# Create the spatial datasets with 1999 as base year

*2017-03-28*

## Summary

This file recreates the spatial file from the raw csv files that were downloaded from the spatial database. The files are cleaned, merged and brought to a based year of 1999.

## Loading the raw csv files

This section of the code loops through the csv files and merges them to create a separate file for the years 2001 and 2011. There are 20 csv files in 2001 and 40 in 2010. Some of the data (particularly in 2001) have incomplete district coverage, while some others are only present at the "total" geography and not the rural or urban.

Each file is uniquely identified by the variables id and geography.

```
rm(list = ls())
library(tidyverse); library(stringr); library(readxl)

##Create the list of files in the directory
allFiles2001 <- list.files(path = "../../data/Spatial Database/All Files/2001/")
allFiles2011 <- list.files(path = "../../data/Spatial Database/All Files/2011/")

##Load the first file in each directory
dataString2001 <- str_c("../../data/Spatial Database/All Files/2001/", allFiles2001[1])
dataString2011 <- str_c("../../data/Spatial Database/All Files/2011/", allFiles2011[1])

data2001 <- read_csv(dataString2001)
data2011 <- read_csv(dataString2011)

## Note: There are a few files that have incomplete coverage (one which only has a subset of the distric
##2001
for (i in 2:length(allFiles2001)){
        dataString2001 <- str_c("../../data/Spatial Database/All Files/2001/", allFiles2001[i])
        temp <- read_csv(dataString2001) %>%
                select(-spatial_data_yr, -L0_code, -L0_name, -L1_name, -L1_code, -L2_code, -L2_name)

        ##Add to the data from previous iteration
        data2001 <- left_join(data2001, temp, by = c("id", "geography"))
}

##2010
for (i in 2:length(allFiles2011)){
        dataString2011 <- str_c("../../data/Spatial Database/All Files/2011/", allFiles2011[i])
        temp <- read_csv(dataString2011) %>%
                select(-spatial_data_yr, -L0_code, -L0_name, -L1_name, -L1_code, -L2_code, -L2_name)

        ##Add to the data from previous iteration
        data2011 <- left_join(data2011, temp, by = c("id", "geography"))
}
```

## Match column names and combine the years

2011 has 450 variables while 2001 only has 210. For the purpose of our analysis, we only need those variable that are covered in both years. So I match the names and drop those that are not present in both years and then combine the two.

```
##Identify the common names
varNamesCommon <- names(data2011)[names(data2011) %in% names(data2001)]

colNums2011 <- match(varNamesCommon,names(data2011)) ##gives the posiion of the column in the data frame
colNums2001 <- match(varNamesCommon,names(data2001)) ##gives the posiion of the column in the data frame

data2011 <- data2011 %>%
        select(colNums2011) ## selects and orders columns based on the varNamesCommon order

data2001 <- data2001 %>%
        select(colNums2001) ## selects and orders columns based on the varNamesCommon order

##Check if the variable names are identical
identical(names(data2001), names(data2011))
```

```
## [1] TRUE
```

```
##Combine the data
spatialAll <- rbind(data2001, data2011)
```

So now we have the combined spatial dataset that has 647 districts  and  210 variables .

## Bring the data to 1999

The 1999 correspondence file maps district ids to the NSS district list in 1999. I merge this information into the spatial dataset. The variables that are levels are summed  while rates are averaged (weighted with either area or population when appropriate).

### Select Variables of Interest

Before summarising the data based on the final id variable we need to identify the ones that are used in the analysis. I maintain separate files with the names of the outcome and control variables. The variables that we need for the analysis are listed below.

```
outcomeVariables <- read_csv("../../data/1 Cleaned files for analysis/Regression Variables/outcomeVariab
        select(varNames, varDescription, summaryType)
controlVariables <- read_csv("../../data/1 Cleaned files for analysis/Regression Variables/controlVariab
        select(varNames, varDescription, summaryType)
varsOfInterest <- rbind(outcomeVariables, controlVariables) %>%
        filter(!duplicated(varNames)) %>%
        arrange(varNames)
varsOfInterest
```

```
## # A tibble: 21 × 3
##       varNames
##          <chr>
## 1           ap
## 2           at
## 3       bank_t
## 4           cm
## 5         dens
## 6  edu_lit_7_t
## 7         elev
## 8      emp_7_f
## 9      emp_7_t
## 10   emp_rwg_f
## # ... with 11 more rows, and 2 more variables: varDescription <chr>,
## #   summaryType <chr>
```

One of the control variables, share of urban population, needs to constructed, since it does not exist in the dataset. So I create this variable.

```
spatialAll <- spatialAll %>%
        arrange(id, spatial_data_yr, geography) %>% #imp. for later steps that used indexes
        group_by(id, spatial_data_yr) %>% #three obs. in the order rural, total, urban (since geo is so
        mutate(urbanPopShare = pop[3]/pop[2]) %>% # urban/total
        ungroup()
```

The variables that are to be summed are as follows.

```
sumVars <- varsOfInterest %>%
        filter(summaryType == "sum")
sumVars
```

```
## # A tibble: 2 × 3
##   varNames                  varDescription summaryType
##      <chr>                          <chr>       <chr>
## 1      gdp GDP  (current USD, in millions)         sum
## 2      pop          Population (thousands)         sum
```

The variables that are to averaged (simple mean) are,

```
simpleMeanVars <- varsOfInterest %>%
        filter(summaryType == "mean")
simpleMeanVars
```

```
## # A tibble: 12 × 3
##       varNames
##          <chr>
## 1           ap
## 2           at
## 3       bank_t
## 4           cm
## 5  edu_lit_7_t
## 6         elev
## 7      emp_7_f
## 8      emp_7_t
## 9    emp_rwg_f
## 10   emp_rwg_t
## 11   hh_elec_t
## 12          nd
## # ... with 2 more variables: varDescription <chr>, summaryType <chr>
```

The variables that will be averaged using population weights are as follows.

```
popMeanVars <- varsOfInterest %>%
        filter(summaryType == "pop weighted mean")
```

```
popMeanVars
```

```
## # A tibble: 4 × 3
##        varNames                            varDescription
##           <chr>                                     <chr>
## 1        gdp_pc              GDP per capita (current USD)
## 2        ntl_pc           Light intensity per 1000 people
## 3            sc Scheduled Caste (SC) population (percent)
## 4 urbanPopShare                 Share of urban population
## # ... with 1 more variables: summaryType <chr>
```

The variables that will averaged using area weights are as follows.

```
areaMeanVars <- varsOfInterest %>%
        filter(summaryType == "area weighted mean")
areaMeanVars
```

```
## # A tibble: 3 × 3
##   varNames                          varDescription        summaryType
##      <chr>                                   <chr>              <chr>
## 1     dens Population density (people per sq. km.) area weighted mean
## 2     fo_s                Forest (percent of area) area weighted mean
## 3    ntl_a                Light intensity per area area weighted mean
```

### Load and merge the final ids (1999)

Now I add the ids ('finalId') that are based on the 1999 base year.

```
load("../../data/1 Cleaned files for analysis/Correspondence Files/districtCorrespondence99.RDA")

spatialAll <- left_join(spatialAll, districtCorrespondence99, by = c("id" = "spatialId"))
```

### Summarise the variables to the base year of 1999

In this step, I group the data by the final id, year and geography and summarise variables based on their summary type (i.e. sum, simple mean or pop/area weighted mean).

```
spatialAll <- spatialAll %>%
        group_by(finalId, geography, spatial_data_yr) %>%
        mutate_at(areaMeanVars$varNames, .funs = funs(sum(. * area, na.rm = T)/sum(area, na.rm = T))) %
        mutate_at(popMeanVars$varNames, .funs = funs(sum(. * pop, na.rm = T)/sum(pop, na.rm = T))) %>%
        mutate_at(simpleMeanVars$varNames, .funs = funs(mean(., na.rm = T))) %>%
        mutate_at(sumVars$varNames, .funs = funs(sum(., na.rm = T))) %>%
        filter(row_number() == 1) %>%
        ungroup() %>%
        select(finalId, geography, year = spatial_data_yr, one_of(varsOfInterest$varNames))
spatialAll
```

```
## # A tibble: 2,910 × 24
##     finalId geography  year    ap    at bank_t    cm      dens edu_lit_7_t
##       <chr>     <chr> <int> <dbl> <dbl>  <dbl> <dbl>     <dbl>       <dbl>
## 1   3_1_1_0     Rural  2001   NaN   NaN   8.90   NaN  241.0000       42.40
## 2   3_1_1_0     Total  2001 53.30 0.130   9.90 106.0  248.0000       43.20
## 3   3_1_1_0     Urban  2001   NaN   NaN  35.90   NaN 1146.0000       62.80
## 4   3_1_1_0     Rural  2011   NaN   NaN  49.10   NaN  299.0000       62.90
## 5   3_1_1_0     Total  2011 65.00 0.170  50.20  99.7  332.0000       64.50
## 6   3_1_1_0     Urban  2011   NaN   NaN  61.10   NaN 1904.0000       75.60
## 7  3_1_10_0     Rural  2001   NaN   NaN  17.70   NaN  165.9473       40.45
## 8  3_1_10_0     Total  2001 35.95 0.225  28.35 104.0  488.4082       50.90
## 9  3_1_10_0     Urban  2001   NaN   NaN  42.15   NaN 1660.2632       62.30
## 10 3_1_10_0     Rural  2011   NaN   NaN  56.70   NaN  199.8598       55.00
## # ... with 2,900 more rows, and 15 more variables: elev <dbl>,
## #   emp_7_f <dbl>, emp_7_t <dbl>, emp_rwg_f <dbl>, emp_rwg_t <dbl>,
## #   fo_s <dbl>, gdp <dbl>, gdp_pc <dbl>, hh_elec_t <dbl>, nd <dbl>,
## #   ntl_a <dbl>, ntl_pc <dbl>, pop <dbl>, sc <dbl>, urbanPopShare <dbl>
```

Now we have the data summarised at the 1999 id. The next step is to separate out the different geographies and save the data for later use.

```
spatialTotal <- spatialAll %>%
        filter(geography == "Total") %>%
        select(-geography) %>%
        arrange(finalId, year)
save(spatialTotal, file = "../../data/1 Cleaned files for analysis/Spatial Database/spatialTotal.RDA")

spatialUrban <- spatialAll %>%
        filter(geography == "Urban") %>%
        select(-geography) %>%
        arrange(finalId, year)
save(spatialUrban, file = "../../data/1 Cleaned files for analysis/Spatial Database/spatialUrban.RDA")

spatialRural <- spatialAll %>%
        filter(geography == "Rural") %>%
        select(-geography) %>%
        arrange(finalId, year)
save(spatialRural, file = "../../data/1 Cleaned files for analysis/Spatial Database/spatialRural.RDA")
```