UIC Weight and Veterans' Environments Study

Accessibility of businesses: Weight and Veterans' Environments Study GIS protocol

Jones KK, Xiang W, Matthews SA, Zenk SN.

UIC Neighborhoods + Health

Overview

This protocol describes the process through which WAVES processed commercial business list data. We include sections on measure definitions, bias assessments, and deduplication. Finally, we include a section describing how we used the processed business list data to create national raster surfaces showing neighborhood environment measures.

Acknowledgements

This protocol was developed with grant support from the National Cancer Institute (R21CA195543) and the Department of Veterans Affairs (IIR 13-085), co-led by Shannon Zenk and Elizabeth Tarlov. We thank Lisa Powell, Irina Horoi, Sandy Slater for input into this protocol. Haytham Abu Zayd, Shubhayan Ukil and Abby Klemp helped to edit the protocol for dissemination.

Suggested Citation

Jones K, Xiang W, Matthews SA, Zenk SN. (2018). Accessibility of businesses: Weight and Veterans' Environments Study GIS protocol, Version 1. Retrieved from Weight and Veterans' Environments Study website: https://waves.uic.edu/.

Table of Contents

Background

This document describes the protocol followed by WAVES to construct measures of the commercial environment, particularly food stores and restaurants. Food store and restaurant data was purchased from commercial business list vendors (InfoUSA and Dun and Bradstreet) for the continental United States. This process created "grids", or raster files that identified the commercial environment within 4 different Euclidean buffers of any given 30mX30m cell within the continental US. In this way, the environmental data was distinct from the study population data, providing significant flexibility for both use and storage. The environmental data from the grids was joined to participant data using the Extract Multi Values to Points function in ArcGIS.

For additional information on this process, please see:

Jones, K. K., Zenk, S. N., Tarlov, E., Powell, L. M., Matthews, S. A., & Horoi, I. (2017). A step-by-step approach to improve data quality when using commercial business lists to characterize retail food environments. *BMC research notes*, *10*(1), 35.

Data

Definitions

In order to maintain the greatest amount of flexibility in analysis, stores and restaurants were classified into subtypes that could be combined in a variety of ways to identify store types. The basic subtypes and constructed types are shown for each category are shown below. Grids were only made for subtypes, and the constructed variables were made out of those subtypes outside of the GIS environment after the subtypes had been joined to the participant points.

Food Stores

Supermarkets and Grocery Stores

- Type 1: Less than $2million in annual sales, no chain name, less than eleven stores in the region
- Type 2: Less than $2million in annual sales, no chain name, eleven or more stores in the region
- Type 3: Less than $2million in annual sales, chain name, less than eleven stores in the region
- Type 4: Less than $2million in annual sales, chain name, eleven or more stores in the region
- Type 5: At least $2million in annual sales, no chain name, less than eleven stores in the region
- Type 6: At least $2million in annual sales, no chain name, eleven or more stores in the region
- Type 7: At least $2million in annual sales, chain name, less than eleven stores in the region
- Type 8: At least $2million in annual sales, chain name, eleven or more stores in the region
- Grocery Stores: Type 1 + Type 2
- Chain Supermarkets: Type 3 + Type 4 + Type 7 + Type 8
- All Supermarkets: Type 3 + Type 4 + Type 5 + Type 6 +Type 7 + Type 8
- AllSupGroc: Type 1 + Type 2 + Type 3 + Type 4 + Type 5 + Type 6 +Type 7 + Type 8
- Large Supermarket: Type 3 with more than 50 employees + Type 4 with more than 50 employees + Type 5 with more than 50 employees + Type 6 with more than 50 employees + Type 7 with more than 50 employees + Type 8 with more than 50 employees

  - Large Chain Supermarket: Type 3 with more than 50 employees + Type 4 with more than 50 employees + Type 7 with more than 50 employees + Type 8 with more than 50 employees
  - Small Supermarket: All Supermarket minus Large Supermarket
  - Small Chain Supermarket: Chain Supermarket minus Large Chain Supermarket

Convenience Stores

- Chain as defined by chain name list
- Non-chain
- All = chain + nonchain

Liquor Stores

- No sub-types

General Merchandise Stores

- Member: Sams Club, Costco
- Non-Member: Target, Meijer, Walmart, Kmart
- BJ's Wholesale (This got dropped because there were only 3 in the first several years and then there were something in the teens)
- All = Member + NonMember

Pharmacies

- Chain as defined by Chain name list
- Non-Chain
- All = Chain + NonChain

Physical Activity Facilities

- 1: Clubs
  - SIC Codes: 799705 799707 799708 799717 799719 799720 799721 799722 799727 799728 799798
- 2: Instructional
  - SIC Codes: 791101 791104 799902 799914 799916 799924 799932 799936 799944 799945 799950 799957 799961 799962 799965 799966 799967 799970 799978
- 3: General and Recreational Fitness
  - SIC Codes: 799101 799102 799103 799105 799107 799109 799110 799701 799949
- 4: Fitness Places
  - SIC Codes: 791105 793301 799201 799202 799702 799703 799704 799706 799711 799723 799901 799903 799908 799929 799931 799958 799969 799971 799984 799968 799994 799111 799729
- 5: Secondary Pull: Personal Trainers

  - Secondary SIC Codes: 799106
- 6: Secondary Pull: Health Fitness and Exercise Programs
  - Secondary SIC Codes: 729901 729906
- 7: YMCA
  - Defined by name
- All = Clubs + Instructional + General Recreational Fitness + Fitness Places + Secondary Pull: Personal Trainers + Secondary Pull: Health Fitness and Exercise Programs + YMCA
- All_noTrainers = Clubs + Instructional + General Recreational Fitness + Fitness Places + Secondary Pull: Health Fitness and Exercise Programs + YMCA
- All_noSecondary = Clubs + Instructional + General Recreational Fitness + Fitness Places + YMCA

Restaurants

- Component Types
  - 1: Top 5 including McDonalds, Taco Bell, Wendy’s Burger King, KFC (identified by name)
  - 2: Subway (identified by name)
  - 3: Non Top 5 National Chain (name found on chain name list) (identified by name)
  - 4: Coffeeshops: Starbucks, Dunkin Donuts, Tim Hortons (identified by name)
  - 5: Top Chain name Pizza (name found on chain name list and identified as Pizza) (identified by name)
  - 6: Chain Pizza (identified by SIC code, name not on chain name list)
  - 7: Non-chain pizza (identified by SIC code)
  - 8: Non-chain coffee (identified by SIC code)
  - 9: Not elsewhere classified
- Constructed Types
  - Top5 = Type 1
  - NonPizzaChain = Type 1 + Type 2 + Type 3
  - ChainCoffee = Type 4
  - AllCoffee = Type 4 + Type 8
  - ChainNamePizza = Type 5
  - AllChainPizza = Type 5 + Type 6
  - AllPizza = Type 5 + Type 6 + Type 7
  - NonChainFFR = Type 9
  - NonChainFFR_wPizza = Type 7 + Type 9
  - AllChain_noCoffee = Type 1 + Type 2 + Type 3 + Type 5 + Type 6
  - AllFFR_noCoffee = Type 1 + Type 2 + Type 3 + Type 5 + Type 6 + Type 7 + Type 9
  - AllFFR = Types 1 through 9

Cleaning and Measure Construction

Assessment of Purchased Data

A full assessment was made of the received data. Assessment included:

- matching of record numbers received to record numbers quoted
- matching of SIC codes received to SIC codes requested

Additional issues uncovered and addressed were as follows:

- variables were not consistently named or formatted across data years
- Codebooks were not originally provided with the dataset. We had to request specific codebooks to understand the data

Bias Assessment: Matchcode

There are biases in these data. Those biases include by neighborhood type and by urbanicity. We examine this with our matchcode assessments.

Demographics

- Are there differences in Matchcode by demographic characteristics? We examined Racial/ethnic neighborhood makeup, and Median Houshehold Income.
- Race/Ethnicity was defined following Powell (2011) as >70% Hispanic, non-Hispanic white, non-Hispanic black, or Mixed, with data from 2010 Census at the block group level. Median Household Income was evaluated in tertiles of census tracts of the entire USA for the ACS 5-year estimates 2006-2010. In addition to evaluating these separately, they were evaluated in conjunction.

Urbanicity

- Urban location was defined as within a CBSA (2010). Matchcode by urbanicity was examined alone, as well as median household income by urbanicity, race/ethnicity by urbanicity, and race/ethnicity by median household income by urbanicity.

ZIP2/ZIP5

- Locations geocoded to ZIP2 and ZIP5 were examined to determine if they represented the geographic centroids of either the 2-digit or 5-digit zip codes. They do not, but they are all in the same place.

PO Box

- Locations with PO Boxes for addresses included all matchcodes. If the PO Box was the only address available to accomplish the geocoding, it should be geocoded to ZIP centroid. The matchcodes did not match this, and the actual geocoded locations also did not match this. Therefore, it is unclear what addresses were used to geocode locations with PO Box addresses.
- Because of this, when we could legitimately follow a particular business across time and later in the dataset the business had a street address used to geocode, then the PO Box-based geocode was replaced by the street address-based geocode.

Food Store Classification

Classification of the data was done in STATA v. 13. The complete code is found in Appendix A.

The following is pseudocode for the do-file:

```
1. Standardization of variable names and reduction of variables to those necessary for classification and measure construction.
    a. Variables necessary for measure construction were:
        i. Latitude/Longitude
        ii. Store name
        iii. Total annual sales
        iv. Primary SIC code/Secondary SIC codes
        v. Address
        vi. Cityname
        vii. Statealpha (State)
        viii. Digitzipcode ( Zip Code)
2. Ensure no duplicate records by company name and full address
    a. Full addresses are necessary to not eliminate two stores belonging to the same chain in the same town
3. Classification of records:
    a. Store Types
        i. Liquor by SIC code
        ii. General Merchandise by store name
            1. Walmart neighborhood stores are not general merchandise stores, so GM classification removed
            2. “Super” stores are not treated separately
                a. There is no guarantee that the record switched from a non-“Super” to a “Super” at the same time the store did
                b. Stores converting to “Super” were found to begin carrying fresh food products prior to their designation as “Super”
        iii. Pharmacies by SIC
            1. Declassify pharmacies that are departments of general merchandise stores or supermarkets as pharmacies
                a. Name search on general merchandise stores
                b. Duplicates by address and lat/lon
        iv. Convenience stores by SIC
            1. Reclassify pharmacies (by chain name list) that have convenience store SIC codes to pharmacy
            2. Declassify chain supermarkets and general merchandise stores with convenience store SIC codes
        v. Supermarkets and Grocery stores by SIC
            1. Declassify general merchandise stores with SG SIC codes
            2. Reclassify pharmacies by store name if SG SIC codes
```

            3. Reclassify convenience stores by store name if SG SIC codes
            4. Declassify if store name = BOZZUTO
    b. Chain Status
        i. Supermarkets/Grocery Stores. For Supermarkets/Grocery stores only:
            1. Flag if sales > $2 million
            2. Flag if 11 or more stores in (census) region
            3. Flag if store name on Supermarket chain list
        ii. Convenience stores: For convenience stores only:
            1. Flag if store name on Convenience store chain list
        iii. Pharmacies: for pharmacies only:
            1. Flag if store name on Pharmacy chain list
            2. Flag to drop records with names NOT on the chain list but in the requested chain list SIC code and not the requested pharmacy SIC code (a record without a name on the chain list and no 591205 SIC, but a different 5912 SIC)
        iv. General Merchandise: for GM only:
            1. Flag as member only if Costco, Sams, BJ's Wholesalers
    c. All other stores are left unclassified to be processed as physical activity facilities
    d. Check for cross-classification in all years
        i. Cross-classification found in (GM-liquor), (GM-pharmacy), (GM-Convenience store)
        ii. All cross-classifications involving GM converted fully to GM
4. PO Box addresses
    a. Many records had PO Box addresses
    b. Some stores with PO Box addresses could be followed through time with their ABI (InfoUSA identification number). For PO Box address stores that in another year had a legitimate and satisfactory street address-based geocode, geocodes were replaced.
        i. By store type
        ii. Merge to earliest later year available
5. Deduplication
    a. There are duplicated food stores in the data. Two different strategies were used to deduplicate – one for general merchandise stores and another for other food stores
        i. General Merchandise
            1. General merchandise stores were deduplicated by location because there were instances found where different departments were found having different listings with different names and even different addresses but that represented the same store.
                a. Because of this, impossible to deduplicate by either store name or address
            2. Determined (expert opinion and small area pilot testing) that two general merchandise stores of the same brand would not be found within 1 mile of each other.

3. All general merchandise locations imported to ArcGIS by type:
    - a. Each location buffered at 0.5 miles (Buffer)
    - b. Overlapping buffers dissolved (Dissolve)
    - c. Centroid of new buffer is new point (Features to points)

- ii. Non-General Merchandise food stores
    1. Deduplication by type (e.g. convenience stores deduplicated from convenience stores but not liquor stores)
    2. Same address and different company name
        - a. If one was a chain and duplicate addresses were non-chain, choose chain observation
    3. Same company name and slight differences in address field
        - a. Examples: "Street" vs "St"; "Unit B" in one observation and not the other
        - b. Determine first three "words" of address field (usually a number and two words)
            - i. Drop duplicates randomly
    4. Different street number
        - a. We looked at same name with slightly different address numbers
            - i. E.g. if the address number contained a typo
            - ii. Addresses within a certain raw number from each other , e.g. difference of 50
            - iii. Addresses where the ratio of the two numbers is close to one.
            - iv. Easily found real-world examples where the stores both existed
        - b. Chose not to deduplicate these as we could not guarantee that our deduplication was doing more good than bad.

- b. Restaurants were not deduplicated because of joint-operation locations (e.g. Dunkin Donuts and Baskin Robbins operating out of the same building) beyond complete address and name match
- c. Physical Activity facilities were not deduplicated beyond complete address and name match

6. Matchcodes
    - a. Only records with matchcodes of sufficient quality were retained:
    - b. Matchcodes retained:
        - i. "0" = Exact match
        - ii. "P" = Parcel Match
        - iii. "4" = Zip-4 Match (Match to XXXXX-XXXX in Zip Code fields)
    - c. Matchcodes not retained:
        - i. Zip5
        - ii. Zip2

   iii. City

Restaurant Classification

  - Restaurant classification DO-files is given in Appendix B
- Psuedocode:
  - Records are classified as the first storetype matched
    - All matches should be mutually exclusive
    - Storetype 1:
      - Identify stores by stringmatch
    - Storetype 2:
      - Identify stores by stringmatch
    - Storeype 3:
      - Identify stores by stringmatch
    - Storetype 4:
      - Identify stores by stringmatch
    - Storetype 5:
      - Identify stores by stringmatch
    - Storetype 6:
      - Identify stores by SIC code
    - Storetype 7:
      - Identify stores by SIC code
    - Storetype 8:
      - Identify stores by SIC code
    - Storetype 9:
      - Anything not elsewhere identified
    - Clean out convenience stores
      - By convenience store chain name
    - Clean out full service restaurants
      - By full service chain name list
  - Drop if matchcode or accuracy (geocoding quality variables) are unsatisfactory
    - Matchcode == T
    - Matchcode == Z
    - Matchcode == C
    - Accuracy == City Centroid
    - Accuracy == Zip+2
    - Accuracy == Zipcode

Measure Construction

Classified store and restaurant listings were imported to ArcGIS 10.2.1 to construct measures. Measures are constructed as maps with 30m grids with raster value equaling counts of resources circular Euclidean neighborhoods of 400m, 1600m, 4800m, and 800m (in 4 different grids). This was achieved in 2 steps:

1. Construction of point Shapefiles
    1. Import XY listing by latitude/longitude of classified stores by type
        1. Longitude = X
        2. Latitude = Y
        3. Geographic Coordinate System: WGS 1984
    2. Export XY-event layer to shapefile
    3. Project to USA Contiguous Albers Equal Area Conic
2. Construction of Grid-based smartmaps
    1. Extent: "-2493045.0 -1429501.25 2342655.0 1703218.75"
    2. By type and by sub-type, as defined above
    3. Use PointStatistics Tool (Spatial Analyst)
        1. Output cell size: 30 (map)
        2. Neighborhood: Circle
        3. Statistic: Sum of counts
        4. Radius: 400m, 1600m, 4800m, 8000m

Software

Database management was achieved using Stata 13.1

GIS work was achieved using ArcGIS 10.2.1

Appendix

A: Scripts

Note: Script will need to be adjusted with project-specific file names and locations.

Measure Construction – Info USA

```
*InfoUSA Measure Construction

/*
*************
Overall file structure:
Within the main folder, there are folders for each of the store types (SupermarketGrocery, Liquor, Pharmacy, GeneralMerchandise, and Convenience) as well as a folder called Working. Within each stor type folder is another called Datasets.
*/



clear all
set more off
cap log close

*set to main folder
cd {insert file location}

*******
*DO File to reduce InfoUSA data files to only necessary variables for making measures (Original Name:

/*
 Grocery_PAFac_'year'_nodup.dta data files were made as follows:
Variables unnecessary for classification were dropped at this point to allow for quicker processing, as the volume of observations made the loading process very slow. Additionally, the field names and order were not standard, so some standardization was carried out at this point.
*/


*InfoUSA-delivered variable names change over years. This first section must be adjusted to account for this variation.
forvalues i = {years of interest}{
	use {input store data with geocodes}, clear
	drop deliverypointbarcode-salesvolumecode franchisespecialtycodes-contactgender callstatus-sortfield subsidiarynumber-longitude6decimal endofrecordmarker-keycode1 census_county_2000-longitude_2010 count year selectednaicstitle selectednaicscode
	save Working\InfoUSAStoreMeasuresDataset_`i'.dta, replace
	duplicates report
	duplicates report companyname address cityname statealpha digitzipcode
	clear
	}


*This portion of the file separates different store types for measure construction

forvalues i= {years of interest}{
	use Working\InfoUSAStoreMeasuresDataset_`i'.dta, clear


*Count field value 1 required to make density maps
	gen count=1

*Liquor Stores
	egen liquor = anymatch(primarysic), values(592102 592103 592104)

*Gen Merch
	gen genmerch=0
	foreach j in MEIJER* WALMART* WAL?MART* SUPER?WAL*MART KMART* SUPERKMART
SUPER?KMART SUPERK-MART SUPER?K-MART K-MART* TARGET SUPERTARGET SUPER?TARGET
TARGET?SUPER* SAM*S?CLUB COSCO* COSTCO* BJ'S?WHOLESALE* BJS?WHOLESALE*{
		quietly replace genmerch=1 if strmatch(companyname,"`j'")
		}
	*NB: the lists of stringmatch names were all iteratively examined to be both properly inclusive
and exclusive
	quietly replace genmerch=0 if companyname == "WAL-MART NEIGHBORHOOD MARKET"
	quietly replace genmerch=0 if companyname == "WALMART NEIGHBORHOOD MARKET"
	quietly replace genmerch=0 if companyname == "COSCOB SPORTS MEDICINE"




*Pharmacies
	*Pharmacies were pulled out of 4-digit SIC code 5912
	quietly tostring primarysic, gen(stringsic)
	gen sic4=substr(stringsic,1,4)
	gen pharm = 1 if sic4=="5912"

	*Lots of Pharmacies are part of grocery stores OR general merchandise stores. For example
pharm flags 764 walmart pharmacies in 2013
	tab companyname if pharm==1 & genmerch==1
	foreach j in COSCO?PHARMACY COSTCO?PHARMACY K-MART?PHARMACY
KMART?STORE?PHARMACY KMART?STORES?TOLEDO?PHARMACY MEIJER?PHARMACY
WALMART?PHARMACY{
		quietly replace pharm=0 if strmatch(companyname,"`j'") & pharm==1
		}
	*For grocery stores this separation step is done down below.
	*Pharmacies that are departments of general merchandise stores should not count as 
pharmacies:
	quietly duplicates tag address cityname statecode zip4 if genmerch==1, 
gen(duppharmgenmerch)
	quietly duplicates tag latitude longitude if genmerch==1, gen(duplatlonggenmerch)


*Convenience Stores
	egen conv = anymatch(primarysic), values(541103 554101 554103)
	*Are any of these convenience stores actually pharmacies?
	foreach j in WALGREENS* CVS* RITE?AID HEALTH?MART KATZ?GROUP SHOPPERS?DRUG?MART 
MEDICINE?SHOPPE?INT* MEDICINE?SHOP?INT* USA?DRUG* KERR KINNEY?DRUG* THRIFTY?WHITE* 
AURORA?PHARMAC* DISCOUNT?DRUG?MART MARC?GLASSMAN* BARTELL?DRUG* CARE?PHARMAC* 
HOMETOWN?PHARMAC* LEWIS?DRUG* BIOSCRIPT BIO?SCRIP* NEIGHBORCARE?PHARMAC* 
NEIGHBOR?CARE* FAIRVIEW?PHARMAC* FAIRIVEW?PHARMACY?SVC 
NAVARRO?DISCOUNT?PHARMAC* HI?SCHOOL?PHARMAC* MEDICINE?CHEST?PHARMAC* 
FRUTH?PHARMAC* HENRY?FORD?HEALTH?SYSTEM?PHARMARC* FORD?HEALTH?SYSTEM?PHARMAC* 
FORD?PHARMAC* HENRY?FORD?PHARMAC* HY?VEE* PHARMACA?INTEGRATIVE?PHARMAC* 
FAGEN?PHARMAC* LIFECHEK?DRUG* ST*JOHN*S?HEALTH* USAVE U?SAVE FAMILY?PHARMAC* 
RECEPT?PHARMAC* MAXOR?NATIONAL?PHARMACY?S* OSBORNE?DRUG* KNIGHT?DRUG* SAV-MOR* 
SAVMOR*{
		quietly replace conv = 0 if strmatch(companyname,"`j'") & conv==1
		}

	*Replace pharmacy ==1  for every pharmacy found under the convenient stores:
	quietly replace pharm=1 if conv==0 & (primarysic==541103 | primarysic==554101 | 
primarysic==554103)

	*Are any of the convenience stores chain supermarkets or general merchandise?
	foreach j in *55?FULTON* *FIFTY*FIVE?FULTON* 99?CENTS?ONLY* A&P A?&?P* ACME* 
AHOLD?USA* A*J'S?FINE?FOODS* *ALBERTSONS* *ALBERTSON'S* ALDI ALDI?INC ALDI?F* ALDI?S* 
ALEX?LEE* ALPS* AMIGO* ASSOCIATED?FOOD?STORE* ASSOCIATED?FOODS ASSOCIATED?INC 
ASSOCIATED?MARKET ASSOCIATED?STORE ASSOCIATED?SUPER* ASSOCIATED?FRESH?MARKET 
ASSOCIATION?SUPERMARKET ASSOICATED?SUPERMARKET ASSOCIATED?WHOLESALERS BAG?N?SAVE 
BAKER'S BASHAS* BEL?AIR* BIG?Y* SUPER*BIG?Y* BILO* BI-LO* BI?LO* BOTTOM?DOLLAR* 
BROOKSHIRE* BUEHLER* BULKLEY?VALLEY?WHOLESALE* BUTERA* C*&*S?WHOLESALE* CARRS* 
CARR'S* CASH*&*CARRY* CASH?WISE?FOOD* CENTRAL?GROCER* CITY?MARKET* COBORN*S* 
COOPER*S?FOOD* COPPO*S* COSTCO* COST?CO* CROPPER*S* CUB?FOOD* 
D*&*W?FRESH?MARKET* DECA?COMMISSARY DEFENSE?COMMISSARY?AGENCY DELHAIZE?AMERICA* 
DEMOULA*S?MARKET?BASKET DILLIONS DOLLAR?GENERAL DOMINICK*S* EARTH?ORIGIN* 
ECONOFOOD* ECONO?FOOD* EL?SUPER*MARKETS FAMILY?DOLLAR FAMILY?FARE* FAMILY?FOOD* 
FAMILY?FRESH* FAMILY?THRIFT?CENTER FAREWAY* FARM?FRESH* FIESTA?MART* FOOD?4?LESS* 
FOOD?BASICS* FOOD?DYNASTY* FOOD?FAIR* FOOD?LION FOODTOWN* FOOD?TOWN* 
FOOD*WORLD* FOOD?CITY FOODCITY FOOD*MAX* FOODS?CO FOREST?HILLS?FOODS FRED?MEYER* 
FRESH?&*EASY* FRESH?ACRES FRY*S?MARKETPLACE FRY*S?FOOD* GERBES* 
GERMANTOWN?FRESH?MARKET GIANT GIANT?EAGLE SUPER*GIANT* GLEN'S?MARKET GOLUB 
GORDON?FOOD?SERVICE* GFS GREENWISE GROCER*S?SUPPLY GROCERY?OUTLET 
GU?FAMILY?MARKET HANNAFORD* HARRIS?TEETER* HARVEST?FOOD* HARVEY'S
```

```
HARVEY'S?SUPERMARKET HARVEYS HARVEYS?SUPERMARKET* HEB?FOOD* H?E?B* HOLIDAY?FARMS*
HORNBACHER*S* HOUCHEN*?MARKET* HY-VEE* *INGLES?MARKET* JAY?C JEWEL?F JEWEL JEWEL?M
KEY?FOOD* KEY?FRESH?&?NATURAL KING?KULLEN* KINGS KING'S KING'S?FOOD KING'S?FOODS
KINGS?FOOD* KINGS?SUPERMARKET* KINGS?MARKET KINGS?SUPER?MARKET* K?MART* K-MART
SUPER*K*MART* KROGER* K?VA?T LOBILL?FOOD* LOCUST?VALLEY?MARKET LOGLI* LOWE'S
LOWE'S?MARKETPLACE LOWE'S?MARKET?PLACE LOWES LOWES?MARKETPLACE
LOWES?MARKET?PLACE LOWES?SUPERMARKET* LOWE'S?SUPERMARKET* LUCKY?SUPERMARKET*
MARIANO'S?FRESH?MARKET MARKET?BASKET MARKET?STREET MARSH?HOMETOWN?MARKET*
MARSH?SUPERMARKET* MARTINS MARTIN'S?FOOD* MARTIN?FOOD* MARTIN'S?SUPERMARKET*
MARTIN'S?SUPER?MARKET* MEIJER* METRO?MARKET MILFORD?FARMS MR.?Z'S* NEIL'S
NEIL'S?FOOD* NO?FRILLS?SUPERMARKET* NOB?HILL?FOOD* O'MALIA'S* OVERWAITEA* OWEN'S P&C*
P?&?C* PATHMARK PAY*LESS?SUPER*MARKET* PEAPOD* PICK*N?SAVE PIGGLY?WIGGLY* PIX
PIX?FOOD* PRAIRIE?MARKET PRICE?CHOPPER* PRICE?RITE* PRICERITE* PRICE-RITE*
PRICESMART?FOODS PUBLIX* QFC* Q?F?C* QUALITY?MARKET* RAINBOW RAINBOW?FOOD*
RAINBOW?GROCER* RAINBOW?MARKET* RAINBOW?SUPERMARKET* RALEY*S* RALPHS
RALPHS?SUPERMARKET* RALPH'S?FOOD* RALPH'S?MARKET* RALPH'S?SUPERMARKET*
RALPHS?SUPER?MARKET* RALPH'S?SUPER?MARKET* RALPHS?GROCER* RALPH'S?GROCER*
RANDALL'S?FOOD* ROSAUER*S* ROUNDY*S* RULER?FOODS SABOR* SAFEWAY* *SHOPRITE*
SAM*S?CLUB* SAVEMART SAVE?MART* SAVE?ON?FOOD* SAVE?SMART SAV*A*LOT SAVE-A-
LOT?FOOD* SCHNUCK* SCOTT'S?FOOD* SEARS?HOLDING SHAW?SUPERMARKET*
SHAW'S?SUPERMARKET* SHAW'S-OSCO SHOP*SAVE SHOPPER*S* SHOP*RITE* SHUR*FINE*
SMART?&?FINAL S-MART?FOODS SMITH'S?FOOD?&?DRUG* SPARTANNASH*
SPROUTS?FARMER*S?MARKET* STAR?MARKET* STATER?BROS* STATER?BROS*MARKETS
STOP?N?SHOP* STOP?&?SHOP* SUPER*STOP?N?SHOP SUPER*STOP?N?SHOP STRACK*&*VAN*TIL*
SUN?MART?FOOD* SUNFLOWER?M* SUPER*1*FOOD* SUPER?DOLLAR?DISCOUNT?FOOD* SUPER?K-
MART SUPER?K?MART SUPERKMART SUPER*FRESH* SUPERIOR?GROCER*
SUPERMERCADO?NUESTRA?FAMILIA SUPERVALU SWEETBAY* TABLE?&?VINE *TARGET TARGET?F*
FOOD?EMPORIUM FRESH?MARKET TOM?THUMB* TOP?FOOD* TOPS?FRIENDLY?MARKET*
TRADER?JOE*S* ULTRA?FOODS* UNITED?NATURAL?FOOD* URBAN?FARES URBAN?MARKET
URM?STORES VALU*LAND VG*S?GROCERY VILLAGE*MARKET VITELIO*S* VONS* VON'S*
WAKEFERN?FOOD WALDBAUM*S* *WALMART* *WAL-MART* WEGMAN*S* WEIS?MARKET*
WHOLE?FOODS* WHOLESALE?FOOD?OUTLET WILD?OATS* WINCO* WINN-DIXIE WOODMAN*S*
YOKE'S?FRESH?MARKET{
        quietly replace conv = 0 if strmatch(companyname,"`j'") & conv==1
        }
        quietly replace conv=1 if companyname == "BIALOWIESKI MARKET"
        quietly replace conv=1 if companyname == "BILO GAS N GO"



*Supermarkets and Grocery Stores
    egen supgroc = anymatch(primarysic), values(541101 541102 541104 541105 541106 541107
541108 541109)

    *Some other stores are included in Supermarkets and Grocery stores that should not be:
    quietly replace supgroc=0 if genmerch==1
    foreach j in WALGREENS* CVS* RITE?AID HEALTH?MART KATZ?GROUP SHOPPERS?DRUG?MART
MEDICINE?SHOPPE?INT* MEDICINE?SHOP?INT* USA?DRUG* KERR KINNEY?DRUG* THRIFTY?WHITE*
```

```
AURORA?PHARMAC* SAV?MOR?PHARMAC* SAVMOR?PHARMAC* DISCOUNT?DRUG?MART
MARC?GLASSMAN* BARTELL?DRUG* CARE?PHARMAC* HOMETOWN?PHARMAC* LEWIS?DRUG*
BIOSCRIPT BIO?SCRIP* NEIGHBORCARE?PHARMAC* NEIGHBOR?CARE* FAIRVIEW?PHARMAC*
FAIRIVEW?PHARMACY?SVC NAVARRO?DISCOUNT?PHARMAC* HI?SCHOOL?PHARMAC*
MEDICINE?CHEST?PHARMAC* FRUTH?PHARMAC* HENRY?FORD?HEALTH?SYSTEM?PHARMARC*
FORD?HEALTH?SYSTEM?PHARMAC* FORD?PHARMAC* HENRY?FORD?PHARMAC*
PHARMACA?INTEGRATIVE?PHARMAC* FAGEN?PHARMAC* LIFECHEK?DRUG* ST*JOHN*S?HEALTH*
USAVE U?SAVE FAMILY?PHARMAC* RECEPT?PHARMAC* MAXOR?NATIONAL?PHARMACY?S*
OSBORNE?DRUG* KNIGHT?DRUG*{
                quietly replace pharm = 1 if strmatch(companyname,"`j'") & supgroc==1
                }
                quietly replace supgroc = 0 if pharm==1

        foreach j in 76 ALLSUP'S?CONVENIENCE?STORE AMOCO AMPM A?PLUS ARCO BIG?APPLE
BIG?JOHN'S BIG?JOHN'S?FOODETTE BIG?JOHN'S?HOUSEHOLD?FOODS BP BP?CONNECT
BUCKY'S?EXPRESS CASEY'S?GENERAL?STORE CEFCO?FOOD?STORE CENEX CHEVRON
CHEVRON?APSI?STORE CHEVRON?AT?CHIMACUM CHEVRON?BREWER CHEVRON?COUNTRY?MARKET
CHEVRON?DOLORES CHEVRON?EXPRESS?21 CHEVRON?FOOD?MART CHEVRON?FOOD?MART?ARA
CHEVRON?FOOD?MART?INC CHEVRON?FOOD?MART-SNAPFIN CHEVRON?FOOD?STORE
CHEVRON?MAIN?STOP CHEVRON?MINIMARKET-LINDSAY CHEVRON?STATION CHEVRON?STATIONS?INC
CHEVRON?TOWN?&?CNTRY?FOOD?STORE CHEVRON?WAYNE'S?MARKET CHUCKY'S?FOOD?STORE
CIRCLE?K CITGO CLARK COASTAL CONOCO CONVENIENT?FOOD?MART CORNER?STORE COUNTRY?FAIR
CUMBERLAND?FARMS DAIRY?MART DANDY?MINI?MART DISCOUNT?FOOD?MART
E?TO?Z?FOOD?MART E?Z?EXPRESS?FOOD?MART E?Z?FOOD?MART E?Z?FOODMART E?Z?MART
EAZY?MART EZ?FOOD?MART E-Z?FOOD?MART EZ?FOODMART E-Z?FOODMART EZ?LANE?FOOD?MART
EZ?MART E-Z?MART EZ?SHOP?FOODMART E-Z?STOP?FOOD?MART E-ZEE?MART E-Z-N?FOOD?MART
EXPRESS?LANE EXXON EXXON?COIN?JOCK EXXON?DISTRIBUTOR EXXON?EXPRESS EXXON?FOOD?MART
EXXON?HANDIPLUS EXXON?KULSUM EXXON?MART EXXON?SHOPS?&?DEALERS FAMILY?FARE
FARM?STORE FAS?MART FAST?STOP FIVE?STAR?FOOD?MART FLASH?FOODS FLASH?MARKET
GAS?EXPRESS HANDI?KORNER HANDI?MARKER HANDI?MARKET?INC HANDI?MART HANDI?SPOT
HANDI?STOP HANDI?STOP?ONE HANDI?STOP?SUPERETTE HANDI?STOP-DINO'S?PIZZA HANDIMART
HANDI-STOP HESS HESS?09381 HESS?CORP HESS?CORP?09424 HESS?CORP?AMERADA
HESS?CORP?STATION?09448 HESS?DO?IT?CTR HESS?EXPRESS HESS?EXPRESS?09389?INC
HESS?EXPRESS?CORP HESS?MART HESS?STATION?09307 HESS?STATION?32391 HESS?STORE HUCKS
HY?VEE?GAS JOE'S?KWIK?MART KANGAROO?EXPRESS KRAUSZER'S?FOOD?STORE
KRAUSZERS?FOOD?STORE KUM?&?GO KWIK?FILL KWIK?PIK KWIK?STOP KWIK?TRIP LITTLE?GENERAL
LUKOIL MAC'S MAC'S?CONVENIENCE?STORES MAPCO?EXPRESS MAVERIK MOBIL MOBIL?MART
NICE?N?EASY?GROCERY?SHOPPE OPTIMA PILLIPS?66 PILOT?FOOD?MART PLAID?PANTRY RACEWAY
RED?APPLE ROAD?RANGER ROADRUNNER?MARKET ROYAL?FARMS SAM'S?FOOD?STORE
SAMS?FOOD?STORE SCOT?MARKET SCOTCHMAN SHEETZ SHELL SINCLAIR SPEEDWAY STAR?STOP
STWEART'S?SHOP STOP?IN STRIPES SUNMART SUNOCO SUPER?AMERICA TAYLOR?PETROLEUM
TEDESCHI?FOOD?SHOP TEXACO TIMEWISE?FOOD?STORE TOM?THUMB TRUE?NORTH UNIMART
UNI?MART UNITED?DAIRY?FARMERS VILLAGE?PANTRY WAWA WILCO?FOOD?MART XTRA?MART
YOUNG'S ZIP?TRIP{
                quietly replace conv = 1 if strmatch(companyname,"`j'") & supgroc==1
                }
                quietly replace supgroc=0 if conv==1
```

```
    gen whole_sale=0
    foreach j in BOZZUTO*S*{
        quietly replace whole_sale=1 if strmatch(companyname,"`j'")
        }

    ******************************************
    **CHAINS: AFTER CLASSIFICATION, pull chains
    ******************************************
*Supermarkets/Grocery Stores
    *Flag stores with > $2 million in annual sales. This will only be useful for Supermarkets/Grocery Stores
    gen twomil = 0
        quietly replace twomil = 1 if actualsalesvolume>1999 & actualsalesvolume!=.
    * Create Regional 11+ designator - shows stores that have 11 or more stores in the region = FMI definition of Supermarket
    gen region=0
        quietly replace region=1 if statealpha=="CT" | statealpha=="ME" | statealpha=="MA" | statealpha=="NH" | statealpha=="RI" | statealpha=="VT"
        quietly replace region=2 if statealpha=="NJ" | statealpha=="NY" | statealpha=="PA"
        quietly replace region=3 if statealpha=="IN" | statealpha=="IL" | statealpha=="MI" | statealpha=="OH" | statealpha=="WI"
        quietly replace region=4 if statealpha=="IA" | statealpha=="KS" | statealpha=="MN" | statealpha=="MO" | statealpha=="NE" | statealpha=="ND" | statealpha=="SD"
        quietly replace region=5 if statealpha=="DE" | statealpha=="DC" | statealpha=="FL" | statealpha=="GA" | statealpha=="MD" | statealpha=="NC" | statealpha=="SC" | statealpha=="VA" | statealpha=="WV"
        quietly replace region=6 if statealpha=="AL" | statealpha=="KY" | statealpha=="MS" | statealpha=="TN"
        quietly replace region=7 if statealpha=="AR" | statealpha=="LA" | statealpha=="OK" | statealpha=="TX"
        quietly replace region=8 if statealpha=="AZ" | statealpha=="CO" | statealpha=="ID" | statealpha=="NM" | statealpha=="MT" | statealpha=="UT" | statealpha=="NV" | statealpha=="WY"
        quietly replace region=9 if statealpha=="AK" | statealpha=="CA" | statealpha=="HI" | statealpha=="OR" | statealpha=="WA"
    forvalues n=1/9{
        quietly duplicates tag companyname if region==`n', gen(regioncount`n')
        }
    gen regioncount=0
        forvalues n=1/9{
        quietly replace regioncount=regioncount`n' if regioncount`n'!=.
        drop regioncount`n'
        }
    gen elevenregion=0
        quietly replace elevenregion=1 if regioncount>9 & regioncount!=.
        quietly replace elevenregion=0 if companyname==""


    *Flag stores that are chain stores
```

```
        gen sgnamechain = 0
        foreach j in *55?FULTON* *FIFTY*FIVE?FULTON* 99?CENTS?ONLY* A&P A?&?P* ACME*
AHOLD?USA* A*J'S?FINE?FOODS* *ALBERTSONS* *ALBERTSON'S* ALDI ALDI?INC ALDI?F* ALDI?S*
ALEX?LEE* ALPS* AMIGO* ASSOCIATED?FOOD?STORE* ASSOCIATED?FOODS ASSOCIATED?INC
ASSOCIATED?MARKET ASSOCIATED?STORE ASSOCIATED?SUPER* ASSOCIATED?FRESH?MARKET
ASSOCIATION?SUPERMARKET ASSOICATED?SUPERMARKET ASSOCIATED?WHOLESALERS BAG?N?SAVE
BAKER'S BASHA*S* BEL?AIR* BIG?Y* SUPER*BIG?Y* BILO* BI-LO* BI?LO* BOTTOM?DOLLAR*
BROOKSHIRE* BUEHLER* BULKLEY?VALLEY?WHOLESALE* BUTERA* C*&*S?WHOLESALE* CARNIVAL*
CARRS* CARR'S* CASH*&*CARRY* CASH?WISE?FOOD* CENTRAL?GROCER* CITY?MARKET* COBORN*S*
COOPER*S?FOOD* COPPO*S* COSTCO* COST?CO* CROPPER*S* CUB?FOOD*
D*&*W?FRESH?MARKET* DECA?COMMISSARY DEFENSE?COMMISSARY?AGENCY DELHAIZE?AMERICA*
DEMOULA*S?MARKET?BASKET DILLON* DOLLAR?GENERAL DOMINICK*S* EARTH?ORIGIN*
ECONOFOOD* ECONO?FOOD* EL?SUPER*MARKETS FAMILY?DOLLAR FAMILY?FARE* FAMILY?FOOD*
FAMILY?FRESH* FAMILY?THRIFT?CENTER FAREWAY* FARM?FRESH* FIESTA?MART* FOOD?4?LESS*
FOOD?BASICS* FOOD?DYNASTY* FOOD?FAIR* FOOD?LION FOOD*TOWN* FOOD*WORLD* FOOD?CITY
FOODCITY FOOD*MAX* FOODS?CO FOREST?HILLS?FOODS FRED?MEYER* FRESH*&*EASY*
FRESH?ACRES FRY*S?MARKETPLACE FRY*S?FOOD* GERBES* GERMANTOWN?FRESH?MARKET GIANT
GIANT?EAGLE SUPER*GIANT* GLEN*S?MARKET GOLUB GORDON?FOOD?SERVICE* GFS GREENWISE
GROCER*S?SUPPLY GROCERY?OUTLET GU?FAMILY?MARKET HANNAFORD* HARRIS?TEETER*
HARVEST?FOOD* HARVEY'S HARVEY'S?SUPERMARKET HARVEYS HARVEYS?SUPERMARKET* HEB?FOOD*
H?E?B* HOLIDAY?FARMS* HORNBACHER*S* HOUCHEN*?MARKET* HY-VEE* IGA *-IGA
*INGLES?MARKET* JAY?C JEWEL* KEY?FOOD* KEY?FRESH?&?NATURAL KING?KULLEN* KINGS KING'S
KING'S?FOOD KING'S?FOODS KINGS?FOOD* KINGS?SUPERMARKET* KINGS?MARKET
KINGS?SUPER?MARKET* K*MART* SUPER*K*MART* KROGER* K?VA?T LOBILL?FOOD*
LOCUST?VALLEY?MARKET LOGLI* LOWE'S LOWE'S?MARKETPLACE LOWE'S?MARKET?PLACE LOWES
LOWES?MARKETPLACE LOWES?MARKET?PLACE LOWES?SUPERMARKET* LOWE'S?SUPERMARKET*
LUCKY?SUPERMARKET* MARIANO*S?FRESH?MARKET MARKET?BASKET MARKET?STREET
MARSH?HOMETOWN?MARKET* MARSH?SUPERMARKET* MARTINS MARTIN'S?FOOD* MARTIN?FOOD*
MARTIN'S?SUPERMARKET* MARTIN'S?SUPER?MARKET* MEIJER* METRO?MARKET MILFORD?FARMS
MR.?Z'S* NEIL'S NEIL'S?FOOD* NO?FRILLS?SUPERMARKET* NOB?HILL?FOOD* O'MALIA'S*
OVERWAITEA* OWEN'S P&C* P?&?C* PATHMARK PAY*LESS?SUPER*MARKET* PEAPOD* PICK*N?SAVE
PIGGLY?WIGGLY* PIX PIX?FOOD* PRAIRIE?MARKET PRICE?CHOPPER* PRICE?RITE* PRICERITE* PRICE-
RITE* PRICESMART?FOODS PUBLIX* QFC* Q?F?C* QUALITY?MARKET* RAINBOW RAINBOW?FOOD*
RAINBOW?GROCER* RAINBOW?MARKET* RAINBOW?SUPERMARKET* RALEY*S* RALPHS
RALPHS?SUPERMARKET* RALPH'S?FOOD* RALPH'S?MARKET* RALPH'S?SUPERMARKET*
RALPHS?SUPER?MARKET* RALPH'S?SUPER?MARKET* RALPHS?GROCER* RALPH'S?GROCER*
RANDALL'S?FOOD* ROSAUER*S* ROUNDY*S* RULER?FOODS SABOR* SAFEWAY* *SHOPRITE*
SAM*S?CLUB* SAVE*MART* SAVE?ON?FOOD* SAVE?SMART SAV*A*LOT SAVE-A-LOT?FOOD* SAV-
MOR* SCHNUCK* SCOTT'S?FOOD* SEARS?HOLDING SHAW?SUPERMARKET* SHAW'S?SUPERMARKET*
SHAW'S-OSCO SHOP*SAVE SHOPPER*S* SHOP*RITE* SHUR*FINE* SMART?&?FINAL S-MART?FOODS
SMITH'S?FOOD?&?DRUG* SPARTANNASH* SPROUTS?FARMER*S?MARKET* STAR?MARKET*
STATER?BROS* STATER?BROS*MARKETS STOP?N?SHOP* STOP?&?SHOP* SUPER*STOP?N?SHOP
SUPER*STOP?N?SHOP STRACK*&*VAN*TIL* SUN*MART?FOOD* SUNFLOWER* SUPER*1*FOOD*
SUPER?DOLLAR?DISCOUNT?FOOD* SUPER*K*MART SUPER*FRESH* SUPERIOR?GROCER*
SUPERMERCADO?NUESTRA?FAMILIA SUPERVALU SWEETBAY* TABLE?&?VINE *TARGET*
FOOD?EMPORIUM FRESH?MARKET TOM?THUMB* TOP?FOOD* TOPS?FRIENDLY?MARKET*
TRADER?JOE*S* ULTRA?FOODS* UNITED?NATURAL?FOOD* URBAN?FARES URBAN?MARKET
URM?STORES VALU*LAND VG*S?GROCERY VILLAGE*MARKET VITELIO*S* VONS* VON'S*
```

```
WAKEFERN?FOOD WALDBAUM*S* *WALMART* *WAL-MART* WEGMAN*S* WEIS?MARKET*
WHOLE?FOODS* WHOLESALE?FOOD?OUTLET WILD?OATS* WINCO* WINN-DIXIE WOODMAN*S*
YOKE'S?FRESH?MARKET{
                quietly replace sgnamechain = 1 if strmatch(companyname,"`j'") & supgroc==1
                }
        quietly replace sgnamechain = 1 if strmatch(companyname,"* IGA *") & supgroc==1

*Convenience Stores
        gen convnamechain=0
        foreach j in 76 2?GO?MART ADMIRAL?PETROLEUM ALLSUP* AMOCO AMPM A?PLUS ARCO
BIG?APPLE BIG?JOHN'S BIG?JOHNS BP* BREAK?TIME BUCKYS?EXPRESS BUCKY'S?EXPRESS
CASEYS?GENERAL?STORE CASEY'S?GENERAL?STORE CEFCO?FOOD?STORE CENEX CERTIFIED CHEVRON*
CHOICE CHUCK*S?FOOD?STORE CHUCK'S?FOOD?STORE CIRCLE?K CITGO CLARK
COAST?GUARD?MINI?MART COASTAL CONOCO CONVENIENT?FOOD?MART CORNER?STORE
COUNTRY?FAIR CUMBERLAND?FARMS DAILY*S?EXPRESS DAILY'S?EXPRESS DAILYS?TRI?STAR?ENERGY
DAILY'S?TRI?STAR?ENERGY DAIRY?MART DANDY?MINI?MART DISCOUNT?FOOD?MART
DUCHESS?SHOPPE E*Z*MART EDDYS?MART EDDY'S?MART EXPRESS?LANE EXXON* FAMILY?FARE
FARM?STORE FAS?MART FAST?FOOD*FUEL FAST?STOP FIVE?STAR?FOOD?MART FLASH?FOODS
FLASH?MARKET FLYING?J FOOD?FAST?STORE FREEDOM?VALU?CENTER GAS?EXPRESS GAS?MART?USA
GATE?FOOD?POST GETGO GETTY GIANT GO?MART?FOOD?STORE GOASIS GULF HANDEE?HUGO'S
HANDEE?HUGOS HANDI* HESS* HOLIDAY?STATIONSTORES HOWDYS?FOODMART
HOWDY'S?FOODMART HUCK'S HUCKS HY?VEE?GAS* JACKSONS?FOOD?STORE JOES?KWIK?MART
JOE'S?KWIK?MART JUMPIN*?JIMMY*S?GAS?MART KANGAROO?EXPRESS KRAUSZER*S?FOOD?STORE
KUM?&?GO KWIK?FARMS KWIK?FILL KWIK?PIK KWIK?SHOP KWIK?STAR KWIK?STOP KWIK?TRIP
LI*L?CRICKET?STORES LITTLE?GENERAL LOVE'S?COUNTRY?STORE LOVES?COUNTRY?STORE
LOVE'S?TRAVEL?STOP LOVES?TRAVEL?STOP LUKOIL MAC'S* MACS* MAPCO?EXPRESS MAPCO?MART
MARINE?CORPS?SHOPPETTE MAVERIK MEIJER?GAS?STATION MFA?OIL MOBIL MOBIL?MART
MOTO?MART MURPHY?EXPRESS MURPHY?USA MUSTANG NEXCOM?MINI?MART NEXT?DOOR?STORES
NICE?N?EASY?GROCERY?SHOPPE OPTIMA PETRO?CARD?24 PETRO?EXPRESS PETRO?STOPPING?CENTER
PHILLIPS?66 PILOT?FOOD?MART PILOT?TRAVEL?CENTER PLAID?PANTRY PRESTO QUICKCHEK QUIKTRIP
RACETRAC RACEWAY RED?APPLE ROAD?RANGER ROADRUNNER?MARKET ROYAL?FARMS
SAM'S?FOOD?STORE SAMS?FOOD?STORE SCOT?MARKET SCOTCHMAN SEVEN?ELEVEN SHEETZ SHELL
SHOR?STOP SINCLAIR SPEEDWAY SPINX?STORE SPRINT?MART STAR?STOP STEWARTS?SHOP
STEWART'S?SHOP STOP?IN STRIPES SUNDIAL?DELI?MART SUNMART SUNOCO SUPER?AMERICA*
TAYLOR?PETROLEUM TEDESCHI?FOOD?SHOP TERRIBLE?HERBST TESORO TEXACO THORNTON'S*
THORNTONS* TIMEWISE?FOOD?STORE TOM?THUMB TOWN?PUMP?FOOD?STORE TROOP?STORE
TRUE?NORTH TURKEY?HILL?MINIT?MART UNI?MART UNIMART UNITED?DAIRY?FARMERS
USA?FUEL?CENTER VILLAGE?PANTRY WAWA WILCO?FOOD?MART WILCO?TRAVEL?PLAZA XTRA?MART
YOUNG'S YOUNGS ZIP?TRIP{
                quietly replace convnamechain = 1 if strmatch(companyname,"`j'") & conv==1
                }

*Pharmacies
        gen phnamechain = 0
        foreach j in WALGREENS* CVS* RITE?AID HEALTH?MART KATZ?GROUP SHOPPERS?DRUG?MART
MEDICINE?SHOPPE?INT* MEDICINE?SHOP?INT* USA?DRUG* KERR KINNEY?DRUG* THRIFTY?WHITE*
AURORA?PHARMAC* DISCOUNT?DRUG?MART MARC?GLASSMAN* BARTELL?DRUG* CARE?PHARMAC*
HOMETOWN?PHARMAC* LEWIS?DRUG* BIOSCRIPT BIO?SCRIP* NEIGHBORCARE?PHARMAC*
```

```
NEIGHBOR?CARE* FAIRVIEW?PHARMAC* FAIRIVEW?PHARMACY?SVC
NAVARRO?DISCOUNT?PHARMAC* HI?SCHOOL?PHARMAC* MEDICINE?CHEST?PHARMAC*
FRUTH?PHARMAC* HENRY?FORD?HEALTH?SYSTEM?PHARMARC* FORD?HEALTH?SYSTEM?PHARMAC*
FORD?PHARMAC* HENRY?FORD?PHARMAC* HY?VEE* PHARMACA?INTEGRATIVE?PHARMAC*
FAGEN?PHARMAC* LIFECHEK?DRUG* ST*JOHN*S?HEALTH* USAVE U?SAVE FAMILY?PHARMAC*
RECEPT?PHARMAC* MAXOR?NATIONAL?PHARMACY?S* SA OSBORNE?DRUG* KNIGHT?DRUG* SAV-
MOR* SAVMOR*{
                quietly replace phnamechain = 1 if strmatch(companyname,"`j'") & pharm==1
                }
        *This variable culls those that should not have been pulled due to the name list. What that
means is that we asked for SIC code 591205, and then for a name search to be performed for the above
list of names on SIC4 5912. So if there is a record that has a SIC4 of 5912 but does not have a name on
this list, it was not pulled as a pharmacy.
        gen untruepull=1 if phnamechain==0 & primarysic!=591205
        quietly replace pharm=0 if untruepull==1
        drop sic4 untruepull


*General Merchandise
        quietly gen genmerchmem=1  if genmerch==1 & (strmatch(companyname,"COSCO*") |
strmatch(companyname,"COSTCO*") | strmatch(companyname,"SAM*S?CLUB")|
strmatch(companyname,"BJ'S?WHOLESALE*") | strmatch(companyname,"BJS?WHOLESALE*"))

*Liquor Stores are not designated to be chain or not.

*Supermarkets are "typed" at this point, so that if in classifying other store types something was turned
into a supermarket, it still gets a type.
        gen type_reg=0
                quietly replace type_reg=1 if twomil==0 & sgnamechain==0 & elevenregion==0
                quietly replace type_reg=2 if twomil==0 & sgnamechain==0 & elevenregion==1
                quietly replace type_reg=3 if twomil==0 & sgnamechain==1 & elevenregion==0
                quietly replace type_reg=4 if twomil==0 & sgnamechain==1 & elevenregion==1
                quietly replace type_reg=5 if twomil==1 & sgnamechain==0 & elevenregion==0
                quietly replace type_reg=6 if twomil==1 & sgnamechain==0 & elevenregion==1
                quietly replace type_reg=7 if twomil==1 & sgnamechain==1 & elevenregion==0
                quietly replace type_reg=8 if twomil==1 & sgnamechain==1 & elevenregion==1
label define classification 1 "<$2 Mill, No Chain Name, <11 Stores" 2 "<$2 Mill, No Chain Name, 11+
Stores" 3 "<$2 Mill, Chain Name, <11 Stores" 4 "<$2 Mill, Chain Name, 11+ Stores" 5 "$2+ Mill, No Chain
Name, <11 Stores" 6 "$2+ Mill, No Chain Name, 11+ Stores" 7 "$2+ Mill, Chain Name, <11 Stores" 8 "$2+
Mill, Chain Name, 11+ Stores"
label values type_reg classification

*It was found that many records had PO Boxes in the address fields.
*Flag PO Boxes by Store Type:
        gen pobox=0
        quietly replace pobox=1 if strmatch(address, "PO BOX*")
        quietly replace pobox=1 if strmatch(address, "*PO BOX*")
        quietly replace pobox=1 if strmatch(address, "POST OFFICE BOX*")
        quietly replace pobox=1 if strmatch(address, "RR BOX*")
        quietly replace pobox=1 if strmatch(address, "*RR BOX*")
        quietly replace pobox=1 if strmatch(address, "RURAL ROUTE BOX*")
        quietly replace pobox=1 if strmatch(address, "HC???BOX*")
        quietly replace pobox=1 if strmatch(address, "HC????BOX*")
        quietly replace pobox=1 if strmatch(address, "HC?????BOX*")
*Characterise these:
        display "Total Raw Count `i'"
        count
        display "PO Boxes `i'"
        tab pobox matchcode, row column mi
        display "PO Boxes `i' Supermarkets/Grocery"
        tab pobox if supgroc==1, mi
        display "PO Boxes `i'General Merchandise"
        tab pobox if genmerch==1, mi
        display "PO Boxes `i' Convenience Stores"
        tab pobox if conv==1, mi
        display "PO Boxes `i' Pharmacies"
        tab pobox if pharm==1, mi
        display "PO Boxes `i' Liquor Stores"
        tab pobox if liquor==1, mi

*Do some housekeeping
        display "Duplicates `i' Supermarkets/Grocery"
        duplicates report address cityname statecode zip4 if supgroc==1
        display "Duplicates `i' General Merchandise"
        duplicates report address cityname statecode zip4 if genmerch==1
        display "Duplicates `i' Convenience Stores"
        duplicates report address cityname statecode zip4 if conv==1
        display "Duplicates `i' Pharmacies"
        duplicates report address cityname statecode zip4 if pharm==1
        display "Duplicates `i' Liquor"
        duplicates report address cityname statecode zip4 if liquor==1


********************************************************
********Checks to code and*****************************
********************************************************
*general merchandise and the 4 other categories:
foreach x of var  pharm liquor conv supgroc{
        display "Cross-Classifications between General Merchandise and `x' in `i'"
        count if genmerch==1 & `x'==1
        }
display "Stores Cross-Classified as General Merchandise and Pharmacy will (quietly) be made General Merchandise"
tab companyname if pharm==1 & genmerch==1
foreach j in BJ'S?WHOLESALE* COSTCO* WALMART* KMART* MEIJER* TARGET* SUPER?TARGET* WAL-MART*{
        quietly replace pharm=0 if pharm==1 & genmerch==1 & strmatch(companyname,"`j'")
        }
display "Stores Cross-Classified as General Merchandise and Liquor will (quietly) be made General
Merchandise"
tab companyname if liquor==1 & genmerch==1
foreach j in COSCO* COSTCO*{
        quietly replace liquor=0 if liquor==1 & genmerch==1 & strmatch(companyname,"`j'")
        }

* pharmacies and 3 other categories (now excluding general merchandise)
foreach x of var liquor conv supgroc{
        display "Cross-Classifications between Pharmacies and `x' in `i'"
        count if pharm==1 & `x'==1
        }

*convenient stores and 2 other categories excluding pharmacies and general merchandise
foreach x of var liquor supgroc{
        display "Cross-Classifications between Convenience and `x' in `i'"
        count if conv==1 & `x'==1
        }

*liqour and grocery stores:
display "Cross-Classifications between Liquor and Supermarkets/Groceries in `i'"
count if liquor==1 & supgroc==1


* WHAT ARE WE NOT CLASSIFYING????
*There are many not classified at this point, including all the PAFacilities
gen typestore=0
quietly replace typestore=1 if genmerch==1
quietly replace typestore=2 if liquor==1
quietly replace typestore=3 if pharm==1
quietly replace typestore=4 if conv==1
quietly replace typestore=5 if supgroc==1

label de type 0 "not classified" 1 "general merchandise" 2 "liquor" 3 "pharmacy" 4 "convenient store" 5
"grocery/super market"
label val typestore type
label var typestore "Macro store type"
display "`i' all stores by type"
tab typestore

*In this section, individual store type files are made. This is important because deduplication is done
within store types, so they have to be separated out, first.

*Supermarkets & Grocery Stores
        preserve
        quietly keep if supgroc==1
```

```
		display "`i' Supermarket/Grocery Stores"
		tab primarysic
		tab primarysicdescription
		save SupermarketsGrocery\Datasets\SupermarketsGrocery_region_`i'.dta, replace
		restore

*Pharmacy
		preserve
		quietly keep if pharm==1
		display "`i' Pharmacies"
		tab primarysic
		tab primarysicdescription
		save Pharmacies\Datasets\Pharmacies_`i'.dta, replace
		restore


*Convenience Store
		preserve
		quietly keep if conv==1
		display "`i' Convenience Stores"
		tab primarysic
		tab primarysicdescription
		save ConvenienceStores\Datasets\ConvenienceStores_`i'.dta, replace
		restore


*Liquor Stores
		preserve
		quietly keep if liquor==1
		display "`i' Liquor Stores"
		tab primarysic
		tab primarysicdescription
		save LiquorStores\Datasets\LiquorStores_`i'.dta, replace
		restore


*General Merchandise
		preserve
		quietly keep if genmerch==1
		display "`i' General Merchandise Stores"
		tab primarysic
		tab primarysicdescription
		save GeneralMerchandise\Datasets\GeneralMerchandise_`i'.dta, replace
		restore


* PA Facilities
		preserve
```

```
    quietly keep if typestore == 0
    display "`i' Unclassified Stores"
    save PAFacilities\Datasets\AllUnclassifiedInfoStores_`i'.dta, replace
    restore
}


*******
* Replace PO Box Geocodes with closest year address Geocodes . Years 2011 through 2013 street
addresses were used to replace earlier year PO Boxes.

foreach r in LiquorStores ConvenienceStores Pharmacies GeneralMerchandise {
    forvalues k = 2011/2013{
        use "`r'\Datasets\\`r'_`k'.dta", clear
        quietly keep abi latitude longitude matchcode
        rename (latitude longitude matchcode) (latitude_`k' longitude_`k' matchcode_`k')
        save "`r'\Datasets\\`r'_`k'_forpoboxes.dta", replace
        clear
        }
    forvalues j = 2007/2010{
        use "`r'\Datasets\\`r'_`j'.dta", clear
        display "Number of `r' in `j'"
        count
        quietly drop if pobox!=1
        display "Number of `r' with PO Box addresses in `j'"
        count
        tab primarysic, mi
        tab matchcode, mi
        gen goodmatch = 0
            quietly replace goodmatch = 1 if matchcode == "0" | matchcode == "P" |
matchcode == "4"
            quietly replace goodmatch = . if matchcode == ""
        tab goodmatch, mi
        quietly keep abi matchcode goodmatch
        quietly gen yearmerge=.
        quietly gen str mergematchcode=""
        forvalues k = 2011/2013{
            display "Merge with `k'"
            merge 1:1 abi using  "`r'\Datasets\\`r'_`k'_forpoboxes.dta"
            quietly drop if _merge==2
            quietly replace yearmerge= `k' if _merge==3 & yearmerge==.
            quietly replace mergematchcode=matchcode_`k' if _merge==3 &
mergematchcode==""
            drop _merge
            }
        display "PO Box `r' from `j' that were merged with later years (only earliest year merge
available is made):"
        tab yearmerge, mi
        tab matchcode mergematchcode
        tab yearmerge matchcode, mi
        tab yearmerge mergematchcode, mi
        gen mergegoodmatch = 0
            quietly replace mergegoodmatch = 1 if mergematchcode == "0" |
mergematchcode == "P" | mergematchcode == "4"
            quietly replace mergegoodmatch = . if mergematchcode == ""
        tab goodmatch mergegoodmatch, mi
        tab yearmerge goodmatch
        tab yearmerge mergegoodmatch
        quietly replace matchcode = mergematchcode if mergematchcode != ""
        quietly gen po_lat=.
        quietly gen po_long=.
        forvalues k = 2011/2013{
            quietly replace po_lat=latitude_`k' if yearmerge==`k'
            quietly replace po_long=longitude_`k' if yearmerge==`k'
            drop latitude_`k' longitude_`k'
            }
        drop yearmerge
        save "`r'\Datasets\\`r'_`j'_po_newlatlong.dta", replace
        clear
        use "`r'\Datasets\\`r'_`j'.dta", clear
            *in 2010 there is a situation where 3 different stores (all without PO boxes)
share the same abi. this doesn't affect anything except that the 1:1 merge won't work, so for
convenience stores it had to be changed to a m:1 merge.
            merge m:1 abi using "`r'\Datasets\\`r'_`j'_po_newlatlong.dta"
            quietly replace latitude=po_lat if po_lat!=.
            quietly replace longitude=po_long if po_long!=.
            drop po_lat po_long
        save "`r'\Datasets\\`r'_`j'_noboxes.dta", replace
        }
}

*naming is different for Supermarkets/Groceries, because of State/US/Regional definitions
    forvalues k = 2011/2013{
        use SupermarketGrocery\Datasets\SupermarketsGrocery_region_`k'.dta, clear
        quietly keep abi latitude longitude matchcode
        rename (latitude longitude matchcode) (latitude_`k' longitude_`k' matchcode_`k')
        save SupermarketGrocery\Datasets\SupermarketsGrocery_region_`k'_forpoboxes.dta,
replace
        clear
        }
    forvalues j = 2007/2010{
        use SupermarketGrocery\Datasets\SupermarketsGrocery_region_`j'.dta, clear
        display "Number of Supermarkets/Grocery Stores in `j'"
        count
        quietly drop if pobox!=1
        display "Number of Supermarkets/Grocery Stores with PO Box addresses in `j'"
```

```
                count
                tab primarysic, mi
                tab matchcode, mi
                gen goodmatch = 0
                        quietly replace goodmatch = 1 if matchcode == "0" | matchcode == "P" |
matchcode == "4"
                        quietly replace goodmatch = . if matchcode == ""
                tab goodmatch, mi
                quietly keep abi matchcode goodmatch
                quietly gen yearmerge=.
                quietly gen mergematchcode=""
                forvalues k = 2011/2013{
                        display "Merge with `k'"
                        merge 1:1 abi using
SupermarketGrocery\Datasets\SupermarketsGrocery_region_`k'_forpoboxes.dta
                        quietly drop if _merge==2
                        quietly replace yearmerge= `k' if _merge==3 & yearmerge==.
                        quietly replace mergematchcode=matchcode_`k' if _merge==3 &
mergematchcode==""
                        drop _merge
                        }
                display "PO Box Supermarkets/Grocery Stores from `j' that were merged with later years
(only earliest year merge available is made):"
                tab yearmerge, mi
                tab matchcode mergematchcode, mi
                tab yearmerge matchcode, mi
                tab yearmerge mergematchcode, mi
                gen mergegoodmatch = 0
                        quietly replace mergegoodmatch = 1 if mergematchcode == "0" |
mergematchcode == "P" | mergematchcode == "4"
                        quietly replace mergegoodmatch = . if mergematchcode == ""
                tab goodmatch mergegoodmatch, mi
                tab yearmerge goodmatch, mi
                tab yearmerge mergegoodmatch
                quietly replace matchcode = mergematchcode if mergematchcode != ""
                quietly gen po_lat=.
                quietly gen po_long=.
                forvalues k = 2011/2013{
                        quietly replace po_lat=latitude_`k' if yearmerge==`k'
                        quietly replace po_long=longitude_`k' if yearmerge==`k'
                        drop latitude_`k' longitude_`k'
                        }
                drop yearmerge
                save
SupermarketGrocery\Datasets\SupermarketsGrocery_region_`j'_po_newlatlong.dta, replace
                clear
                use SupermarketGrocery\Datasets\SupermarketsGrocery_region_`j'.dta, clear
```

```
                              merge 1:1 abi using
SupermarketGrocery\Datasets\SupermarketsGrocery_region_`j'_po_newlatlong.dta
                              quietly replace latitude=po_lat if po_lat!=.
                              quietly replace longitude=po_long if po_long!=.
                              drop po_lat po_long
                    save SupermarketGrocery\Datasets\SupermarketsGrocery_region_`j'_noboxes.dta,
replace
                    }


**********
*Deduplicaton
```

/* We found duplicate records in the data. We believe that data is only tagged as duplicate by InfoUSA if every field matches exactly. That means there are several possibilities
for duplicates to enter the system. The duplicates that we found and addressed were as follows:
1. Same store type with exact same address (street address, city, state, zip) but different (slightly or very) company name.
2. Same store type with exactly the same name, but slight differences in address field:
We found two different tendencies for this event:
1. Differences at the end of the address field, ex. one year the address is "100 Main Street" and the next year it is "100 Main Street #3". We deduplicated this type.
2. Slight differences in address number, eg "235 Main Street" vs. "238 Main Street". We did not deduplicate these because we had no way of ensuring that we were actually deduplicating duplicates, and it wasn't just that there were two of the same store a couple of blocks away from each other (eg CVS pharmacies on every corner)
*/

*General Merchandise

*General Merchandise is special because it gets deduplicated differently. We found with the gen merch stores that fairly often we would get multiple departments of the same store, and that sometimes they would have slightly different addresses (the assumption is that the lot can be so big and maybe there are different loading docks???). So we could not deduplicate by address alone, nor could we deduplicate by store name ("Walmart Pharmacy" and "Walmart" being different names). As a group and based on some unscientific spot checking at various locations around the country, the environmental team agreed that it would be highly unusual for a big box general merchandise store to locate within one mile of another store of the same brand (two Walmarts, or two Targets), so deduplication was carried out as follows: At this point the general merchandise dataset gets culled of bad geocodes, then it is transformed into a shapefile. Each of the records is buffered 1/2 mile. Then the buffers are dissolved by brand. This way, if there are two Walmarts in our records that are the same so they appear right next to eachother, and one is a Superwalmart and one is a Walmart Eye Center, then the buffers will overlap and it will become one polygon. Then new centroids are taken of theresulting polygon layer, and that is the deduplicated general merchandise dataset from which grids are made.

```
forvalues i = {years of interest}{
        use GeneralMerchandise\Datasets\GeneralMerchandise_`i'_noboxes.dta, clear
        quietly keep companyname genmerchmem latitude longitude matchcode count
        quietly gen brand="WALMART" if strmatch(companyname,"WALMART*") |
strmatch(companyname,"SUPER?WAL*MART") | strmatch(companyname,"WAL?MART*")
        quietly replace brand="MEIJER" if strmatch(companyname,"MEIJER*")
        quietly replace brand="KMART" if strmatch(companyname,"KMART*") |
strmatch(companyname,"SUPERKMART") | strmatch(companyname,"SUPER?KMART") |
strmatch(companyname,"SUPERK-MART") | strmatch(companyname,"SUPER?K-MART") |
strmatch(companyname,"K-MART*")
        quietly replace brand="TARGET" if strmatch(companyname,"TARGET") |
strmatch(companyname,"SUPERTARGET") | strmatch(companyname,"SUPER?TARGET") |
strmatch(companyname,"TARGET?SUPER*")
        quietly replace brand="SAMS" if strmatch(companyname,"SAM*S?CLUB")
        quietly replace brand="COSTCO" if strmatch(companyname,"COSCO*") |
strmatch(companyname,"COSTCO*")
        quietly replace brand="BJS" if strmatch(companyname,"BJ'S?WHOLESALE*") |
strmatch(companyname,"BJS?WHOLESALE*")
        display "Brands represented in `i'"
        tab brand, mi
        display "Total raw number of General Merchandise stores in `i'"
        count
        drop if matchcode!="0" & matchcode!="P" & matchcode!="4"
        display "Number of General Merchandise stores with good geocodes in `i'"
        count
        drop matchcode companyname
        export delimited using GeneralMerchandise\Datasets\GeneralMerchandise_`i'_gdcd.csv,
replace
        }

***** Convenience Stores

forvalues i={years of interest}{
use ConvenienceStores\Datasets\ConvenienceStores_`i'_noboxes.dta, clear

*de-duplicate of same addresses (usually choose the chain over the non chain)
        quietly duplicates tag address cityname statealpha digitzipcode, gen(mult_sto)
        *tab mult_sto
        sort address cityname statealpha digitzipcode
        * the following commented outline was used to determine how to deal with duplicates: list
companyname address cityname statealpha primarysic secondarysic1 if mult_sto==2, dropping those
who are non-chain if one duplicate is chain
        egen    min_chain= min(convnamechain), by(address cityname statealpha digitzipcode)
        egen     max_chain= max(convnamechain), by(address cityname statealpha digitzipcode)
        display "Total raw number of Convenience Stores in `i'"
        count
        quietly drop if mult_sto>0 & min_chain==0 & max_chain==1 & sgnamechain==0
        drop min_chain max_chain mult_sto
        quietly duplicates drop address cityname statealpha digitzipcode, force
```

```
            display "Number of Convenience Stores in `i' following deduplication for duplicates with the exact same address but different store names"
            count

*drop duplicates with same comapany name and first 3 "words" of the addres field, city, state, zip
split address, g(add)
quietly duplicates drop companyname cityname statealpha digitzipcode add1 add2 add3, force
display "Number of Convenience Stores in `i' following deduplication for duplicates with the same store name and differences in address at the end of the address. e.g. one has a unit number."
count
save ConvenienceStores\Datasets\ConvenienceStores_`i'_dedup.dta, replace
keep latitude longitude matchcode count convnamechain statealpha
quietly drop if matchcode!="0" & matchcode!="P" & matchcode!="4"
display "Number of Convenience Stores in `i', following deduplication and selection for good geocodes"
count
drop matchcode
quietly drop if statealpha == "AK" | statealpha == "HI"
display "Number of Convenience Stores in `i', following deduplication and selection for good geocodes, not including Alaska and Hawaii"
count
export delimited using ConvenienceStores\Datasets\ConvenienceStores_`i'_dedup_gdcd.csv, replace
}

***** Liquor Stores
forvalues i = {years of interest}{
            use LiquorStores\Datasets\LiquorStores_`i'_noboxes.dta,clear

*Are there any stores duplicated at location because of multiple addresses that are exactly the same?
   duplicates report companyname cityname statealpha digitzipcode address
*de-duplicate of same addresses
            display "Raw number of Liquor Stores in `i'"
            count
   quietly duplicates report address cityname statealpha digitzipcode
            quietly duplicates drop   address cityname statealpha digitzipcode, force
            display "Number of Liquor Stores in `i' following deduplication for stores with the same address, city, state, and zip but different names"
            count

*drop duplicates with same comapny name and first 3 "words" of the address field, city, state, zip
split address, g(add)
quietly duplicates drop companyname cityname statealpha digitzipcode add1 add2 add3, force
display "Number of Liquor Stores in `i' following deduplication for duplicates with the same store name and differences in address at the end of the address. e.g. one has a unit number."
count
save LiquorStores\Datasets\LiquorStores_`i'_dedup.dta, replace
keep longitude latitude count matchcode statealpha
quietly drop if matchcode!="0" & matchcode!="P" & matchcode!="4"
display "Number of Liquor Stores in `i', following deduplication and selection for good geocodes"
```

```
count
drop matchcode
quietly drop if statealpha == "AK" | statealpha == "HI"
display "Number of Liquor Stores in `i', following deduplication and selection for good geocodes, not
including Alaska and Hawaii"
count
export delimited using LiquorStores\Datasets\LiquorStores_`i'_dedup_gdcd.csv, replace
}

****** Pharmacies
forvalues i={years of interest}{
use Pharmacies\Datasets\Pharmacies_`i'_noboxes.dta, clear

*de-duplicate of same addresses (usually choose the chain over the non chain)
        quietly duplicates tag address cityname statealpha digitzipcode, gen(mult_sto)
        *tab mult_sto
        sort address cityname statealpha digitzipcode
        *list companyname address cityname statealpha primarysic secondarysic1 if mult_sto==2
        *dropping those who are non-chain if one duplicate is chain
        egen    min_chain= min(phnamechain), by(address cityname statealpha digitzipcode)
        egen     max_chain= max(phnamechain), by(address cityname statealpha digitzipcode)
        display "Total raw number of Pharmacies in `i'"
        count
        quietly drop if mult_sto>0 & min_chain==0 & max_chain==1 & sgnamechain==0
        drop min_chain max_chain mult_sto
        quietly duplicates drop address cityname statealpha digitzipcode, force
        display "Number of Pharmacies in `i' following deduplication for duplicates with the exact same
address but different store names"
        count

*drop duplicates with same company name and first 3 "words" of the addres field, city, state, zip
split address, g(add)
quietly duplicates drop companyname cityname statealpha digitzipcode add1 add2 add3,force
display "Number of Pharmacies in `i' following deduplication for duplicates with the same store name
and differences in address at the end of the address. e.g. one has a unit number."
count
save Pharmacies\Datasets\Pharmacies_`i'_dedup.dta, replace
keep latitude longitude count phnamechain matchcode statealpha
quietly drop if matchcode!="0" & matchcode!="P" & matchcode!="4"
display "Number of Pharmacies in `i', following deduplication and selection for good geocodes"
count
drop matchcode
quietly drop if statealpha == "AK" | statealpha == "HI"
display "Number of Pharmacies in `i', following deduplication and selection for good geocodes, not
including Alaska and Hawaii"
count
export delimited using Pharmacies\Datasets\Pharmacies_`i'_dedup_gdcd.csv, replace
}
```

```
*****Supermarkets/Groceries
forvalues i={years of interest}{
use SupermarketGrocery\Datasets\SupermarketsGrocery_region_`i'_noboxes.dta, clear
        quietly duplicates tag address cityname statealpha digitzipcode, gen(mult_sto)
        *tab mult_sto
        sort address cityname statealpha digitzipcode
        *list companyname address cityname statealpha primarysic secondarysic1 if mult_sto==2
        *dropping duplicates  with min classification for type_region
        egen    min_chain= min(type_reg), by(address cityname statealpha digitzipcode)
        egen     max_chain= max(type_reg), by(address cityname statealpha digitzipcode)
        display "Total raw number of Supermarkets/Grocery Stores in `i'"
        count
        drop if mult_sto>0 & min_chain<max_chain & min_chain!=. & type_reg<max_chain
        drop min_chain max_chain mult_sto
        duplicates drop address cityname statealpha digitzipcode, force
        display "Number of Supermarkets/Grocery Stores in `i' following deduplication for duplicates
with the exact same address but different store names"
        count

split address, g(add)
*drop duplicates with same comapny name and first 3 "words" of the addres field, city, state, zip
quietly duplicates drop companyname cityname statealpha digitzipcode add1 add2 add3, force
display "Number of Supermarkets/Grocery Stores in `i' following deduplication for duplicates with the
same store name and differences in address at the end of the address. e.g. one has a unit number."
count
save SupermarketGrocery\Datasets\SupermarketsGrocery_region_`i'_dedup.dta, replace
keep latitude longitude type_reg matchcode count statealpha
quietly drop if matchcode!="0" & matchcode!="P" & matchcode!="4"
display "Number of Supermarkets/Grocery Stores in `i', following deduplication and selection for good
geocodes"
count
drop matchcode
quietly drop if statealpha == "AK" | statealpha == "HI"
display "Number of Supermarkets/Grocery Stores in `i', following deduplication and selection for good
geocodes, not including Alaska and Hawaii"
count
export delimited using
SupermarketGrocery\Datasets\SupermarketsGrocery_region_`i'_dedup_gdcd.csv, nolabel replace
}



cap log close
```

Measure Construction – Dun and Bradstreet

```
cap log close
clear all
set more off

forvalues i = {years of interest}{
        use {geocoded input data}, clear

        *Some SIC codes longer than 8 digits, keep first 8 only
        *Very important to keep format the same or number change
        gen     double primarysic= us1987sic1
        format      primarysic %25.0g
        tostring    primarysic, gen(prim) force format(%25.0g)

        bysort year: count if length(prim)>8
        gen only8=(length(prim)==8)

        gen prim8digit=substr(prim,1,8)
        destring prim8digit, replace

        gen prim8digits=substr(prim,1,8)

        tab matchcode, mi

        duplicates tag dunsnumber, gen(duplicateduns)

        gen storetype = .
        foreach j in MCDONALD* MC?DONALDS* MC?DONLDS *MCDONALDS?F*
MACDONALDS?IN?WALMART BUFFALO?MCDONALDS?INC* *TACO?BELL* TACCO?BELL WENDYS*
WENDY?0472091 WENDY?OL*FASHION*HAMBURG* WENDY?S* WENDY&APO* 22640?WENDYS*
*BURGER?KING* BUGER?KING* KENTUCKY?FRIED?CHICKEN* KENTUCKY?FRIED?CHKN* KFC* K?F?C*
K?FC KCF KENTUCKY?FRD?CHICKEN* KENTUCKY?FRIED?CHICHEN KENTUCKY?FRIED?CHICKEN*
KENTUCKY?FRIED?CHKN*{
                quietly replace storetype = 1 if strmatch(tradestylename,"`j'") & storetype == .
                quietly replace storetype = 1 if strmatch(businessname,"`j'") & storetype == .
                }

        foreach j in *SUBWAY* SUB?WAY*{
                quietly replace storetype = 2 if strmatch(tradestylename,"`j'") & storetype == .
                quietly replace storetype = 2 if strmatch(businessname,"`j'") & storetype == .
                }

        foreach j in ARBY* ARBY?S* *ARBYS?C?ARG?SOUTH *ARBY?C?US*
BAJA?FRESH?MEXICAN?GRILL* BAJA?FRESH?MEX?GRILL* BLIMPIE?SUBS BLIMPIE* *BLIMPIE BLIMPY*
BLIMBIE?SUBS?AND?SALADS BLIMPE?SUB* BLIMPEE?SUBS* BLIMPEES BLIMPI?EXPRESS BLIMPI?SUB*
```

```
BHMPIES?SUBS?&?SALADS BOJANGLES BO?JANGLE BOJANGLE* BOSTON?MARKET* CAPTAIN?DS
CAPTAIN?DS?SEAFOOD CARL?JR* CARL?S?JR* CARLS?JR* CARL?KARCHER* CARLS?1101481
CARLS?1410?7535 CARLSJR CARLS CARLS?JUNIOR* CHECKERS* CHICK?FIL?A* CHIC?FIL?A* CHIPOTLE*
CHIPOLTE* CHURCH?S?CHICKEN CHURCHS?CHICKEN* CHURCHS?FRIED?CHICKEN* CULVER*S*
DAIRY?QUEEN* DQ?GRILL*CHILL DEL?TACO* EINSTEIN?BROS?BAGELS EINSTEIN?BROTHERS?BAGELS
EL?POLLO?LOCO* FIVE?GUYS* 5?GUY?BURGER* 5?GUYS?BURGER* 5?GUY?FAMOUS?BURGER*
5?GUYS?FAMOUS?BURGERS* 5GUYS* 5?GUYS?DIBERVILLE?MS *HARDEES* HARDYS* HARDIES IN-N-
OUT* IN?&?OUT?BURGER* IN?&?OUT?DRIVEIN IN?N?OUT?BURGER* JACK?IN?THE?BOX* JACK-N-TH-
BOX?DRV-THRU* JACK?IN?BOX* JASONS?DELI* JASON?DELI JASON?S?DELI*
JIMMY?JHNES?GOURMET?SANDWICH* JIMMY?JHNS?GOU?MET?SANDWICH* JIMMY?JOHN*
JIMMY?JONS?GOURMET?SANDWICH* JIMMY?JHNS?GRMET?SNDWICH?SHOPS
JIMMYS?JHNS?GOURMET?SANDWICHES IMMY?JOHNS?GOURMET?SANDWICHES KRYSTAL*
*LONG?JOHN?SILVER* LONJ?JOHN?SILVERS LONGJOHN?SILVERS* LONG?JOHN?SLVERS*
MOES?SOUTH?WEST?GRILL MOE?S?SOUTH*WEST?GRILL* MOE?S?SW?GRILL* MOES?SW?GRILL*
MOE?SOUTHWEST?GRILL* MOES?SOUTH?GRILL MOES?SOUTHWEST* MOES?TO?GO*
MOESS?SOUTHWEST?GRILL MOESTOGO?CORPORATION MOS?SOUTHWEST?GRILL PANDA?EXPRESS*
PANDA?PANDA?EXPRESS?INC PANERA* PANERO?BREAD POPEYES?CHICKEN?&?BISCUITS
POPEYES?CHICKEN* POPEYES?CHKN?&?BISCUITS* POPEYES?FRIED* POPEYES?LOUISIANA?KITCHEN*
POPEYES?JANJ?138?K POPEYE?CHICKENS?AND?BISCUITS POPEYE?S* POPEYES?FRD?CHKN
POPEYES?N?BRUNSWICK POPEYS?OMNI* POPEYES POPYES???TRANSFARE* POPEYES?1* POPEYES?3*
POPEYES?4* POPEYES?5* POPEYES?7* POPEYES?8* POPEYES?9* POPEYES?DOWNTOWN*
POPEYES?FAMOUS* QBODA* QDOBA* QDOVA *QUIZNO* QUITNOS* QUIZ?NOS* QUIZCOS* QUIZINOS
QUIZMOS* QUIZONS* QUIZZNOS* QUZINOS* GUIZNOS?CLASSIC?SUBS QUINOS*
QUINZNOS?CLASSIC?SUBS QUINZO?S?SUBS QUINZOS* QUISNOS* QUIVNOS?SUBS
QUIZ?SUBS?CORPORATION QUIZANO?S?SUBS QUIZINO?SUB QUIZN?S?2516 QUIZNES QUIZNISSUB
QUIZONES?SUB QUIZZINOS?INC QULZNOS* QUTZNOS?CLASSIC?SUBS QUZNIOS QUZNOS?S?B
*SBARRO* SONIC* SONI SONID?DRIVE-INN?OF?JASPER SONIE?DRIVE-IN WHATABURGER*
WHAT?A?BURGER* WHATA?BURGER* WHATABRGER?CNTL?HSTON?LRNG?CTR WHITE?CASTLE*
WINGSTOP* WING?STOP* ZAXBY*{
                quietly replace storetype = 3 if strmatch(tradestylename,"`j'") & storetype == .
                quietly replace storetype = 3 if strmatch(businessname,"`j'") & storetype == .
                }

        foreach j in STARBUCKS* DUNKIN?DONUTS* DUNKIN??DONUTS* DUNKI?DONUTS
TIM?HORTON* TIM?S?HORTON{
                quietly replace storetype = 4 if strmatch(tradestylename,"`j'") & storetype == .
                quietly replace storetype = 4 if strmatch(businessname,"`j'") & storetype == .
                }

        foreach j in  CALIFORNIA?PIZZA?KITCHEN CICIS?PIZZA CICI?ENTERPRISES?INC DOMINO?PIZZA*
DOMINOS?PIZZA* LITTLE?CAESARS* LITTLE?CAESAR* *LITTLECAESARS* PAPA?JOHNS*
PAPA?MURPHYS* PIZZA?HUT*{
                quietly replace storetype = 5 if strmatch(tradestylename,"`j'") & storetype == .
                quietly replace storetype = 5 if strmatch(businessname,"`j'") & storetype == .
                }
```

```
    *Remaining Chain Pizza as classified by D&B sic code
    quietly replace storetype = 6 if prim8digit==58120601 & storetype == .
    *Non Chain Pizza
    quietly replace storetype = 7 if (prim8digit==58120602 | prim8digit==58120600) & storetype ==
.
    *Remaining Coffee Shops
    quietly replace storetype = 8 if prim8digit==58120304 & storetype == . & new == 0
    *All else
    quietly replace storetype = 9 if storetype == .


    tab storetype new, mi

    *drop full-service restaurants and convenience stores
    gen conv = .
    foreach j in 76 2?GO?MART ADMIRAL?PETROLEUM ALLSUP* AMOCO AMPM A?PLUS ARCO
BIG?APPLE BIG?JOHN'S BIG?JOHNS BP* BREAK?TIME BUCKYS?EXPRESS BUCKY'S?EXPRESS
CASEYS?GENERAL?STORE CASEY'S?GENERAL?STORE CEFCO?FOOD?STORE CENEX CERTIFIED CHEVRON*
CHOICE CHUCK*S?FOOD?STORE CHUCK'S?FOOD?STORE CIRCLE?K CITGO CLARK
COAST?GUARD?MINI?MART COASTAL CONOCO CONVENIENT?FOOD?MART CORNER?STORE
COUNTRY?FAIR CUMBERLAND?FARMS DAILY*S?EXPRESS DAILY'S?EXPRESS DAILYS?TRI?STAR?ENERGY
DAILY'S?TRI?STAR?ENERGY DAIRY?MART DANDY?MINI?MART DISCOUNT?FOOD?MART
DUCHESS?SHOPPE E*Z*MART EDDYS?MART EDDY'S?MART EXPRESS?LANE EXXON* FAMILY?FARE
FARM?STORE FAS?MART FAST?FOOD*FUEL FAST?STOP FIVE?STAR?FOOD?MART FLASH?FOODS
FLASH?MARKET FLYING?J FOOD?FAST?STORE FREEDOM?VALU?CENTER GAS?EXPRESS GAS?MART?USA
GATE?FOOD?POST GETGO GETTY GIANT GO?MART?FOOD?STORE GOASIS GULF HANDEE?HUGO'S
HANDEE?HUGOS HANDI* HOLIDAY?STATIONSTORES HOWDYS?FOODMART HOWDY'S?FOODMART
HUCK'S HUCKS HY?VEE?GAS* JACKSONS?FOOD?STORE JOES?KWIK?MART JOE'S?KWIK?MART
JUMPIN*?JIMMY*S?GAS?MART KANGAROO?EXPRESS KRAUSZER*S?FOOD?STORE KUM?&?GO
KWIK?FARMS KWIK?FILL KWIK?PIK KWIK?SHOP KWIK?STAR KWIK?STOP KWIK?TRIP
LI*L?CRICKET?STORES LITTLE?GENERAL LOVE'S?COUNTRY?STORE LOVES?COUNTRY?STORE
LOVE'S?TRAVEL?STOP LOVES?TRAVEL?STOP LUKOIL MAPCO?EXPRESS MAPCO?MART
MARINE?CORPS?SHOPPETTE MAVERIK MEIJER?GAS?STATION MFA?OIL MOBIL MOBIL?MART
MOTO?MART MURPHY?EXPRESS MURPHY?USA MUSTANG NEXCOM?MINI?MART NEXT?DOOR?STORES
NICE?N?EASY?GROCERY?SHOPPE OPTIMA PETRO?CARD?24 PETRO?EXPRESS PETRO?STOPPING?CENTER
PHILLIPS?66 PILOT?FOOD?MART PILOT?TRAVEL?CENTER PLAID?PANTRY PRESTO QUICKCHEK QUIKTRIP
RACETRAC RACEWAY RED?APPLE ROAD?RANGER ROADRUNNER?MARKET ROYAL?FARMS
SAM'S?FOOD?STORE SAMS?FOOD?STORE SCOT?MARKET SCOTCHMAN SEVEN?ELEVEN SHEETZ SHELL
SHOR?STOP SINCLAIR SPEEDWAY SPINX?STORE SPRINT?MART STAR?STOP STEWARTS?SHOP
STEWART'S?SHOP STOP?IN STRIPES SUNDIAL?DELI?MART SUNMART SUNOCO SUPER?AMERICA*
TAYLOR?PETROLEUM TEDESCHI?FOOD?SHOP TERRIBLE?HERBST TESORO TEXACO THORNTON'S*
THORNTONS* TIMEWISE?FOOD?STORE TOM?THUMB TOWN?PUMP?FOOD?STORE TROOP?STORE
TRUE?NORTH TURKEY?HILL?MINIT?MART UNI?MART UNIMART UNITED?DAIRY?FARMERS
USA?FUEL?CENTER VILLAGE?PANTRY WAWA WILCO?FOOD?MART WILCO?TRAVEL?PLAZA XTRA?MART
YOUNG'S YOUNGS ZIP?TRIP{
                quietly replace conv = 1 if strmatch(businessname,"`j'")==1   & storetype > 6
                quietly replace conv = 1 if strmatch(tradestylename,"`j'")==1 & storetype > 6
                }
    *browse businessname tradestylename if conv==1
    count if conv == 1
    quietly drop if conv == 1

    gen fullservice = .
    foreach j in APPLEBEE* APPLE?BEES?N* APPLEEBEES* BAHAMA?BREEZE BAR?LOUIE*
BEEF*BRADYS* BEEF*BRADY?S* BENIHANA* BENNIGAN* BERTUCCIS* *FRISCH* BIG?BOY?255 BJS*
BLACK?ANGUS* BOB?EVANS* BONEFISH* BONE?FISH?GRILL* BRAVO?CUCINA?ITALIANA
BRAVO?BRIO?REST?GROUP?INC BRAVO?DEVELOPMENT?INC BRIO?TUSCAN?GRILLE BRIO?115
BRIO?PIZZERIA?AND?RESTAURANT BRIO?TSCAN?GRLLE* BRIO?TUSCAN?GRILL* BRIO?TUSCHAN?GRILL
BUBBA?GUMP* BUCA?DI?BEPPO *BUFFALO?WILD?WING* BUFFLO?WILD?WINGS
BUFFOLA?WILD?WINGS FFALO?WILD?WINGS* BUFALO?WILD?WINGS?GRILL?AND?BA
BUFFALO?WILD?GRILL* BUFFALO?WILD?WI BUFFALO?WILD?WILD?WINGS BUFFALO?WILD?WNGS*
BUFFALO?WILDS?WINGS BUFFALO?WILDWINGS BUFFALO?WILS?WINGS* *CAPITAL?GRILLE*
CAPITAL?GRILL *CARINOS* CARINO?ITAL?GRILL CARRABA* CARRABBA* CHAMPPS*
CHAMPS?SPORT?BAR* CHAMPS?SPORTS?BAR* CHAMPS?SPORTSBAR* CHAMPS?BAR*GRILL*
CHEDDARS CHEDDAR?GRILL CHEVYS?FRESH?MEX* CHILIS* CHILI?S?950 CHILI?S?GRILL* CHUYS CHUY?S
CLAIM?JUMPER COCOS CRACKER?BARREL* DAVE*BUSTERS* DENNYS* EAT?N?PARK EL?TORITO
EL?TORITO?MEXICAN?GRILL* ELEPHANT?BAR?RESTAURANT FAMOUS?DAVES* FLEMINGS?PRIME*
FRIENDLYS?ICE?CREAM FRIENDLYS FRIENDLYS?REST?FRANCHISE FUDDRUCKERS* GOLDEN?CORRAL
GRAND?LUX?CAFE HARD?ROCK?CAFE HOMETOWN?BUFFET HOOTERS* HOULIHANS
HOULAHANS?TAVERN*GRILL HOUSTONS HUDDLE?HOUSE* HUDDLE HUDDLE?HOULE HUDDLE?H IHOP
INTERNATIONAL?HOUSE?OF?PANCAKES INTERNATIONAL?HOUSE?PANCACKES J?ALEXANDERS*
JOES?CRAB?SHACK* LEGAL?SEA?FOOD* LOGANS?ROADHOUSE* LONE?STAR?STEAKHOUSE*
LONGHORN?STEAKHOUSE LUBYS?CAFETERIA MAGGIANOS* MARIE?CALLENDARS* MAX*ERMAS*
MCCORMICK*SCHMICKS MIMIS?CAFE* MORTONS* NINETY?NINE* 99* OCHARLEYS* OLD?CHICAGO
OLD?CHICAGO?DBA OLD?CHICAGO?EDEN?PRAIRIE ROCK?BOTTOM?INC OLD?COUNTRY?BUFFET
OLIVE?GARDEN ON?THE?BORDER?MEX* ON?THE?BRDER?MXCAN* ON?THE?BORDER?CORPORATION
OUTBACK?STEAKHOUSE OUTBACK?STEAK?HSE?REGIONAL?OFF PF?CHANGS* PERKINS*
PICADILLY?CAFETERIA PONDEROSA?STEAKHOUSE PONDERSA QUAKER?STEAK*LUBE RAINFOREST?CAFE
RED?LOBSTER RED?ROBIN* RED?ROBBIN?GOURMET?BURGERS RED?ROBN?AMRCS?GRMT?BRGR*
RED?ROBN?GOURMET* ROMANOS?STRONGSVILLE ROMANOS ROMANOS?DRIVE?IN
ROMANOS?MACARON* ROMANOS?MACARRONI?GRILL RUBY?TUESDAY RUTHS?CHRIS?STEAK?HOUSE
RYANS?GRILL* SALTGRASS* SEASONS?52 SEASONS?GRILL* SHARIS?RESTAURANT* SHONEYS SIZZLER
SIZZLER?USA?RESTAURANTS?INC SMOKEY?BONES* SONNYS?REAL?PIT* STEAK?N?SHAKE
SWEET?TOMATOES SWEET?TOMATO TGIF* TGIFRIDAYS TGI?FRIDAY* TEXAS?ROADHOUSE
*CHEESECAKE?FACTORY *MELTING?POT UNO?CHICAGO*GRILL UNOS?CHICAGO*GRILL PIZZERIA?UNO
VILLAGE?INN WAFFLE?HOUSE* YARD?HOUSE{
        quietly replace fullservice = 1 if strmatch(businessname,"`j'")==1    & storetype > 6
        quietly replace fullservice = 1 if strmatch(tradestylename,"`j'")==1  & storetype > 6
        }

    count if fullservice == 1
    quietly drop if fullservice==1

    tab storetype new, mi
```

```
	la def FFRLabels 1 "Top 5 Non-Sandwich, Non-Coffee Chains" 2 "Subway" 3 "Top Chains,
Excluding Top 5, Coffeeshops, and Pizza" 4 "Top Chain CoffeeShops" 5 "Large Chain Pizza" 6 "Chain Pizza,
SIC: 58120601" 7 "Non-Chain Pizza, SIC: 58120602 or 58120600" 8 "Non-Chain CoffeeShops, SIC:
58120304" 9 "FFR, Not Elsewhere Classified"
	la val storetype FFRLabels




	save FF_class_`i'.dta,replace

	*count variable necessary for grids generation
	gen count = 1
	*Geocoding level designators differed in 2014
	tab matchcode if year < 2014
	tab acccode if year < 2014
	tab accuracy if year < 2014
	count if (matchcode == "T" | matchcode == "Z" | matchcode == "C" | accuracy == "City
Centroid" | accuracy == "City Centroid" | accuracy == "Zip+2" | accuracy == "Zipcode" | accuracy ==
"Zipcode Centro") & year < 2014
	drop if (matchcode == "T" | matchcode == "Z" | matchcode == "C" | accuracy == "City  Centroid"
| accuracy == "City Centroid" | accuracy == "Zip+2" | accuracy == "Zipcode" | accuracy == "Zipcode
Centro") & year < 2014

	count if (accuracy == "City  Centroid" | accuracy == "City Centroid" | accuracy == "Zip+2" |
accuracy == "Zipcode" | accuracy == "Zipcode Centro") & year==2014
	drop if (accuracy == "City  Centroid" | accuracy == "City Centroid" | accuracy == "Zip+2" |
accuracy == "Zipcode" | accuracy == "Zipcode Centro") & year==2014
	export delimited storetype count latitude longitude using "FFR_`i'_forGrid.csv", replace
	clear
}
```