Who Lives in the New Orleans Area?

Daniel Moscoe

3/12/2021

Procedure Overview In this section of the project, we examine demographic data from the New Orleans Metropolitan Area for the period 2000-2019. The data are contained in an Excel spreadsheet. Some of the features of the data in the spreadsheet enhance human readability. For example, geographic regions are listed as column headers, even though each of these regions represents a level of a variable. Variables, like different age groups, are listed as row headers. While this arrangement makes the table easier to read at a glance, it means the table does not have a "tidy" structure. In this section, we import the data from the Excel sheet, transform it into a tidy form, and conduct some exploratory analysis.

Each of the tables from the Excel sheet required similar transformations: * Transpose the table so that observations reside in rows and variables in columns. * Repair row names. * Repair column names. * Drop empty rows and columns. * Convert data to a numeric form.

While the transformation of each table conformed to this rough outline, each table also required some special manipulation.

The product of the transformation of these individual tables is one large table, `combined`, that contains all the information from all the imported tables.

Procedure Detail Import the raw data from the Excel sheet. Each table is named for the position of its upper-left cell.

```
S1A14 <- read_excel(xlsx_src, range = "A14:C18")
```

```
## New names:
## * '' -> ...1
```

```
S1A22 <- read_excel(xlsx_src, range = "A22:U48")
```

```
## New names:
## * '' -> ...1
## * '' -> ...3
## * '' -> ...5
## * '' -> ...7
## * '' -> ...9
## * ...
```

```
S1A53 <- read_excel(xlsx_src, range = "A53:B68")
```

```
## New names:
## * '' -> ...2
```

```r
S1A72 <- read_excel(xlsx_src, range = "A72:J88")
```

```
## New names:
## * `` -> ...1
```

```r
S1A91 <- read_excel(xlsx_src, range = "A91:I104")
```

```
## New names:
## * `` -> ...1
## * `` -> ...3
## * `` -> ...5
## * `` -> ...7
## * `` -> ...9
```

```r
S1A108 <- read_excel(xlsx_src, range = "A108:I110")
```

```
## New names:
## * `` -> ...1
## * `` -> ...3
## * `` -> ...5
## * `` -> ...7
## * `` -> ...9
```

We begin by transforming the most complex table, `S1A22`, in order to demonstrate completely all the elements of the general transformation procedure described above.

```r
###S1A22###
#TRANSPOSE
S1A22 <- S1A22 %>%
  t() %>%
  data.frame()

#FIX ROW NAMES
S1A22 <- rownames_to_column(S1A22)
S1A22$rowname <- gsub('.[0-9]+', NA, S1A22$rowname)
S1A22 <- fill(S1A22, rowname)

#FIX COLNAMES
S1A22 <- rename(S1A22, "parish" = rowname,
                "year" = X1,
                "f_wht_not_hisp" = X3, #f_ indicates fraction, not count
                "f_blk_not_hisp" = X4,
                "f_hisp_any" = X5,
                "f_asn_not_hisp" = X6,
                "age_btw_0_4" = X9,
                "age_btw_5_9" = X10,
                "age_btw_10_14" = X11,
                "age_btw_15_19" = X12,
                "age_btw_20_24" = X13,
                "age_btw_25_29" = X14,
                "age_btw_30_34" = X15,
```

```
                "age_btw_35_39" = X16,
                "age_btw_40_44" = X17,
                "age_btw_45_49" = X18,
                "age_btw_50_54" = X19,
                "age_btw_55_59" = X20,
                "age_btw_60_64" = X21,
                "age_btw_65_69" = X22,
                "age_btw_70_74" = X23,
                "age_btw_75_79" = X24,
                "age_btw_80_84" = X25,
                "age_geq_85" = X26)


#DROP EMPTY ROWS AND COLUMNS
S1A22 <- S1A22[-c(1),-c(3,8,9)]
S1A22 <- remove_rownames(S1A22)

#CONVERT TO NUMERIC
S1A22[2:length(S1A22[1,])] <- sapply(S1A22[2:length(S1A22[1,])], as.numeric)
```

Choosing variable names carefully will be important here, since using these names consistently across tables will allow us to more easily construct `combined`.

```
###S1A14###
#TRANSPOSE
S1A14 <- S1A14 %>%
  t() %>%
  data.frame()

#FIX ROW NAMES
S1A14 <- rownames_to_column(S1A14)
S1A14 <- cbind(parish = "Orleans", S1A14)

#FIX COLNAMES
S1A14 <- rename(S1A14, "year" = rowname,
                "wht_not_hisp" = X2,
                "blk_not_hisp" = X1,
                "hisp_any" = X3,
                "asn_not_hisp" = X4)

#DROP EMPTY ROWS AND COLUMNS
S1A14 <- S1A14[-c(1),-c(3,5)]
S1A14 <- remove_rownames(S1A14)

#CONVERT TO NUMERIC
S1A14[2:length(S1A14[1,])] <- sapply(S1A14[2:length(S1A14[1,])], as.numeric)


###S1A53###
#FIX ROW NAMES

S1A53 <- cbind(parish = "Orleans", S1A53)

#FIX COLNAMES
S1A53 <- rename(S1A53,
```

```r
                    "year" = Year,
                    "blk_not_hisp" = ...2)

#CONVERT TO NUMERIC
S1A53[2:length(S1A53[1,])] <- sapply(S1A53[2:length(S1A53[1,])], as.numeric)


###S1A72###
S1A72 <- pivot_longer(S1A72, `Orleans`:`New Orleans Metro`)

#FIX COLNAMES
S1A72 <- rename(S1A72, "parish" = name,
                "year" = ...1,
                "hisp_any" = value)

#IMPROVE CONSISTENCY ACROSS TABLES
S1A72$parish[S1A72$parish == "St. John the Baptist"] <-
  "St. John"

#DROP EMPTY ROWS AND COLUMNS AND REARRANGE
S1A72 <- S1A72[-c(1:9),]
S1A72 <- S1A72[,c(2,1,3)]



#CONVERT TO NUMERIC
S1A72[2:length(S1A72[1,])] <- sapply(S1A72[2:length(S1A72[1,])], as.numeric)


###S1A91###
#TRANSPOSE
S1A91 <- S1A91 %>%
  t() %>%
  data.frame()

#FIX ROW NAMES
S1A91 <- rownames_to_column(S1A91)
S1A91$rowname <- gsub('.[0-9]+', NA, S1A91$rowname)
S1A91 <- fill(S1A91, rowname)

#FIX COLNAMES
S1A91 <- rename(S1A91, "parish" = rowname,
                "year" = X1,
                "f_hisp_cub" = X3,
                "f_hisp_dom" = X4,
                "f_hisp_mex" = X5,
                "f_hisp_pr" = X6,
                "f_hisp_hon" = X7,
                "f_hisp_gua" = X8,
                "f_hisp_nic" = X9,
                "f_hisp_sal" = X10,
                "f_hisp_ca" = X11,
                "f_hisp_sa" = X12,
                "f_hisp_oth" = X13)

#DROP EMPTY ROWS AND COLUMNS
```

```
S1A91 <- S1A91[-c(1,3,5,7,9),-3]
S1A91 <- remove_rownames(S1A91)

#CONVERT TO NUMERIC
S1A91[2:length(S1A91[1,])] <- sapply(S1A91[2:length(S1A91[1,])], as.numeric)


###S1A108###

#TRANSPOSE
S1A108 <- S1A108 %>%
  t() %>%
  data.frame()

#FIX ROW NAMES
S1A108 <- rownames_to_column(S1A108)
S1A108$rowname <- gsub('.[0-9]+', NA, S1A108$rowname)
S1A108 <- fill(S1A108, rowname)

#FIX COLNAMES
S1A108 <- rename(S1A108, "parish" = rowname,
                 "year" = X1,
                 "age_btw_0_18" = X2)

#DROP EMPTY ROWS AND COLUMNS
S1A108 <- S1A108[-c(1),]
S1A108 <- remove_rownames(S1A108)

#CONVERT TO NUMERIC
S1A108[2:length(S1A108[1,])] <- sapply(S1A108[2:length(S1A108[1,])], as.numeric)
```

Now that each table conforms to a tidy format, each variable name is used consistently whenever it appears across tables, and each observation is led by a parish and a year in the first two columns, we are ready to join the tables. Since we don't want to lose any information, we use `full_join` on `parish` and `year` on one table after the other. The result is a large table containing all the information, called `combined`.

```
combined <-
  full_join(S1A14, S1A22, by = c("parish", "year")) %>%
  full_join(S1A53, by = c("parish", "year")) %>%
  full_join(S1A72, by = c("parish", "year")) %>%
  full_join(S1A91, by = c("parish", "year")) %>%
  full_join(S1A108, by = c("parish", "year"))
```

A first glance at `combined` shows a significant proportion of missing values. It's interesting to compare `combined` to the original Excel tables, which contained no empty cells. Combining the data from each table together helps highlight gaps in the dataset that were hidden by the "full" appearance of the Excel tables. For example, the dataset is missing year-by-year counts of the white, Asian, and African American populations in most or all parishes.

With a tidy table containing all the data, we can now proceed to some exploratory analysis.

Analysis   *How did the number of African American and Hispanic residents in select parishes change during the period 2000-2019?*

This question allows us to explore some of the more complete regions of this dataset. While we might have asked this same question about the other demographic groups described in the dataset, complete year-by-year data is only available for Hispanic residents (in all parishes) and African American residents (in Orleans parish).
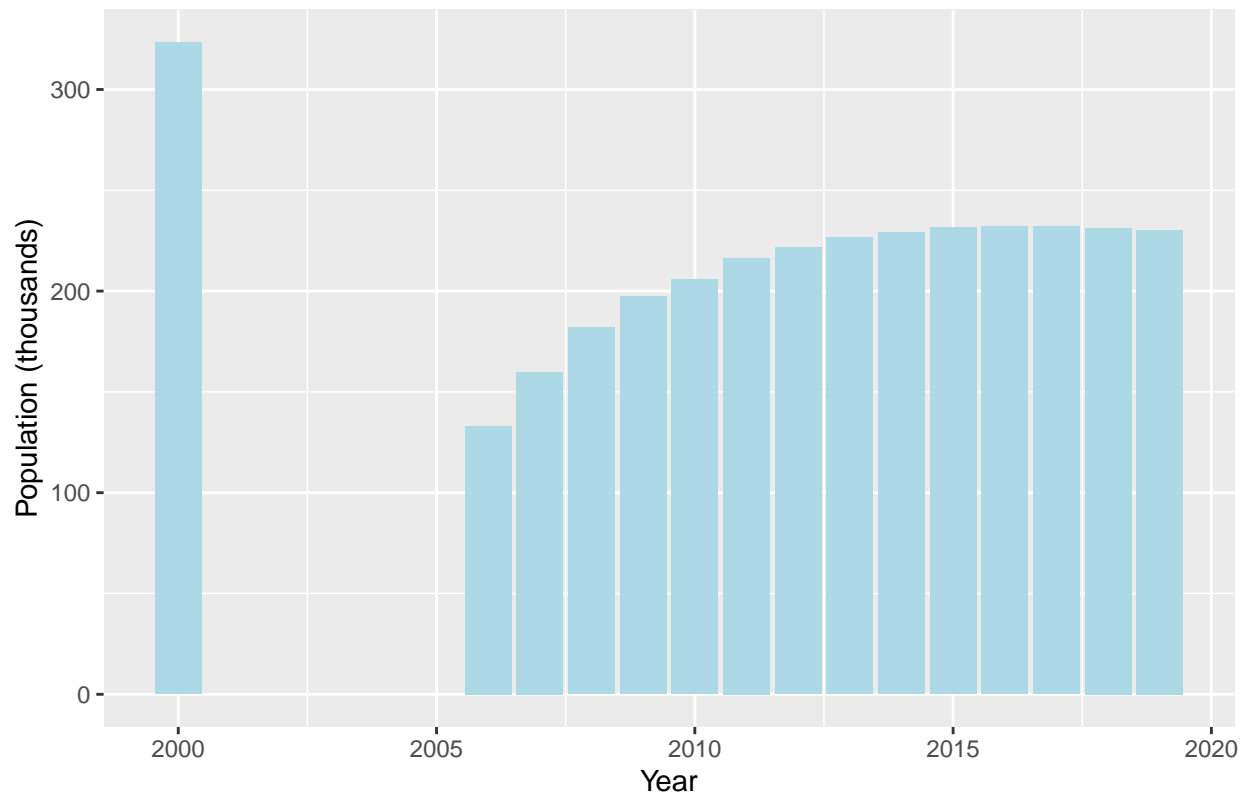
```
combined %>%
  filter(parish %in% c("Orleans", "Jefferson", "New Orleans Metro")) %>%
  ggplot() +
  geom_col(mapping = aes(x = year, y = hisp_any/1000), fill = "light blue") +
  facet_wrap(~parish) +
  labs(title = "Hispanic Population in Selected Regions, 2000-2019", x = "Year", y = "Population (thousa
```


Hispanic Population in Selected Regions, 2000–2019
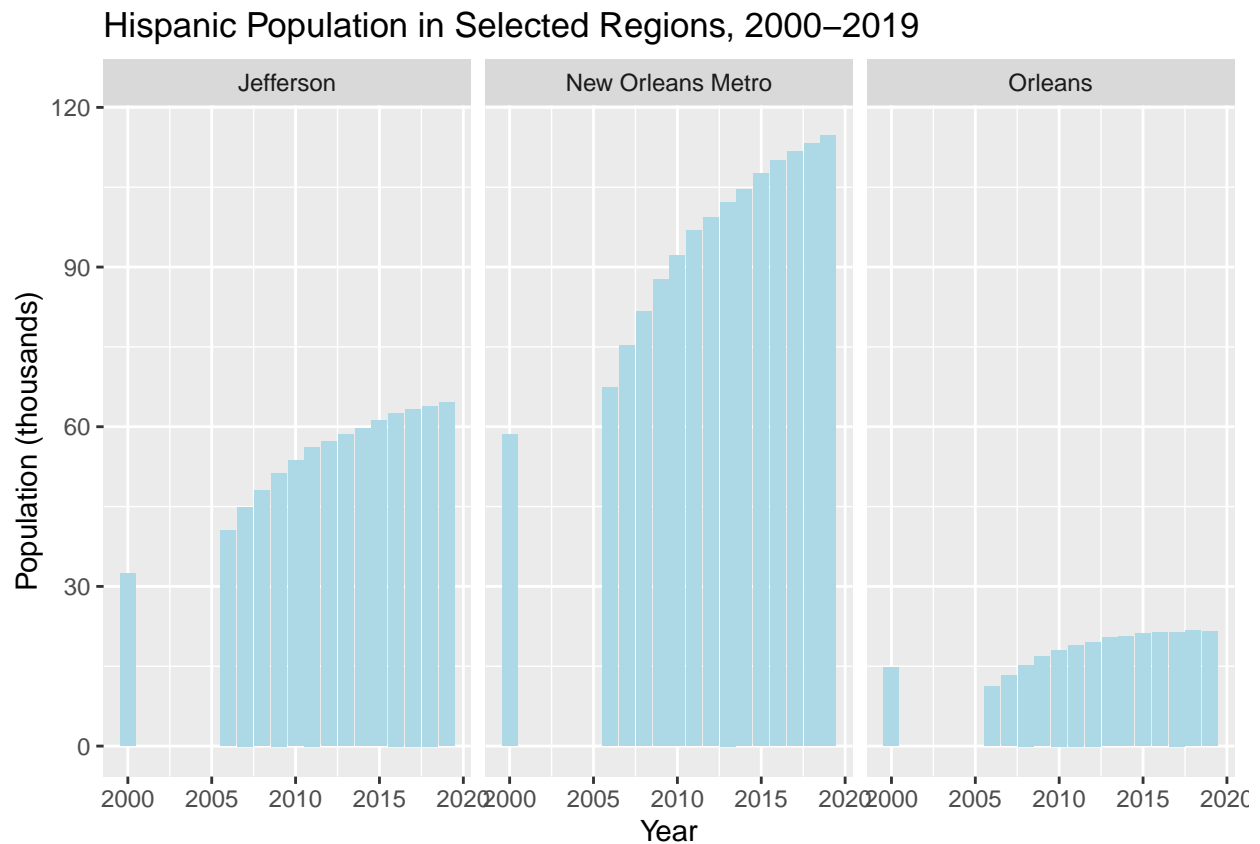
The plot shows that in the three parishes with the greatest numbers of Hispanic residents, the number of Hispanic residents has increased in the period from 2000 to 2019. In the New Orleans Metro area broadly, the number of Hispanic residents has nearly doubled. In Jefferson and Orleans, increases have been more modest. Measurements are missing for the period from 2001-2005.

```
combined %>%
  filter(parish == "Orleans") %>%
  ggplot() +
  geom_col(mapping = aes(x = year, y = blk_not_hisp/1000), fill = "light blue") +
  labs(title = "African American Population in Orleans Parish, 2000-2019", x = "Year", y = "Population
```

African American Population in Orleans Parish, 2000–2019

By contrast, the number of African Americans in Orleans Parish has declined significantly during the period from 2000 to 2019. The decline is sharpest during the period for which data is missing, which includes Hurricane Katrina. The graph above is consistent with the claim that Hurricane Katrina caused a very large fraction–well over half–of African American residents to leave Orleans Parish. While some have returned, the total African American population remains about 30% smaller than it was in 2000.