



Weight and Veterans' Environments Study

Distance to nearest environmental setting: Weight and Veterans' Environments Study GIS protocol

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



UIC Neighborhoods + Health

Overview

This protocol describes the construction of a set of measures representing the Euclidean distance to the five closest resources of a variety of types from participants' home addresses.

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. Haytham Abu Zayd, Shubhayan Ukil and Abby Klemp helped to edit the protocol for dissemination.

Suggested Citation

Xiang W, Jones K, Matthews SA, Zenk SN. (2018). Distance to nearest environmental setting: Weight and Veterans' Environments Study GIS protocol, Version 1. Retrieved from Weight and Veterans' Environments Study website: <https://waves.uic.edu/>.

Table of Contents

Overview	2
Acknowledgements.....	2
Suggested Citation	2
Background	4
Data.....	5
Sources.....	5
Cleaning.....	5
Generate Near Table using ArcGIS and Python	5
Export Near Table to CSV	5
Post-GIS processing using Stata	5
Software.....	5
Appendix	6
Scripts.....	6
A: Closest Resource	6
B: Dbf to file text.....	8
C: Do file	10

Background

This document describes the process of constructing closest resources measures. The measure is defined as the distances of the closest 5 resources of the same type from the veteran's home location to the resource (e.g. 5 of the closest supermarkets, 5 of the closest chain fast food restaurants, etc.). This distance is calculated as a straight line distance between the veteran's home point and the resource.

For WAVES, the nearest distances were calculated for the following resource types:

1. All supermarkets
2. Chain supermarkets
3. Non-chain supermarkets
4. Convenience stores
5. Non-member general merchandise
6. All fast food restaurants no coffee
7. Chain fast food restaurants
8. Non chain fast food restaurants
9. Grocery
10. PA facilities
11. All parks

Only the All parks category is polygon, all other resources are points.

Data

Sources

The resources files are from multiple sources: InfoUSA (stores data), Dun & Bradstreet (fast food data), Navteq and TeleAtlas (parks data).

Cleaning

For reference in our scripts, all the files of a resource must be stored singly by year in a geodatabase. For example, all 2008 supermarkets should be in one geodatabase and all 2009 supermarkets should be in a separate geodatabase.

Decisions

Generate Near Table using ArcGIS and Python

To do the distance analysis, ArcGIS tool "Generate Near Table" is used in a Python script, given in [Appendix A](#).

The output from the Python script are the "near tables" which are stored in the GIS geodatabases by the resource type and the year.

Export Near Table to CSV

Then the tables in the geodatabases are exported as .csv files. Python script used to export the .csv files is given in [Appendix B](#).

Post-GIS processing using Stata

After the .csv files have been generated, the post-GIS processing is done in Stata. The DO file is given in [Appendix C](#).

Software

The distance calculation is done in ArcGIS 10.3 using "Generate Near Table" tool in the Python scripting environment. The post-GIS analysis is done in Stata/MP 14.1.

Appendix

Scripts

A: Closest Resource

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

```
import arcpy, sys, os, time
from arcpy.sa import *

#Run near table for each year with different resources
year = "FY2015"
resourceYear_workspace= {insert workspace location}
vet_workspace = {insert workspace location}
ChainFFR_dir = {insert file location}
AllFFRNoCoffee_dir = {insert directory location}
AllSuperMarkets_dir = {insert directory location}
ChainSuperMarkets_dir = {insert directory location}
ConvenienceStores_dir = {insert directory location}
NonMemberGenMerch_dir = {insert directory location}
NonChainFFR_dir = {insert directory location}
NonChainSup_dir = {insert directory location}
Grocery_dir = {insert directory location}
PAFAC_dir = {insert directory location}
Park_dir = {insert directory location}

# Generate near Table

arcpy.env.workspace = resourceYear_workspace
resourceList = arcpy.ListFeatureClasses()

for count2,resource in enumerate(resourceList):
    #if resource.find("NonChainFFR")>-1 or resource.find("NonChainSup")>-1 or
    resource.find("Grocery")>-1 or resource.find("PAFAC")>-1:
        if resource.find("Allpark")>-1:
            arcpy.env.workspace = resourceYear_workspace
            resource_lyr = "resourcelyr" + str(count2)
            arcpy.MakeFeatureLayer_management(resource,resource_lyr)
            near_feature = resource_lyr
            print resource
            arcpy.env.workspace = vet_workspace
            fList = arcpy.ListFeatureClasses()
```

```

for count,feature in enumerate(fList):
    arcpy.env.workspace = vet_workspace
    lyr = "lyr" + str(count)
    arcpy.MakeFeatureLayer_management(feature,lyr)
    in_feature = lyr
    if resource.find("ChainFFR")>-1:
        outTable_workspace = ChainFFR_dir
    if resource.find("AllFFRNoCoffee")>-1:
        outTable_workspace = AllFFRNoCoffee_dir
    if resource.find("AllSuperMarket")>-1:
        outTable_workspace = AllSuperMarkets_dir
    if resource.find("ChainSuperMarket")>-1:
        outTable_workspace = ChainSuperMarkets_dir
    if resource.find("ConvenienceStores")>-1:
        outTable_workspace = ConvenienceStores_dir
    if resource.find("GenMerch")>-1:
        outTable_workspace = NonMemberGenMerch_dir
    #added 1/18/17
    if resource.find("NonChainFFR")>-1:
        outTable_workspace = NonChainFFR_dir
    if resource.find("NonChainSup")>-1:
        outTable_workspace = NonChainSup_dir
    if resource.find("Grocery")>-1:
        outTable_workspace = Grocery_dir
    if resource.find("PAFAC")>-1:
        outTable_workspace = PAFAC_dir
    if resource.find("Allpark")>-1:
        outTable_workspace = Park_dir

    out_table = outTable_workspace + feature + "_nearTable"
    search_radius = ""
    location = "NO_LOCATION"
    angle = "NO_ANGLE"
    closest = "ALL"
    closest_count = 5
    #generate near table

arcpy.GenerateNearTable_analysis(in_feature,near_feature,out_table,search_radius,location,angle,closest,closest_count)
    print count

'''
except:

```

```
print arcpy.GetMessages()
'''
```

B: Dbf file to text

#This script rename table fields in a batch

```
import arcpy,sys, os, time, csv
from arcpy.sa import *
```

```
year = "FY2015"
directory = {insert file location}
```

```
'''
```

```
NonChainFFR = "BG_Vets_wClosestResources2_NonChainFFR.gdb"
NonChainSuperMarkets = "BG_Vets_wClosestResources2_NonChainSuperMarkets.gdb"
AllPark = "BG_Vets_wClosestResources2_AllPark.gdb"
PAFAC = "BG_Vets_wClosestResources2_PAFAC.gdb"
Grocery = "BG_Vets_wClosestResources2_Grocery.gdb"
```

```
AllFFR = "BG_Vets_wClosestResources_AllFFRNoCoffee.gdb"
AllSup = "BG_Vets_wClosestResources_AllSuperMarkets.gdb"
ChainFFR = "BG_Vets_wClosestResources_ChainFFR.gdb"
'''
```

```
ChainSup = "BG_Vets_wClosestResources_ChainSuperMarkets.gdb"
Conv = "BG_Vets_wClosestResources_ConvenienceStores.gdb"
'''
```

```
NonMemGenMerch = "BG_Vets_wClosestResources_NonMemberGenMerch.gdb"
'''
```

```
#gdbname = [NonChainFFR,NonChainSuperMarkets,AllPark,PAFAC,Grocery]
```

```
#gdbname = [Conv,NonMemGenMerch]
```

```
#gdbname = [NonMemGenMerch]
```

```
gdbname = [ChainSup,Conv]
```

```
csv_directory = {insert file location}
```

```
'''
```

```
NonChainFFR_csv = "NonChainFFR"
NonChainSuperMarkets_csv = "NonChainSuperMarkets"
AllPark_csv = "AllPark"
PAFAC_csv = "PAFAC"
Grocery_csv = "Grocery"
'''
```



```

#foldername =
[NonChainFFR_csv,NonChainSuperMarkets_csv,AllPark_csv,PAFAC_csv,Grocery_csv]
#foldername = ["ConvenienceStores","NonMemberGenMerch"]
#foldername = ["NonMemberGenMerch"]
foldername = ["ChainSuperMarkets","ConvenienceStores"]

for count,in_workspace in enumerate(gdbname):
    start = time.time()
    arcpy.env.workspace = directory+in_workspace
    print arcpy.env.workspace
    out_workspace = csv_directory+foldername[count]+"\"
    print out_workspace
    arcpy.env.overwriteOutput = True
    tableList = arcpy.ListTables()
    tableList.sort()
    for table in tableList:
        fieldList = arcpy.ListFields(table) #get a list of fields for each feature class
        del fieldList[0]#delete OBJECTID no need to write to output
        #del fieldList[1]#delete NEAR_FID field
        field_names = [field.name for field in fieldList]
        near_dist_field = "NEAR_DIST_"+foldername[count]
        field_names[2] = near_dist_field
        outfile = out_workspace + table + ".csv"
        #open output csv file
        if os.path.isfile(outfile) is not True:
            f = open(outfile,'wb')
            w = csv.writer(f)
            #write field names to the output file
            w.writerow(field_names)
            for row in arcpy.SearchCursor(table):
                field_vals = []
                for field in fieldList:
                    if row.getValue(field.name)is not None:###this is very important
                        fieldValue = row.getValue(field.name)
                    else:
                        fieldValue = 0
                    #print fieldValue
                    field_vals.append(fieldValue)
                #field_vals = [row.getValue(field.name) for field in fieldList]
                #print field_vals
                w.writerow(field_vals)
            del row

```

```
print table
f.close()
```

```
end = time.time()
print "time elapsed for this vet point file ....."
print str((end-start)/60) + " minutes"
```

C: Do-File

* DO File to process Closest Facilities Near Tables

*December 2015

/*

Each of the five closest facilities by facility type were identified with distance to each using the near table tool in arcgis.

Near table joins in_fid to near_fid. in_fid must then be merged to the original input file to determine studyid.

*/

```
clear all
cap log close
set more off
cd [set to workspace]
```

*Convert near table csv files to stata after merging for studyid

```
foreach year in 2009 2010 2011 2012 2013 2014 2015{
```

```
    foreach type in AllFFRNoCoffee AllSuperMarkets ChainFFR ChainSuperMarkets
```

```
    ConvenienceStores NonMemberGenMerch AllPark Grocery NonChainFFR NonChainSuperMarkets
```

```
    PAFAC{
```

```
        forvalues i = 1/44{
```

```
            import {near tables}
```

```
            merge m:1 in_fid using {participant data files}
```

```
            drop in_fid near_fid _merge xcoord ycoord
```

```
            rename year_year
```

*There are cases where there are ties: near_rank and near_dist are the same for more than one observation. This needs to be amended before reshaping is possible.

```
            sort studyid near_dist
```

```
                *first observation of this studyid
```

```
                replace near_rank = 1 if studyid[_n] != studyid[_n-1]
```

```
                *same studyid and year as previous observation
```

```
                replace near_rank = near_rank[_n-1]+1 if studyid[_n] == studyid[_n-1]
```

```
            *Reshape Long to Wide
```

```
            reshape wide near_dist, i(studyid year) j(near_rank)
```

```
            save ClosestResources\\`type'\ClosestResources_`year'_'`type'_'_pt`i'.dta, replace
```

```

        clear
      }
    }
  }

```

*Append all parts together

```

foreach type in AllFFRNoCoffee AllSuperMarkets ChainFFR ChainSuperMarkets ConvenienceStores
NonMemberGenMerch AllPark Grocery NonChainFFR NonChainSuperMarkets PAFAC{
  use ClosestResources\\`type'\ClosestResources_2009_`type'_pt1.dta
  forvalues i = 2/44{
    append using ClosestResources\\`type'\ClosestResources_2009_`type'_pt`i'.dta
  }
  foreach year in 2010 2011 2012 2013 2014 2015{
    forvalues i = 1/44{
      append using ClosestResources\\`type'\ClosestResources_`year'_`type'_pt`i'.dta
    }
  }
  save ClosestResources\\`type'\ClosestResources_`type'_All.dta, replace
}

```

*This merges all the individual resource files into one

```

use ClosestResources\AllFFRNoCoffee\ClosestResources_AllFFRNoCoffee_2009_2015.dta
foreach type in AllSuperMarkets ChainFFR ChainSuperMarkets ConvenienceStores
NonMemberGenMerch AllPark Grocery NonChainFFR NonChainSuperMarkets PAFAC{
  merge 1:1 studyid year using ClosestResources\\`type'\ClosestResources_`type'_2009_2015.dta
  rename _merge merge_`type'
}

```

```

foreach type in allffrnocoffee allsupermarkets chainffr chainsupermarkets conveniencestores
nonmembergenmerch allpark grocery nonchainffr nonchainsupermarkets pafac{
  gen meandist_`type' = (near_dist_`type'1 + near_dist_`type'2 + near_dist_`type'3 +
near_dist_`type'4 + near_dist_`type'5)/5
}

```

```

foreach type in allffrnocoffee allsupermarkets chainffr chainsupermarkets conveniencestores
nonmembergenmerch allpark grocery nonchainffr nonchainsupermarkets pafac{
  la var studyid "Studyid"
  la var year "Veteran year"
  la var near_dist_`type'1 "Distance in meters to closest `type'"
  la var near_dist_`type'2 "Distance in meters to second closest `type'"
  la var near_dist_`type'3 "Distance in meters to third closest `type'"
  la var near_dist_`type'4 "Distance in meters to fourth closest `type'"
  la var near_dist_`type'5 "Distance in meters to fifth closest `type'"
  la var meandist_`type' "Mean distance in meters to five closest `type'"
}

```

```
}  
foreach type in AllSuperMarkets ChainFFR ChainSuperMarkets ConvenienceStores  
NonMemberGenMerch AllPark Grocery NonChainFFR NonChainSuperMarkets PAFAC{  
  drop merge_`type'  
}
```