Instructor: Nara Yoon, Ph.D. (yoonnx@jmu.edu)

Spring 2024 | Thursdays 4-7pm, Hart 3008

Data Analytics for Managerial Leadership

WK 11: Data wrangling II

PLAN FOR TODAY

Overview of core concepts

- Census ACS data
- Data manipulation (with Census ACS data)

CASE IN THE NEWS

Press, G. (Mar 23, 2016). Cleaning big data: Most time-consuming, least enjoyable data science task, survey says. Forbes.

READINGS

Matloff, N. (2011).

- Ch 11. String manipulation (11.1)

Long, J.D., & Teeter, P. (2019).

- Ch 6. Data transformations

U.S. Bureau Census data

What is Census (ACS) data?

THE U.S. CENSUS DATA

The U.S. Census Bureau provides major source of data on…

1. Socioeconomic & demographic characteristics

2. Geographic characteristics

THE U.S. CENSUS DATA

We use ACS (American Community Survey)

- The largest household survey
- Information available at various geographic level (state, county, tract…)

THE MOST POPULAR ACS DATA TABLES

DP02 →

Selected Social Characteristics

Disability Status, Educational Attainment, Language Spoken at Home, Veteran...

Selected Economic Characteristics

Commuting (Journey to Work), Health Insurance Coverage, Income and...

← DP03

DP04 →

Selected Housing Characteristics

Computer and Internet Use, House Heating Fuel, Owner/Renter (Tenure),...

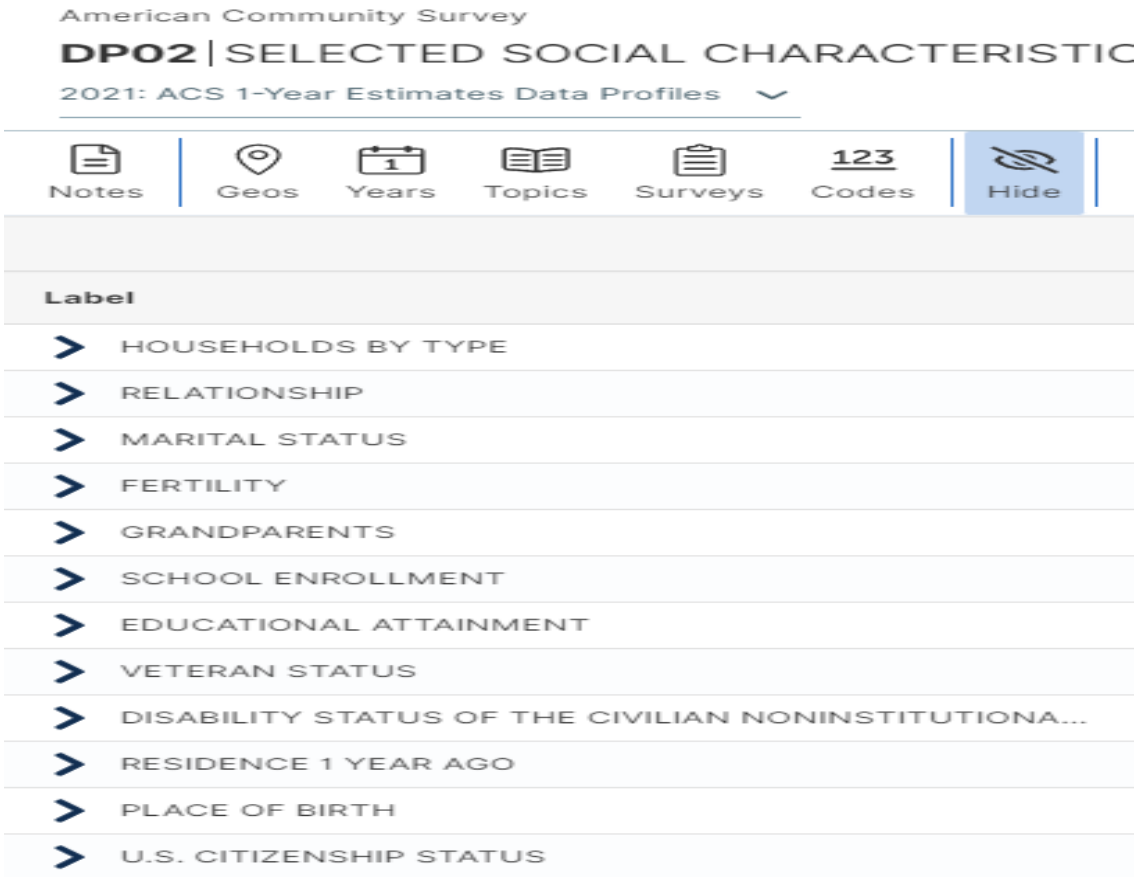
Demographic and Housing Estimates

Age and Sex, Group Quarters Population, Hispanic or Latino Origin, Race, and more

← DP05

Source: U.S. Census Bureau

SOCIAL CHARACTERISTICS
DP02



Source: Census

ECONOMIC CHARACTERISTICS
DP03

American Community Survey

DP03 | SELECTED ECONOMIC CHARACTERISTICS

2021: ACS 1-Year Estimates Data Profiles

Notes | Geos | Years | Topics | Surveys | Codes | Hide | Transpose

Label	United States
EMPLOYMENT STATUS	
COMMUTING TO WORK	
OCCUPATION	
INDUSTRY	
CLASS OF WORKER	
INCOME AND BENEFITS (IN 2021 INFLATION-ADJU...	
HEALTH INSURANCE COVERAGE	
PERCENTAGE OF FAMILIES AND PEOPLE WHOSE IN...	

Source: Census

HOUSING CHARACTERISTICS
DP04

American Community Survey

DP04 | SELECTED HOUSING CHARACTERISTICS

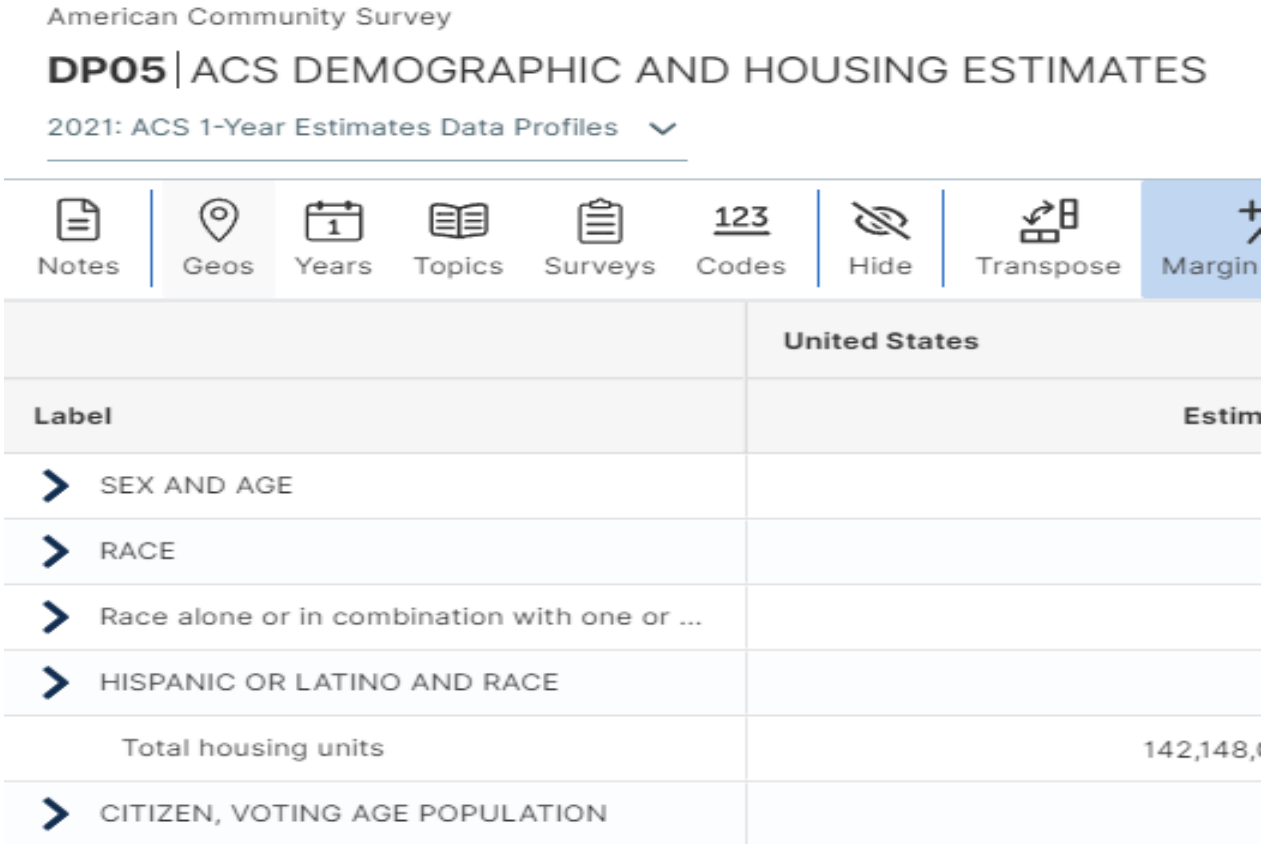
2021: ACS 1-Year Estimates Data Profiles

Notes | Geos | Years | Topics | Surveys | 123 Codes | Hide | Transp

Label	United States
HOUSING OCCUPANCY	
UNITS IN STRUCTURE	
YEAR STRUCTURE BUILT	
ROOMS	
BEDROOMS	
HOUSING TENURE	
YEAR HOUSEHOLDER MOVED IN...	
VEHICLES AVAILABLE	
HOUSE HEATING FUEL	
SELECTED CHARACTERISTICS	
OCCUPANTS PER ROOM	
VALUE	
MORTGAGE STATUS	

Source: Census

DEMOGRAPHIC ESTIMATES
DP05



American Community Survey

DP05 | ACS DEMOGRAPHIC AND HOUSING ESTIMATES

2021: ACS 1-Year Estimates Data Profiles

Notes | Geos | Years | Topics | Surveys | Codes | Hide | Transpose | Margin

	United States
Label	**Estim**
SEX AND AGE	
RACE	
Race alone or in combination with one or ...	
HISPANIC OR LATINO AND RACE	
Total housing units	142,148,(
CITIZEN, VOTING AGE POPULATION	

Source: Census

How to download Census data?

OBTAINING US CENSUS DATA

3 ways to do it

- **Download from Census website (easiest; we use this today!)**
- Use API with *tigris* and *censusapi* (better)
- Use API with *tidycensus* (best)

FOR YOUR REFERENCE:
U.S. BUREAU CENSUS DATA

Download file

Census API

Census data overview

Data manipulation (ACS dataset)

INSTALL PACKAGES

Like last week, we use functions from **dplyr** package

But we load all of the **tidyverse** packages just in case

tidyverse

First, install *tidyverse* and load package

```
install.packages("tidyverse")
library(tidyverse)
```

It includes a set of packages that make it easier for us to import, manipulate, and visualize data

- ggplot2 (to draw graphs)
- tibble (create data tables)
- tidyr (wrangle data)
- readr (load data)
- **dplyr (manipulate data)**

DATA MANIPULATION: *dplyr*


```
install.packages("dplyr")

library(dplyr)
```

- %>%: pipe operator defines sequence of operations
- filter() or select(): to subset the data
- mutate(): to create new variables
- arrange(): to reorder the rows (e.g., ascending, descending)
- group_by(): to split the data
- summarize(): to summarize information

LOAD DATA: CENSUS ACS 2010-2015, NEW YORK

```
dp02 <- read_csv("ACS_DP02_2010_2015_merged.csv")
```

ACS DP02 – social

```
dp03 <- read_csv("ACS_DP03_2010_2015_merged.csv")
```

ACS DP03 – economic

```
dp05 <- read_csv("ACS_DP05_2010_2015_merged.csv")
```

ACS DP05 – demographic

CHECKING DATA:

glimpse()

str()

head()

HOW DOES EACH DATA LOOK LIKE?
glimpse()

```
glimpse(dp02)
```

```
## Rows: 372
## Columns: 4
## $ `County code`
## $ `County name`
## $ Year
## $ `Percent bachelor's degree
```

HOW DOES EACH DATA LOOK LIKE? glimpse()

```
glimpse(dp03)
```

```
## Rows: 372
## Columns: 6
## $ `County code`
## $ `County name`
## $ Year
## $ `Percent unemployed`
## $ `Per capita income (2010 inflati
## $ `Percent of people whose income
```

HOW DOES EACH DATA LOOK LIKE?
glimpse()

```
glimpse(dp05)
```

```
## Rows: 372
## Columns: 13
## $ `County code`
## $ `County name`
## $ Year
## $ Population
## $ `Median age`
## $ `Percent 65 yrs and over`
## $ `Percent female`
## $ `Percent white`
## $ `Percent black or african american`
## $ `Percent indian and alaska native`
## $ `Percent asian`
## $ `Percent native hawaiian and other
## $ `Percent some other race`
```

MERGING DATA:
join()

COMBINE THREE DATASET

```
dp_merge <- left_join(dp02,dp03)
```

```
## Joining, by = c("County code", "County name", "Year")
```

```
dp_merge <- left_join(dp_merge, dp05)
```

```
## Joining, by = c("County code", "County name", "Year")
```

HOW DOES MERGED DATA LOOK LIKE?

```
head(dp_merge)
```

```
## # A tibble: 6 × 17
##   County…¹ Count…² Year Perce…³ Perce…⁴ Per c…⁵ Perce…⁶ Popul…⁷ Media…⁸ Perce…⁹
##      <dbl> <chr>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1    36001 Albany   2010    37.6     6     30863    12.6  304032    38.1    13.8
## 2    36003 Allega…  2010    18.6     8.4   20058    16.5   49105    37.1    14.8
## 3    36005 Bronx    2010    17.6    12.1   17575    28.4 1365725    32.6    10.4
## 4    36007 Broome   2010    25.1     6.8   24314    15.5  200804    39.8    16.2
## 5    36009 Cattar…  2010    18.1     8.4   20824    16     80776    40.3    15.1
## 6    36011 Cayuga   2010    18.4     6.5   22959    12.1   80431    40.7    15
## # … with 7 more variables: `Percent female` <dbl>, `Percent white` <dbl>,
## #   `Percent black or african american` <dbl>,
## #   `Percent indian and alaska native` <dbl>, `Percent asian` <dbl>,
## #   `Percent native hawaiian and other pacific island` <dbl>,
## #   `Percent some other race` <dbl>, and abbreviated variable names
## #   ¹`County code`, ²`County name`,
## #   ³`Percent bachelor's degree or higher (population 25 yrs and over)`, …
```

RENAMING VARIABLES:
rename()

RENAME VARIABLES

```
colnames(dp_merge)   # view all column names
```

← Variable names in the current dataset are too long to use – let's clean them

```
##  [1] "County code"
##  [2] "County name"
##  [3] "Year"
##  [4] "Percent bachelor's degree or higher (population 25 yrs and over)"
##  [5] "Percent unemployed"
##  [6] "Per capita income (2010 inflation-adjusted dollars)"
##  [7] "Percent of people whose income in the past 12 months is below poverty level"
##  [8] "Population"
##  [9] "Median age"
## [10] "Percent 65 yrs and over"
## [11] "Percent female"
## [12] "Percent white"
## [13] "Percent black or african american"
## [14] "Percent indian and alaska native"
## [15] "Percent asian"
## [16] "Percent native hawaiian and other pacific island"
## [17] "Percent some other race"
```

ASSIGN NEW NAMES!

Rename (“NEW”=”OLD”)

```
dp_merge <- dp_merge %>%
            rename ("cty_code" ="County code",
                    "cty_name" = "County name",
                    "yr"= "Year",
                    "pct_bachelor" = "Percent bachelor's degree or higher (population 25 yrs and over)",
                    "pct_unemp" = "Percent unemployed",
                    "income" = "Per capita income (2010 inflation-adjusted dollars)",
                    "pct_poverty" = "Percent of people whose income in the past 12 months is below poverty level",
                    "pop" = "Population",
                    "age"= "Median age",
                    "pct_65" ="Percent 65 yrs and over",
                    "pct_female" = "Percent female",
                    "pct_white" = "Percent white")
```

MODIFY VARIABLE BASED ON EXISTING VARIABLES:
mutate()

In the dataset, we have “%white”

Using this, we create a new variable: “%non-white”

FIRST, WE DELETE UNNECESSARY VARIABLES

```
dp_merge <- select(dp_merge, -13:-17)   # deleting 13th to 17th columns
dp_merge
```

← To create nonwhite variable, we don't need all race categories (i.e., black, indian, asian, hawaiian, other race)

```
## # A tibble: 372 × 12
##    cty_code cty_name       yr pct_b…¹ pct_u…² income pct_p…³ pop   age   pct_65
##       <dbl> <chr>       <dbl>   <dbl>   <dbl>  <dbl>   <dbl> <dbl> <dbl>   <dbl>
##  1    36001 Albany       2010    37.6     6    30863    12.6 3.04e5  38.1    13.8
##  2    36003 Allegany     2010    18.6     8.4  20058    16.5 4.91e4  37.1    14.8
##  3    36005 Bronx        2010    17.6    12.1  17575    28.4 1.37e6  32.6    10.4
##  4    36007 Broome       2010    25.1     6.8  24314    15.5 2.01e5  39.8    16.2
##  5    36009 Cattaraugus  2010    18.1     8.4  20824    16   8.08e4  40.3    15.1
##  6    36011 Cayuga       2010    18.4     6.5  22959    12.1 8.04e4  40.7    15
##  7    36013 Chautauqua   2010    20.3     8.1  21033    17.1 1.35e5  40.3    16.3
##  8    36015 Chemung      2010    20.9     7.9  23457    15.2 8.87e4  40.6    15.5
##  9    36017 Chenango     2010    17       8.1  22036    13.6 5.10e4  42.2    16
## 10    36019 Clinton      2010    21.7     8.1  22660    13.3 8.24e4  38.1    12.8
## # … with 362 more rows, 2 more variables: pct_female <dbl>, pct_white <dbl>,
```

NOW, CREATE NEW VARIABLE: %NONWHITE

```
dp_merge_ratio <- dp_merge %>%
                  mutate(pct_nonwhite = 100-pct_white) %>%
                  na.omit() %>%
                  group_by(yr) %>%
                  summarise(avg_nonwhite=mean(pct_nonwhite))
dp_merge_ratio
```

```
## # A tibble: 6 × 2
##      yr avg_nonwhite
##   <dbl>         <dbl>
## 1  2010          12.0
## 2  2011          12.1
## 3  2012          12.1
## 4  2013          12.3
## 5  2014          12.4
## 6  2015          12.6
```

SUBSET DATA:
select() & filter()

SELECT OBSERVATIONS

```
dp_merge_reduced <- dp_merge %>%
                    select(cty_code, cty_name, income, pop, age, yr) %>%
                    filter (age > 40) %>%
                    na.omit()

dp_merge_reduced
```

```
## # A tibble: 226 × 6
##    cty_code cty_name    income    pop   age    yr
##       <dbl> <chr>        <dbl>  <dbl> <dbl> <dbl>
##  1    36009 Cattaraugus  20824  80776  40.3  2010
##  2    36011 Cayuga       22959  80431  40.7  2010
##  3    36013 Chautauqua   21033 135263  40.3  2010
##  4    36015 Chemung      23457  88708  40.6  2010
##  5    36017 Chenango     22036  51045  42.2  2010
##  6    36021 Columbia     31844  63241  44.2  2010
##  7    36025 Delaware     22928  48220  44.6  2010
##  8    36031 Essex        24390  39416  43.6  2010
##  9    36035 Fulton       23147  55497  42    2010
## 10    36037 Genesee      24323  59952  40.9  2010
```

Pick these variables

- cty_code
- cty_name
- income
- pop
- age
- yr

& we don't want to include missing cases – so we omit them!

COUNTING OBSERVATIONS: **count()**

OF UNIQUE OBSERVATION: **n_distinct()**

COUNTING OBSERVATIONS, BY GROUP

```
dp_merge_count <- dp_merge %>%
                  filter(pct_unemp>8.0) %>%
                  group_by(yr) %>%
                  summarise(count=n()) %>%
                  head()
dp_merge_count
```

```
## # A tibble: 6 × 2
##      yr count
##   <dbl> <int>
## 1  2010    17
## 2  2011    22
## 3  2012    31
## 4  2013    39
## 5  2014    34
## 6  2015    23
```

COUNTING DISTINCT OBSERVATIONS

```
dp_distinct <- dp_merge %>%
               select(cty_code, cty_name) %>%
               summarise(n=n(),
                         n_county=n_distinct(cty_name))
dp_distinct
```

```
## # A tibble: 1 × 2
##       n n_county
##   <int>    <int>
## 1   372       62
```

SUMMARY STATS:
group_by() and **summarize()**

REMOVING MISSING VALUES:
na.omit()

Collapse observations by a specified grouping

SUMMARY STATS, WITHOUT NA, BY GROUP

```
dp_merge_ratio <- dp_merge %>%
                  mutate(pct_nonwhite = 100-pct_white) %>%
                  na.omit() %>%
                  group_by(yr) %>%
                  summarise(avg_nonwhite=mean(pct_nonwhite))
dp_merge_ratio
```

← By each year, we calculate the mean value of “%nonwhite” without missing data

```
## # A tibble: 6 × 2
##      yr avg_nonwhite
##   <dbl>        <dbl>
## 1  2010         12.0
## 2  2011         12.1
## 3  2012         12.1
## 4  2013         12.3
## 5  2014         12.4
## 6  2015         12.6
```

SORT/REORDER OBSERVATIONS: **arrange()**

SORTING OBSERVATIONS

```
dp_merge_arrange2 <- dp_merge %>%
                     arrange(desc(cty_code), yr)
dp_merge_arrange2
```

cty_code: in descending order
yr: in ascending order

```
## # A tibble: 372 × 17
##    cty_code cty_name    yr pct_bache…¹ pct_u…² income pct_p…³   pop   age pct_65
##       <dbl> <chr>    <dbl>       <dbl>   <dbl>  <dbl>   <dbl> <dbl> <dbl>  <dbl>
##  1    36123 Yates     2010        22.1     5.3  23255    14.7 25250  39.9   16.3
##  2    36123 Yates     2011        23.4     4.9  23928    15.4 25342  40.2   16.6
##  3    36123 Yates     2012        23.1     5.5  24124    16   25351  40.4   16.6
##  4    36123 Yates     2013        23.2     6.9  24457    15.9 25293  40.5   16.8
##  5    36123 Yates     2014        24.3     7    25436    15.4 25281  40.9   17.3
##  6    36123 Yates     2015        23.7     6.9  25224    14.4 25187  41.5   17.8
##  7    36121 Wyoming   2010        14.8     6.5  20605    10.9 42367  40.1   13.1
##  8    36121 Wyoming   2011        14.6     6.8  21762    10.1 42223  40.8   13.7
##  9    36121 Wyoming   2012        14.5     7.4  22369    10.5 42092  40.7   14
## 10    36121 Wyoming   2013        14.6     7.6  22700    10.5 41923  41.1   14.3
```

SORTING OBSERVATIONS

```
dp_merge_arrange4 <- dp_merge %>%
                     arrange(cty_code, yr)
dp_merge_arrange4
```

cty_code, yr: both in ascending order

```
## # A tibble: 372 × 17
##    cty_code cty_name    yr pct_bach…¹ pct_u…² income pct_p…³    pop   age pct_65
##       <dbl> <chr>    <dbl>      <dbl>   <dbl>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>
##  1    36001 Albany    2010       37.6     6    30863    12.6 304032  38.1   13.8
##  2    36001 Albany    2011       37.8     6.5  31728    12.8 303957  38.1   13.9
##  3    36001 Albany    2012       38.2     7    31924    13.1 304511  38.2   14.1
##  4    36001 Albany    2013       38.8     7.3  32328    13   305279  38     14.3
##  5    36001 Albany    2014       38.7     6.8  32624    13.6 306124  37.9   14.6
##  6    36001 Albany    2015       38.6     6.4  32779    13.5 307463  37.9   14.9
##  7    36003 Allegany  2010       18.6     8.4  20058    16.5  49105  37.1   14.8
##  8    36003 Allegany  2011       17.4     8.6  20047    16.6  48991  37.6   15
##  9    36003 Allegany  2012       18.2     8.8  20571    17.1  48837  37.8   15.3
## 10    36003 Allegany  2013       18.9     9.5  20978    16.5  48618  38.2   15.6
```