Create the urban and rural LF datasets with 1999 as base year

2017-03-28

Summary

This file creates all the datasets for the regression using labor force data and 1999 as the base concordance year (based on the NSS district list from 1999).

The district ids across different years of labour force surveys are consistent. To avoid any potential errors, I match datasets to the 1999 correspondence files using state and district names (unmatched districts are manually matched). We will be using the following datasets. 1. 8 labour force datasets split into urban and rural for the years 1993, 1999, 2004, 2010 2. The NSS correspondences for these years. I use these for names of states and districts 2. The district correspondence files for 1999

The 1993 dataset is a bit more tricky since it involves splitting districts (since we are bringing the districts forwards to 1999), so I tackle this the last. I start from the most recent year and then go backwards.

Data Issues

1. The urban and rural datasets for 2010 contain district ids that are not found in the master list (Final_2010-1989(67th).dta) or the NSS district reference file. These districts start with either 12 or 15 (state ids). Both these could not be placed and had to be filtered out of the data.
2. The urban dataset had two district (798 and 799) that were missing from the master list and the pdf reference file. These were filtered out.
3. The poverty and consumption numbers are often recorded as zero (when they should be missing).

2010 LF data

The lf data is matched with the correpondence file using the names of the states and districts. ### Load the datasets Both the LF datasets have the same number of variables but different number of districts covered (urban = 593, rural = 602)

```
rm(list = ls())
library(tidyverse); library(haven)
rural10 <- read_dta("../../data/labor force surveys/lfs200910_rural.dta")%>%
        mutate(distt_id66 = as.character(distt_id66))
urban10 <- read_dta("../../data/labor force surveys/lfs200910_urban.dta")%>%
        mutate(distt_id66 = as.character(distt_id66))
NSSCodes <- read_dta("../../data/labor force surveys/Final_2010-1989(67th).dta")

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

Match to 1999 correspondence file

In this section I modify the orginial NSS code file by creating the district id variable (adding a preceding zero to single digit district codes) and then merging the spatial id variables from the districtCorrespondence99 file. There are 2 districts Nainital (H) and Dehradun (H) which do not appear in the districtCorrespondence99 file. I **drop** these two. Now we have a file that maps the district ids for 2010 to the final patial ids that are mapped to the NSS 1999 district list.

```
NSSCodes <- NSSCodes %>%
        select(9, 8, 12, 11) %>%
        mutate(NSS2010_dist_code = ifelse(NSS2010_dist_code < 10, paste("0", NSS2010_dist_code, sep = "
        mutate(distt_id66 = paste(NSS2010_state_id, NSS2010_dist_code, sep = "")) %>%
        arrange(NSS2010_state, NSS2010_dist_code) %>%
        select(distt_id66, NSS2010_state_id, NSS2010_state, NSS2010_dist_code, NSS2010_dist) %>%
        left_join(., districtCorrespondence99, by = c("NSS2010_state" = "nss2010State" , "NSS2010_dist"
        select(distt_id66, finalId)
```

Create the urban and rural LF datasets

Both the urban and rural datasets contain unaccounted district ids (starting with either 12 or 15). These district ids do not appear in the master list i.e. NSSCodes dataset or the reference pdf.

There are 24 such observation in the rural dataset

```
rural10 %>% filter(!(distt_id66 %in% NSSCodes$distt_id66))
```

```
## # A tibble: 24 × 35
##    distt_id66 wagework_female_rur hhenterprise_female_rur
##         <chr>               <dbl>                   <dbl>
## 1        1201             449.370                  150.37
## 2        1202             450.200                 2660.21
## 3        1203               0.000                    0.00
## 4        1204            2609.675                    0.00
## 5        1205               0.000                    0.00
## 6        1206             702.630                  571.26
## 7        1207             330.115                  455.53
## 8        1208             219.080                    0.00
## 9        1209             787.050                    0.00
## 10       1210               0.000                    0.00
## # ... with 14 more rows, and 32 more variables: employer_female_rur <dbl>,
## #   selfemployed_female_rur <dbl>, agr_female_rur <dbl>,
## #   mfg_female_rur <dbl>, commerce_female_rur <dbl>,
## #   transportation_female_rur <dbl>, other_services_female_rur <dbl>,
## #   population_female_rur <dbl>, REG_WAGE_female_rur <dbl>,
## #   FORMAL_female_rur <dbl>, EMPLOYED_female_rur <dbl>,
## #   UNEMPLYD_female_rur <dbl>, INACTIVE_female_rur <dbl>,
## #   wagework_male_rur <dbl>, hhenterprise_male_rur <dbl>,
## #   employer_male_rur <dbl>, selfemployed_male_rur <dbl>,
## #   agr_male_rur <dbl>, mfg_male_rur <dbl>, commerce_male_rur <dbl>,
## #   transportation_male_rur <dbl>, other_services_male_rur <dbl>,
## #   population_male_rur <dbl>, REG_WAGE_male_rur <dbl>,
## #   FORMAL_male_rur <dbl>, EMPLOYED_male_rur <dbl>,
## #   UNEMPLYD_male_rur <dbl>, INACTIVE_male_rur <dbl>,
## #   cons_pc_mean_rur <dbl>, poor_rur <dbl>, population_rur <dbl>,
## #   povrate_rur <dbl>
```

And 19 in the urban.

```
urban10 %>% filter(!(distt_id66 %in% NSSCodes$distt_id66))
```

```
## # A tibble: 19 × 35
##    distt_id66 wagework_female_urb hhenterprise_female_urb
##         <chr>               <dbl>                   <dbl>
## 1        1201             311.720                   7.790
## 2        1202             238.500                 492.750
## 3        1203             474.690                   0.000
## 4        1204            1486.020                   0.000
## 5        1205             211.000                   0.000
## 6        1206             923.000                   0.000
## 7        1207             114.000                   0.000
## 8        1208             484.310                   0.000
## 9        1211             157.815                   0.000
## 10       1212             612.000                   0.000
## 11       1213             207.810                   0.000
## 12       1216             744.000                   0.000
## 13       1501             389.815                1391.255
## 14       1502             169.625                4163.005
## 15       1503            7740.405                8220.295
## 16       1504             483.880                2733.785
## 17       1505             448.085                3262.065
## 18       1506            1322.380                1311.280
## 19       1508             298.750                 577.500
## # ... with 32 more variables: employer_female_urb <dbl>,
## #   selfemployed_female_urb <dbl>, agr_female_urb <dbl>,
## #   mfg_female_urb <dbl>, commerce_female_urb <dbl>,
## #   transportation_female_urb <dbl>, other_services_female_urb <dbl>,
## #   population_female_urb <dbl>, REG_WAGE_female_urb <dbl>,
## #   FORMAL_female_urb <dbl>, EMPLOYED_female_urb <dbl>,
## #   UNEMPLYD_female_urb <dbl>, INACTIVE_female_urb <dbl>,
## #   wagework_male_urb <dbl>, hhenterprise_male_urb <dbl>,
## #   employer_male_urb <dbl>, selfemployed_male_urb <dbl>,
## #   agr_male_urb <dbl>, mfg_male_urb <dbl>, commerce_male_urb <dbl>,
## #   transportation_male_urb <dbl>, other_services_male_urb <dbl>,
## #   population_male_urb <dbl>, REG_WAGE_male_urb <dbl>,
## #   FORMAL_male_urb <dbl>, EMPLOYED_male_urb <dbl>,
## #   UNEMPLYD_male_urb <dbl>, INACTIVE_male_urb <dbl>,
## #   cons_pc_mean_urb <dbl>, poor_urb <dbl>, population_urb <dbl>,
## #   povrate_urb <dbl>
```

I remove these district ids from both the datasets (since they cannot be matched to state or district names).

```
urban10 <- filter(urban10, distt_id66 %in% NSSCodes$distt_id66)
rural10 <- filter(rural10, distt_id66 %in% NSSCodes$distt_id66)
```

Now I merge the NSS code file that contains the final spatial id variable to the urban and rural datasets. We need an end result that is mapped to this final spatial id variable. I sum all the levels variables with the same final id variables to create datasets that are mapped to unique final spatial ids.

```
urban10 <- left_join(urban10, NSSCodes, by = "distt_id66") %>%
        filter(!is.na(finalId)) %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 2010) %>%
        select(36:37, 2:35)
```

```
rural10 <- left_join(rural10, NSSCodes, by = "distt_id66") %>%
        filter(!is.na(finalId)) %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 2010) %>%
        select(36:37, 2:35)
```

The datasets for 2010 are good to go.

2004 LF Data

I won't detail individual steps unless there is something to point out. I follow the same steps as for 2010.

```
rural04 <- read_dta("../../data/labor force surveys/lfs200405_rural.dta")%>%
        mutate(distt_id61 = as.character(distt_id61))
urban04 <- read_dta("../../data/labor force surveys/lfs200405_urban.dta")%>%
        mutate(distt_id61 = as.character(distt_id61))
NSSCodes <- read_dta("../../data/labor force surveys/Final_2004-1989(60th).dta")
```

First I match the state and district names to the NSS 2010 There are 37 districts in the LF 2004 code list that were not matched. I filter these out and then match these based on the 2001 NSS state and district names. This leaves 4 unmatched observations that I match manually and add to list.

```
##Create district id and match based on 2010 NSS names
NSSCodes <- NSSCodes %>%
        select(9, 8, 12, 11) %>%
        mutate(NSS2004_dist_code = ifelse(NSS2004_dist_code < 10, paste("0", NSS2004_dist_code, sep = "
        mutate(distt_id61 = paste(NSS2004_state_id, NSS2004_dist_code, sep = "")) %>%
        arrange(NSS2004_state, NSS2004_dist_code) %>%
        select(distt_id61, NSS2004_state_id, NSS2004_state, NSS2004_dist_code, NSS2004_dist) %>%
        left_join(., districtCorrespondence99, by = c("NSS2004_state" = "nss2010State" , "NSS2004_dist"
##Match the unmatched ones using 2001 names
unMatched <- NSSCodes %>%
        filter(is.na(finalId)) %>%
        select(1:5) %>%
         left_join(., districtCorrespondence99, by = c("NSS2004_state" = "nss2001State" , "NSS2004_dist

##store unmatched for manual match
write_csv(filter(unMatched, is.na(finalId)), "../../data/labor force surveys/99 concordance/unmatched20

##Combine everything
unMatched <- unMatched %>%
        filter(!is.na(finalId)) %>%
        select(distt_id61, finalId)

NSSCodes <- NSSCodes %>%
        filter(!is.na(finalId)) %>%
        select(distt_id61, finalId) %>%
        rbind(., unMatched) %>%
        rbind(., read_csv("../../data/labor force surveys/99 concordance/matched2004LF.csv"))
```

There are two datapoints that are missing in the urban dataset that could not be placed. I filter these out

```
urban04 %>% filter(!(distt_id61 %in% NSSCodes$distt_id61))
```

```
## # A tibble: 2 × 35
##   distt_id61 wagework_female_urb hhenterprise_female_urb
##        <chr>               <dbl>                   <dbl>
## 1        798             45894.0                    0.00
## 2        799            267709.3                21939.67
## # ... with 32 more variables: employer_female_urb <dbl>,
## #   selfemployed_female_urb <dbl>, agr_female_urb <dbl>,
## #   mfg_female_urb <dbl>, commerce_female_urb <dbl>,
## #   transportation_female_urb <dbl>, other_services_female_urb <dbl>,
## #   population_female_urb <dbl>, REG_WAGE_female_urb <dbl>,
## #   FORMAL_female_urb <dbl>, EMPLOYED_female_urb <dbl>,
## #   UNEMPLOYED_female_urb <dbl>, INACTIVE_female_urb <dbl>,
## #   wagework_male_urb <dbl>, hhenterprise_male_urb <dbl>,
## #   employer_male_urb <dbl>, selfemployed_male_urb <dbl>,
## #   agr_male_urb <dbl>, mfg_male_urb <dbl>, commerce_male_urb <dbl>,
## #   transportation_male_urb <dbl>, other_services_male_urb <dbl>,
## #   population_male_urb <dbl>, REG_WAGE_male_urb <dbl>,
## #   FORMAL_male_urb <dbl>, EMPLOYED_male_urb <dbl>,
## #   UNEMPLOYED_male_urb <dbl>, INACTIVE_male_urb <dbl>,
## #   cons_pc_mean_urb <dbl>, poor_urb <dbl>, population_urb <dbl>,
## #   povrate_urb <dbl>
```

```
urban04 <- urban04 %>%
        filter(distt_id61 %in% NSSCodes$distt_id61)
```

```
urban04 <- left_join(urban04, NSSCodes, by = "distt_id61") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 2004) %>%
        select(36:37, 2:35)

rural04 <- left_join(rural04, NSSCodes, by = "distt_id61") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 2004) %>%
        select(36:37, 2:35)
```

1999 LF data

This dataset is the most straightforward one to merge since the 1999 correspondence file was created using the list of districts from LF 1999 data.

```
rural99 <- read_dta("../../data/labor force surveys/lfs199900_rural.dta") %>%
        mutate(distt_id55 = as.character(distt_id55))
urban99 <- read_dta("../../data/labor force surveys/lfs199900_urban.dta")%>%
        mutate(distt_id55 = as.character(distt_id55))
```

```
distrinCorrespondenceIds <- districtCorrespondence99 %>%
        select(distt_id55, finalId)
urban99 <- left_join(urban99, distrinCorrespondenceIds, by = "distt_id55") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 1999) %>%
        select(36:37, 2:35)

rural99 <- left_join(rural99, distrinCorrespondenceIds, by = "distt_id55") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 1999) %>%
        select(36:37, 2:35)
```

1993 LF data

1993 is the trickiest of the survey rounds to merge, because we need to manage district splits, and the presence of ‘mega districts’ in the 1999 correspondence list. The new concordance file maps district ids in 93 to those in 99. I use these to split the 1993 dataset to the 1999 level. Folowwing which we merge the spatial ids and merge (to accomodate mega districts).

```
rural93 <- read_dta("../../data/labor force surveys/lfs9394_rural.dta") %>%
        mutate(distt_id50 = as.character(distt_id50))
urban93 <- read_dta("../../data/labor force surveys/lfs9394_urban.dta") %>%
        mutate(distt_id50 = as.character(distt_id50))

newConcordance <- read_dta("../../data/labor force surveys/distcode_asi88 asi99 nss43 nss50 nss55 code9
        mutate(distt_id55 = as.character(distt_id55), distt_id50 = as.character(distt_id50))
```

Matching with the new concordance file

First I check if all the ids in the data are present in the new concordance file

```
ids93 <- unique(c(unique(rural93$distt_id50), unique(urban93$distt_id50)))
ids93[!(ids93 %in% newConcordance$distt_id50)]
```

```
## character(0)
```

All the 1993 district ids are present in the concordance file. Now I select the unique 1999 ids (the concordance file is based on a later year and therefore the ids are non unique).

```
newConcordance <- newConcordance %>%
        select(distt_id55, distt_id50) %>%
        group_by(distt_id55) %>%
        filter(row_number() == 1) %>%
        group_by()
```

Next step is to merge this data into the urban and rural files and split the data for the ids that are repeated (to account for district split).

```
urban93 <- left_join(urban93, newConcordance, by = "distt_id50") %>%
        group_by(distt_id55) %>%
        mutate_at(2:34, .funs = funs(./n())) %>%
        select(distt_id55, 2:35)
rural93 <- left_join(rural93, newConcordance, by = "distt_id50") %>%
        group_by(distt_id55) %>%
        mutate_at(2:34, .funs = funs(./n())) %>%
        select(distt_id55, 2:35)
```

In the next step I use the district Correspondence file to merge in the final ids. The variable values are then summed to account for mega districts (except for rates)

```
urban93 <- left_join(urban93, distrinCorrespondenceIds, by = "distt_id55") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 1993) %>%
        select(36:37, 2:35)

rural93 <- left_join(rural93, distrinCorrespondenceIds, by = "distt_id55") %>%
        group_by(finalId) %>%
        mutate_at(2:34, sum) %>%
        mutate_at(35, mean) %>% ## this is a rate variable
        filter(row_number() == 1) %>% ## keep only the first row
        ungroup() %>%
        mutate(year = 1993) %>%
        select(36:37, 2:35)
rm("distrinCorrespondenceIds", "unMatched", "newConcordance", "NSSCodes", "ids93")
```

Combining all the datasets

In this section I combine all the datasets ###Making sure that all the datasets have the same names There were a few variable names that were mismatched, I correct these manually.

```
names(urban10)[names(urban10) != names(urban04)] <- c("UNEMPLOYED_female_urb", "UNEMPLOYED_male_urb") #
if(identical(names(urban93), names(urban99)) & identical(names(urban99), names(urban04)) & identical(na
        print("All urban variable names are identical")
}
```

```
## [1] "All urban variable names are identical"
```

Now I merge the urban datasets

```
urbanAll <- rbind(urban93, urban99, urban04, urban10) %>%
        arrange(finalId, year)
save(urbanAll, file = "../../data/1 Cleaned files for analysis/LF/urbanAll.RDA")
```

```
names(rural10)[names(rural10) != names(rural04)]<- c("UNEMPLOYED_female_rur", "UNEMPLOYED_male_rur") ##

if(identical(names(rural93), names(rural99)) & identical(names(rural99), names(rural04)) & identical(na
        print("All variable rural names are identical")
}
```

```
## [1] "All variable rural names are identical"
```

```
ruralAll <- rbind(rural93, rural99, rural04, rural10) %>%
        arrange(finalId, year)
save(ruralAll, file = "../../data/1 Cleaned files for analysis/LF/ruralAll.RDA")
```

The next step is to combine both the datasets (with separate rural + urban variables) and then create additional variables at the 'total' level.

Getting the data ready for regressions

The final dataset would contain both urban and rural variables along with contructed total (rural + urban) variables.

As the first step I merge the rural and urban datasets

```
lfAll <- left_join(ruralAll, urbanAll, by = c("finalId", "year"))
rm(list = setdiff(ls(), "lfAll"))
```

Next we create the following outcome variables

1. For male and female combined, by rural, urban and rural + urban: poverty rate; mean household per capita consumption
2. By gender (male, female, both), and by rural, urban and rural + urban: % employed, % inactive (denominator is population)
3. By gender (male, female, both), cut by rural, urban and rural + urban: % employment in wage jobs, regular wage jobs, formal, self-employed, hhenterprise (denominator is # employed)

4. For male and female combined, cut by rural, urban and rural + urban: % employment in agriculture, manufacturing, transport, commerce (denominator is # employed)

Step 1 Create the poverty and mean consumption variables

We need to create two variables here (since the other 4 exist in the data)

```
##Step 1 variables
lfAll <- ungroup(lfAll) %>%
        group_by(finalId, year) %>%
        mutate(povrate_tot = sum(c(povrate_urb * population_urb, povrate_rur * population_rur), na.rm =
        mutate(cons_pc_mean_tot = sum(c(cons_pc_mean_urb * population_urb, cons_pc_mean_rur * population
```

Step 2: Percent Employed and inactive

We need to create 18 variables here. Gender (m/f/both) x Geography(rur/urb/tot) x Variable(emp/inactive)

```
##Step 2 variables: There are 18 variables that we need to create here
lfAll <- ungroup(lfAll) %>%
        group_by(finalId, year) %>%
        mutate(pct_emp_f_rur = EMPLOYED_female_rur/population_female_rur, pct_emp_m_rur = EMPLOYED_male
        mutate(pct_inactive_f_rur = INACTIVE_female_rur/population_female_rur, pct_inactive_m_rur = INAC
```

Step 3: % employment in wage jobs, regular wage jobs, formal, self-employed, hhenterprise

Here we need to create 45 variables. Gender (m/f/both) x Geography(rur/urb/tot) x 5 Variables % employment in wage jobs, regular wage jobs, formal, self-employed, hhenterprise (denominator is # employed)

```
lfAll <- ungroup(lfAll) %>%
        group_by(finalId, year) %>%
        mutate(pct_wage_f_rur = wagework_female_rur/EMPLOYED_female_rur, pct_wage_m_rur = wagework_male_
        mutate(pct_regWage_f_rur = REG_WAGE_female_rur/EMPLOYED_female_rur, pct_regWage_m_rur = REG_WAG
        mutate(pct_formal_f_rur = FORMAL_female_rur/EMPLOYED_female_rur, pct_formal_m_rur = FORMAL_male_
        mutate(pct_selfemp_f_rur = selfemployed_female_rur/EMPLOYED_female_rur, pct_selfemp_m_rur = sel
        mutate(pct_hhemp_f_rur = hhenterprise_female_rur/EMPLOYED_female_rur, pct_hhemp_m_rur = hhenterp
```

Step 4: Emplyment by sector

Here we need to create 36 variables. Gender (m/f/both) x Geography(rur/urb/tot) x 4 Variables Here we create 9 x 4 variables. For male and female combined, cut by rural, urban and rural + urban: % employment in agriculture, manufacturing, transport, commerce (denominator is # employed)

```
lfAll <- ungroup(lfAll) %>%
        group_by(finalId, year) %>%
        mutate(pct_agrEmp_f_rur = agr_female_rur/EMPLOYED_female_rur, pct_agrEmp_m_rur = agr_male_rur/E
        mutate(pct_mfgEmp_f_rur = mfg_female_rur/EMPLOYED_female_rur, pct_mfgEmp_m_rur = mfg_male_rur/E
        mutate(pct_trspEmp_f_rur = transportation_female_rur/EMPLOYED_female_rur, pct_trspEmp_m_rur = t
        mutate(pct_comEmp_f_rur = commerce_female_rur/EMPLOYED_female_rur, pct_comEmp_m_rur = commerce_
```

Now that all the variables are created, we can select those we want want for the analysis and discard the rest.

```
lfOutcomes <- read_csv("../../data/1 Cleaned files for analysis/Regression Variables/lfOutomes.csv")
lfAll <- ungroup(lfAll) %>%
        select(finalId, year, one_of(lfOutcomes$varNames))
save(lfAll, file = "../../data/1 Cleaned files for analysis/LF/lfAll.RDA")
```

The Labor Force surveys are now ready for use.