Data Science for Public Policy

Xiyu Zhang - xz551

PPOL 670 | Assignment 3
Applied Introduction to R's Tidyverse

```
library(tidyverse)
library(ipumsr)
library(srvyr)
library(haven)
```

Exercise 01 (1 point)

```
library(ipumsr)
library(tidyverse)
ddi <- read_ipums_ddi("cps_00002.xml")
data <- read_ipums_micro(ddi)
```

```
## Use of data from IPUMS CPS is subject to conditions including that users should
## cite the data appropriately. Use command 'ipums_conditions()' for more details.
```

```
glimpse(data)
```

```
## Rows: 212,608
## Columns: 26
## $ YEAR      <dbl> 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, 2020, ~
## $ SERIAL    <dbl> 1, 2, 2, 2, 3, 5, 5, 5, 6, 7, 7, 8, 9, 9, 11, 11, 12, 12, 14~
## $ MONTH     <int+lbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,~
## $ HWTFINL   <dbl> 1796.811, 2923.587, 2923.587, 2923.587, 1792.269, 1912.517, ~
## $ CPSID     <dbl> 2.02003e+13, 2.02002e+13, 2.02002e+13, 2.02002e+13, 2.02001e~
## $ PERNUM    <dbl> 1, 1, 2, 3, 1, 1, 2, 3, 1, 1, 2, 1, 1, 2, 1, 2, 1, 2, 1, 1, ~
## $ WTFINL    <dbl> 1796.811, 2923.587, 3618.171, 4634.866, 1792.269, 1912.517, ~
## $ CPSIDP    <dbl> 2.02003e+13, 2.02002e+13, 2.02002e+13, 2.02002e+13, 2.02001e~
## $ RELATE    <int+lbl> 101, 101, 301, 501, 101, 101, 1260, 1260, 101, 101, 202,~
## $ AGE       <int+lbl> 72, 21, 1, 27, 59, 67, 33, 33, 48, 80, 80, 80, 44, 19, 7~
## $ SEX       <int+lbl> 2, 2, 1, 1, 2, 1, 2, 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 1, 2,~
## $ RACE      <int+lbl> 100, 200, 200, 200, 200, 100, 100, 100, 100, 100, 100, 1~
## $ MARST     <int+lbl> 4, 4, 9, 6, 5, 3, 4, 6, 6, 1, 1, 6, 6, 6, 1, 1, 1, 1, 4,~
```

```
## $ POPSTAT   <int+lbl> 1, 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ EMPSTAT   <int+lbl> 10, 21, 0, 21, 10, 10, 34, 34, 10, 36, 10, 36, 10, 34, 1~
## $ LABFORCE  <int+lbl> 2, 2, 0, 2, 2, 2, 1, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 2, 1,~
## $ OCC       <dbl> 2205, 4720, 0, 710, 3255, 2040, 0, 0, 3930, 0, 4920, 0, 9645~
## $ IND       <dbl> 7870, 8680, 0, 4670, 8191, 9160, 0, 0, 4971, 0, 7071, 0, 629~
## $ UHRSWORKT <int+lbl> 40, 999, 999, 999, 40, 65, 999, 999, 997, 999, 997, 999,~
## $ UHRSWORK1 <int+lbl> 40, 999, 999, 999, 40, 65, 999, 999, 997, 999, 997, 999,~
## $ UHRSWORK2 <int+lbl> 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 9~
## $ AHRSWORKT <dbl+lbl>  40, 999, 999, 999,  40,  40, 999, 999,  45, 999,  24, 9~
## $ AHRSWORK1 <int+lbl> 40, 999, 999, 999, 40, 40, 999, 999, 45, 999, 24, 999, 4~
## $ AHRSWORK2 <int+lbl> 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 999, 9~
## $ ABSENT    <int+lbl> 0, 2, 0, 2, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 3, 1, 0, 0, 1,~
## $ WKSTAT    <int+lbl> 11, 60, 99, 50, 11, 11, 99, 99, 11, 99, 41, 99, 11, 99, ~
```

Exercise 02 (2 points)

#2.

```
library(srvyr)
cps_svy <- as_survey_design(data, weights = WTFINL)
class(data)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

```
class(cps_svy)
```

```
## [1] "tbl_svy"        "survey.design2" "survey.design"
```

#3. The difference is that, the as_survey_design function turn the tibble into a "tbl_svy", object, and create two new catagories called "survey.design" & "survey.design2"

#4. Yes, there is a value of NIU: #for UHRSWORKT, 1999-onward ASEC:0 = No hours; 997 = Hours vary; 999 = Not in universe (NIU) #for AHRSWORKT, 999 = NIU (Not in universe)

```
NIU.UHR.AHR <- data %>%
  group_by(YEAR, UHRSWORKT, AHRSWORKT) %>%
  summarise(
    unweighted = n()
  ) %>%
  filter(UHRSWORKT == 999 | AHRSWORKT == 999) %>%
  arrange(YEAR, desc(UHRSWORKT))
NIU.UHR.AHR
```

```
## # A tibble: 128 x 4
## # Groups:   YEAR, UHRSWORKT [128]
##     YEAR        UHRSWORKT                  AHRSWORKT unweighted
##    <dbl>        <int+lbl>                  <dbl+lbl>      <int>
##  1  2020 999 [NIU]        999 [NIU (Not in universe)]      59895
##  2  2020 997 [Hours vary] 999 [NIU (Not in universe)]        345
##  3  2020 100              999 [NIU (Not in universe)]          2
##  4  2020  99              999 [NIU (Not in universe)]          1
##  5  2020  90              999 [NIU (Not in universe)]          5
##  6  2020  89              999 [NIU (Not in universe)]          1
##  7  2020  84              999 [NIU (Not in universe)]          4
##  8  2020  82              999 [NIU (Not in universe)]          1
##  9  2020  80              999 [NIU (Not in universe)]         12
## 10  2020  74              999 [NIU (Not in universe)]          1
## # ... with 118 more rows
```

#observed that for all observations in 2020 and 2021, while UHRSWORKT == 999, then AHRSWORKT is inevitabily 999, that means to sum up the observations that are not in the universe in these two years, we could only group_by the AHRSWORKT variable. Rewrite our code:

```
NIU.UHR.AHR <- data %>%
  group_by(YEAR, AHRSWORKT) %>%
  summarise(
    unweighted = n()
  ) %>%
  filter(AHRSWORKT == 999) %>%
  arrange(YEAR, desc(AHRSWORKT))
NIU.UHR.AHR
```

```
## # A tibble: 2 x 3
## # Groups:   YEAR [2]
##    YEAR                   AHRSWORKT unweighted
##   <dbl>                   <dbl+lbl>      <int>
## 1  2020 999 [NIU (Not in universe)]      63321
## 2  2021 999 [NIU (Not in universe)]      61703
```

#Thus, in 2020, there were 63321 observations that are not in the universe for UHRSWORKT and AHRSWORKT; in 2021, there were 61703 of them.

#5.Filter out observations not in the universe for UHRSWORKT and create a new data frame called cps_subset_svy with the assignment operator.

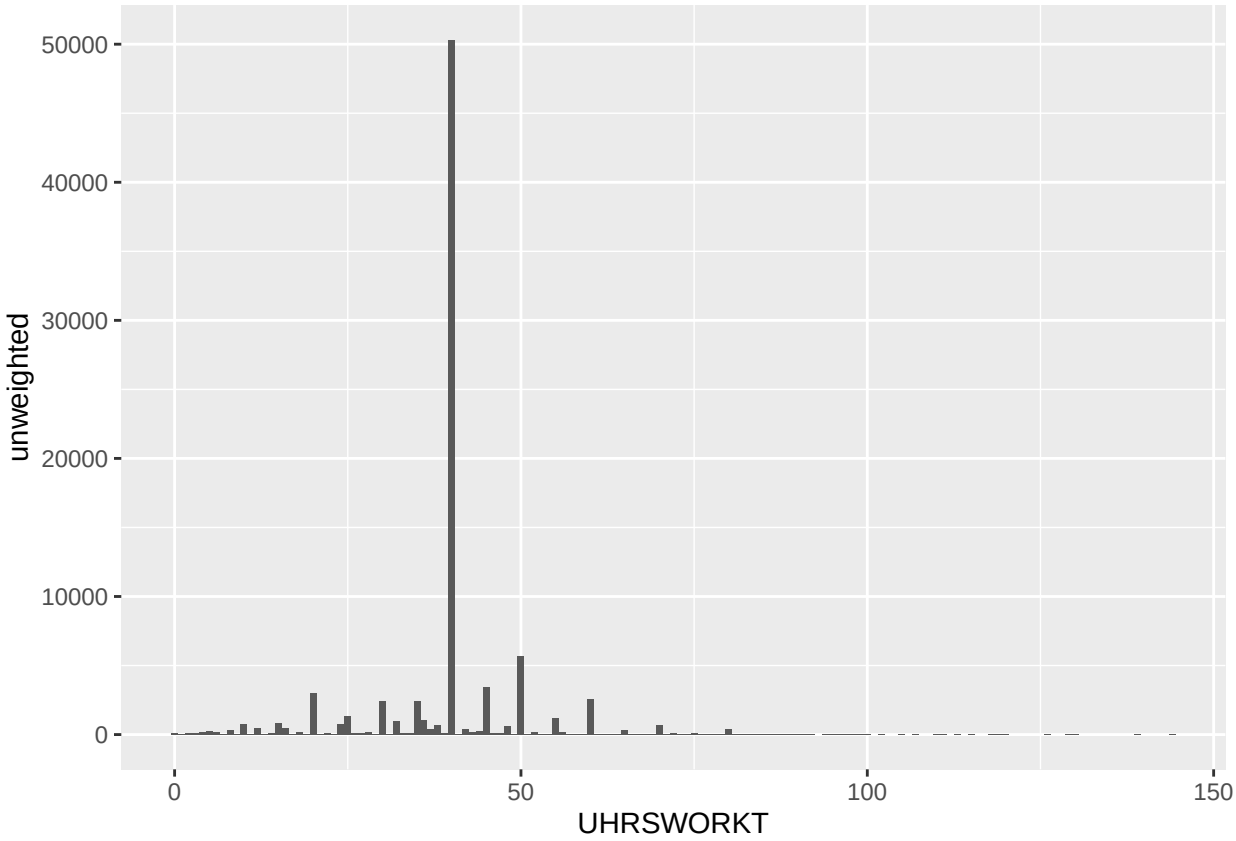
```
cps_subset_svy <- data %>%
  filter(UHRSWORKT != 999)
cps_subset_svy
```

```
## # A tibble: 92,684 x 26
##     YEAR SERIAL     MONTH HWTFINL   CPSID PERNUM WTFINL  CPSIDP     RELATE   AGE
##    <dbl>  <dbl> <int+lbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>  <int+lbl> <int>
##  1  2020      1 4 [April]   1797. 2.02e13      1  1797. 2.02e13 101 [Head~    72
##  2  2020      3 4 [April]   1792. 2.02e13      1  1792. 2.02e13 101 [Head~    59
##  3  2020      5 4 [April]   1913. 2.02e13      1  1913. 2.02e13 101 [Head~    67
##  4  2020      6 4 [April]   2000. 2.02e13      1  2000. 2.02e13 101 [Head~    48
##  5  2020      7 4 [April]   1967. 2.02e13      2  1967. 2.02e13 202 [Oppo~    80
##  6  2020      9 4 [April]   3407. 2.02e13      1  3407. 2.02e13 101 [Head~    44
##  7  2020     11 4 [April]   1967. 2.02e13      1  1967. 2.02e13 101 [Head~    75
##  8  2020     12 4 [April]   2457. 2.02e13      1  2457. 2.02e13 101 [Head~    47
##  9  2020     12 4 [April]   2457. 2.02e13      2  2477. 2.02e13 202 [Oppo~    45
## 10  2020     20 4 [April]   1794. 2.02e13      2  1940. 2.02e13 202 [Oppo~    46
## # ... with 92,674 more rows, and 16 more variables: SEX <int+lbl>,
## #   RACE <int+lbl>, MARST <int+lbl>, POPSTAT <int+lbl>, EMPSTAT <int+lbl>,
## #   LABFORCE <int+lbl>, OCC <dbl>, IND <dbl>, UHRSWORKT <int+lbl>,
## #   UHRSWORK1 <int+lbl>, UHRSWORK2 <int+lbl>, AHRSWORKT <dbl+lbl>,
## #   AHRSWORK1 <int+lbl>, AHRSWORK2 <int+lbl>, ABSENT <int+lbl>,
## #   WKSTAT <int+lbl>
```

#6.Count the number of unweighted responses for each value for UHRSWORKT.

```
cps_subset_6 <- cps_subset_svy %>%
  group_by(UHRSWORKT) %>%
  filter(UHRSWORKT != 997) %>%
  summarize (
```

```
    unweighted = n()
  )
ggplot(data = cps_subset_6) +
  geom_col(mapping = aes(x = UHRSWORKT, y = unweighted))
```



#7.Calculate the mean usual hours worked (UHRSWORKT) in 2020 and 2021. Exclude workers with "Hours vary".

```
cps_subset_7 <- cps_subset_svy %>%
  group_by(YEAR) %>%
  filter(UHRSWORKT != 997) %>%
  summarise(mean_UHRSWORKT = mean(UHRSWORKT))
cps_subset_7
```

```
## # A tibble: 2 x 2
##    YEAR mean_UHRSWORKT
##   <dbl>          <dbl>
## 1  2020           39.7
## 2  2021           39.4
```

#8.Calculate the proportion of workers who usually worked exactly 40 hours in 2021.

```
cps_subset_8 <- cps_subset_svy %>%
  filter(YEAR == 2021) %>%
```

```
  mutate(
    HOURS_40 = if_else(condition = UHRSWORKT == 40, true = 1, false = 0)
    ) %>%
  group_by(YEAR) %>%
  summarise(proportion_40 = mean(HOURS_40))
cps_subset_8
```

```
## # A tibble: 1 x 2
##    YEAR proportion_40
##   <dbl>         <dbl>
## 1  2021         0.534
```

#9.Calculate the proportion of workers who worked less, the same, and more than usual in April 2020 and April 2021 (separately) by comparing UHRSWORKT and AHRSWORKT.

```
cps_subset_91 <- cps_subset_svy %>%
  filter(UHRSWORKT != 997) %>%
  select(YEAR, UHRSWORKT, AHRSWORKT) %>%
  mutate(
    AprilWork = case_when(
      AHRSWORKT > UHRSWORKT ~ "work more",
      AHRSWORKT < UHRSWORKT ~ "work less",
      TRUE ~ "work the same"
    )
  )
cps_subset_92 <- as_survey_design(cps_subset_91) %>%
  group_by(YEAR, AprilWork) %>%
  summarise(prop = survey_prop())
cps_subset_92
```

```
## # A tibble: 6 x 4
## # Groups:   YEAR [2]
##    YEAR AprilWork       prop prop_se
##   <dbl> <chr>          <dbl>   <dbl>
## 1  2020 work less     0.172  0.00191
## 2  2020 work more     0.150  0.00181
## 3  2020 work the same 0.678  0.00236
## 4  2021 work less     0.0970 0.00136
## 5  2021 work more     0.109  0.00143
## 6  2021 work the same 0.794  0.00186
```

Exercise 03 (2 points)

#2. Using cps_svy, calculate the weighted count of observations ages 16 or older for each year with survey_count(). Your results should match the April 2020 and April 2021 values to the closest thousand because of rounding. #Sorry I don't have any idea about "survey_count", it seems not appear when I was trying to search it. I also don't know how to put the age >= 16 into weight.

#3. Create three numeric indicator variables.

```
cps_svy_33 <- cps_svy %>%
  mutate(
    labor_force = case_when(
      LABFORCE == 2 ~ 1,
      LABFORCE == 1 ~ 0,
      TRUE ~ 999
    )
  ) %>%
  mutate(
    employed = case_when(
      EMPSTAT == 10 | EMPSTAT == 12 ~ 1,
      EMPSTAT == 20 | EMPSTAT == 21 | EMPSTAT == 22 ~ 0,
      TRUE ~ 999
    )
  ) %>%
  mutate(
    unemployed = case_when(
      EMPSTAT == 10 | EMPSTAT == 12 ~ 0,
      EMPSTAT == 20 | EMPSTAT == 21 | EMPSTAT == 22 ~ 1,
      TRUE ~ 999
    )
  )
```

#4.Filter to the civilian population ages 16 or older, and calculate the relavant population.

```
cps_svy_34 <- cps_svy_33 %>%
  filter(AGE >= 16) %>%
  filter(POPSTAT == 1) %>%
  summarise(
    LABFOR_cal = survey_total(labor_force == 1),
    EMPLOY_cal = survey_total(employed == 1),
    UNEMPLOY_cal = survey_total(unemployed == 1)
            )
cps_svy_34
```

```
## # A tibble: 1 x 6
##   LABFOR_cal LABFOR_cal_se EMPLOY_cal EMPLOY_cal_se UNEMPLOY_cal UNEMPLOY_cal_se
##        <dbl>         <dbl>      <dbl>         <dbl>        <dbl>           <dbl>
## 1 316601435.       922160. 284601546.       891906.    31999889.         374905.
```

#5. Compare the result with microdata to the official tabulation for April 2021. #The official data is: civilian labor force - 160,379,000; employed - 151,160,000; unemployed - 9,220,000. #What I got here are (shown above): civilian labor force - 316,601,435; employed - 284,601,546; unemployed - 31,999,889. They are far from "close", thus there might be some problems in my code.

Exercise 04 (2 points)

#1.-3

```
cps2021 <- filter(data, YEAR == 2021)
cps2021_sum <- cps2021 %>%
  summarise(n())
cps2021_sum
```

```
## # A tibble: 1 x 1
##    `n()`
##    <int>
## 1 111003
```

```
nber <- read_dta("cpsb202104.dta")
nber_sum <- nber %>%
  select(hufinal, pwcmpwgt) %>%
  summarise(n())
nber_sum
```

```
## # A tibble: 1 x 1
##    `n()`
##    <int>
## 1 133449
```

#3. Notice that n = 133,449. There are 133,449 observations in dataset "nber", while there are 111,003 observations in dataset "cps2021".

#4. As it shows, there are 111,003 observations in nber_44, which match with the amount of the observations in dataset "cps2021".

```
nber_44 <- filter(nber, hufinal <= 205)
summarise(nber_44, n())
```

```
## # A tibble: 1 x 1
##    `n()`
##    <int>
## 1 111003
```

#5. Combine the columns.

```
nber_cps <- bind_cols(cps2021, nber_44)
nber_cps
```

```
## # A tibble: 111,003 x 418
##     YEAR SERIAL     MONTH HWTFINL   CPSID PERNUM WTFINL  CPSIDP     RELATE   AGE
##    <dbl>  <dbl> <int+lbl>   <dbl>   <dbl>  <dbl>  <dbl>   <dbl>  <int+lbl> <int>
##  1  2021      1 4 [April]   1381. 2.02e13      1  1381. 2.02e13  101 [Hea~    35
##  2  2021      1 4 [April]   1381. 2.02e13      2  1757. 2.02e13  202 [Opp~    40
##  3  2021      1 4 [April]   1381. 2.02e13      3  2165. 2.02e13  301 [Chi~     5
##  4  2021      1 4 [April]   1381. 2.02e13      4  1564. 2.02e13  301 [Chi~     6
```

```
##  5  2021      3 4 [April]   1783. 2.02e13      1  1783. 2.02e13  101 [Hea~    73
##  6  2021      6 4 [April]   2051. 2.02e13      1  2051. 2.02e13  101 [Hea~    69
##  7  2021      6 4 [April]   2051. 2.02e13      2  1827. 2.02e13 1260 [Oth~    34
##  8  2021      6 4 [April]   2051. 2.02e13      3  1366. 2.02e13 1260 [Oth~    34
##  9  2021      7 4 [April]   1685. 2.02e13      1  1685. 2.02e13  101 [Hea~    49
## 10  2021      8 4 [April]   1859. 2.02e13      1  1650. 2.02e13  101 [Hea~    80
## # ... with 110,993 more rows, and 408 more variables: SEX <int+lbl>,
## #   RACE <int+lbl>, MARST <int+lbl>, POPSTAT <int+lbl>, EMPSTAT <int+lbl>,
## #   LABFORCE <int+lbl>, OCC <dbl>, IND <dbl>, UHRSWORKT <int+lbl>,
## #   UHRSWORK1 <int+lbl>, UHRSWORK2 <int+lbl>, AHRSWORKT <dbl+lbl>,
## #   AHRSWORK1 <int+lbl>, AHRSWORK2 <int+lbl>, ABSENT <int+lbl>,
## #   WKSTAT <int+lbl>, hrhhid <dbl>, hrmonth <dbl>, hryear4 <dbl>,
## #   hurespli <dbl>, hufinal <dbl+lbl>, hetenure <dbl+lbl>, ...
```

#6.-9. adjust the weights to previous results.

```
nber_cps_svy <- nber_cps %>%
  mutate(pwcm_cal = pwcmpwgt/10000) %>%
  as_survey_design(weights = pwcm_cal) %>%
  mutate(
    labor_force = case_when(
      LABFORCE == 2 ~ 1,
      LABFORCE == 1 ~ 0,
      TRUE ~ 999
    )
  ) %>%
  mutate(
    employed = case_when(
      EMPSTAT == 10 | EMPSTAT == 12 ~ 1,
      EMPSTAT == 20 | EMPSTAT == 21 | EMPSTAT == 22 ~ 0,
      TRUE ~ 999
    )
  ) %>%
  mutate(
    unemployed = case_when(
      EMPSTAT == 10 | EMPSTAT == 12 ~ 0,
      EMPSTAT == 20 | EMPSTAT == 21 | EMPSTAT == 22 ~ 1,
      TRUE ~ 999
    )
  ) %>%
  filter(AGE >= 16) %>%
  filter(POPSTAT == 1) %>%
  summarise(
    LABFOR_cal = survey_total(labor_force == 1),
    EMPLOY_cal = survey_total(employed == 1),
    UNEMPLOY_cal = survey_total(unemployed == 1)
           )
nber_cps_svy
```

```
## # A tibble: 1 x 6
##   LABFOR_cal LABFOR_cal_se EMPLOY_cal EMPLOY_cal_se UNEMPLOY_cal UNEMPLOY_cal_se
##        <dbl>         <dbl>      <dbl>         <dbl>        <dbl>           <dbl>
## 1 160379460.       629281. 151159727.       619143.     9219734.         194338.
```

#10.The adjust results came out to be:labor force = 160,379,460; employed = 151,159,727; unemployed = 9,219,734. #Compared with the official data: civilian labor force = 160,379,000; employed = 151,160,000; unemployed = 9,220,000. #Now my results are very close to the official one.

Exercise 05 (2 points)

```
library(ggplot2)
p <- ggplot() +
  geom_bar(data = cps_subset_92,
           aes(x = AprilWork, y = prop, fill = YEAR),
           stat = "identity",
           position = position_dodge())
p + ggtitle(label = "Working Hours 2020 & 2021",
  subtitle = "the proportion of workers comparing working hours between April 2020 and April 2021") +
  labs(caption = "Data source: CPS") +
  xlab("April Work (type)") + ylab("proportion")
```