() +
  geom_point(data = df_compare,
             aes( x = lambda, y = value, col=series),
             size=4) +
  theme_minimal() +
  labs(y = "Average RMSE",
       x = "Log Lambda") +
  scale_color_manual(values=c("red", "blue")) +
  ggtitle("At alpha = 0.2") +
  theme( plot.title = element_text(color="black", size=30)) +
  theme_classic()+
  theme(legend.title = element_blank())

# create correlation matrix
corr <- cor(X)

#prepare to drop duplicates and correlations of 1
corr[lower.tri(corr,diag=TRUE)] <- NA
#drop perfect correlations
corr[corr == 1] <- NA
#turn into a 3-column table
corr <- as.data.frame(as.table(corr))
#remove the NA values from above
corr <- na.omit(corr)
# only show high correlations
large_corr <- corr %>%
  filter(abs(Freq)>0.7) %>%
  distinct(Var1, .keep_all = TRUE) %>%
  arrange(Freq)

# turn into table
stargazer(large_corr, summary = FALSE)
```