Day 4 : Webscrapping, survey weight handling and workshop

Session 13: Webscrapping, duplicates, and missing data

13.1 Webscrapping

E028-webscrapping.R

```
#-------------------------------------------------------------------
# importing csv data directly from the web
#-------------------------------------------------------------------
df <- read.csv("http://s.anilz.net/wb_energy")
head(df)

dx <- read.csv("https://data.ny.gov/api/views/d6yy-54nr/rows.csv")
head(dx)

#-------------------------------------------------------------------
# Using rvest package for static website scraping
#-------------------------------------------------------------------
#loading necessary packages
library(rvest) #see https://rvest.tidyverse.org/articles/harvesting-the-
web.html for details
library(dplyr)

#loading webpage content
webpage <- read_html("https://www.sharesansar.com/today-share-price")

#extracting table from the webpage
tables <- html_table(webpage)

#checking the number of tables available in the webpage
length(tables)

df1 <- tables[[1]]
head(df1)

#filtering upper and lower circuit stock
filtered_df1 <- df1 %>% filter(`Diff %` > 9 | `Diff %`< -9) %>%
arrange(`Diff %`)
filtered_df1

#************************************
#*Obtaining Forex information from NRB
#************************************
#loading webpage content
webpage <- read_html("https://www.nrb.org.np")

#extracting table from the webpage
tables <- html_table(webpage)

#checking the number of tables available in the webpage
length(tables)


df1 <- tables[[1]]
df2 <- tables[[2]]

df1
df2

#keeping USD and JPY only
filtered_df1 <- df1 %>% filter(Currency=='USD' | Currency =='JPY')
filtered_df1
```

Task 6 :

Web-scrape the Historical ranking table from https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings

```
webpage <-
read_html("https://en.wikipedia.org/wiki/ICC_Men%27s_T20I_Team_Rankings")
tables <- html_table(webpage)
length(tables)

df <- tables[[7]]
```

13.2 Finding duplicates

E029-duplicates.R

```
library(dplyr)

# Creating a sample data frame
df <- data.frame(
  ID = c(1, 2, 3, 3, 4, 5, 4, 3),
  Name = c("John", "Jane", "Mark", "Mark", "Luke", "Kate", "Luke", "Mark"),
  Age = c(25, 30, 35, 35, 40, 45, 40, 35)
)

df
  ID Name Age
1  1 John  25
2  2 Jane  30
3  3 Mark  35
4  3 Mark  35
5  4 Luke  40
6  5 Kate  45
7  4 Luke  40
8  3 Mark  35

#showing the duplicated observations
df %>% filter(duplicated(.) == T)
  ID Name Age
1  3 Mark  35
2  4 Luke  40
3  3 Mark  35

#removing the duplicated observations
df %>% filter(!duplicated(.) == T)
  ID Name Age
1  1 John  25
2  2 Jane  30
3  3 Mark  35
4  4 Luke  40
5  5 Kate  45

#counting the duplicate observations
df %>% group_by_all() %>%
  summarise(count = n()) %>%
  filter(count > 1)
# A tibble: 2 × 4
     ID Name    Age count
  <dbl> <chr> <dbl> <int>
1     3 Mark     35     3
2     4 Luke     40     2

#identifying duplicate values based on some variables
library(haven)
df <- read_dta('data/008-nlfs2.dta')

#checking whether there is any duplicate based on selected variables
df %>% count(psu, hhid) %>% filter(n > 1)
```

```
    psu  hhid     n
  <dbl> <dbl> <int>
1  1001     1     4
2  1001     2     2
3  1001     7     2
4  1001     9     3
5  1001    10     2
6  1001    12     2
7  1001    13     2
8  1001    16     3
9  1001    17     2
0  1001    18     2
i 10,937 more rows
i Use `print(n = ...)` to see more rows

#selecting observations that does not have duplicate based on selected
variables
df %>% filter(duplicated(psu, hhid) == F) %>% count(psu, hhid)
    psu  hhid     n
  <dbl> <dbl> <int>
1  1001     1     1
2  1002     3     1
3  1003    13     1
4  1004    15     1
5  1005    16     1
6  1006    20     1
7  1007    11     1
8  1008    12     1
9  1009    19     1
0  1010     6     1
i 789 more rows
i Use `print(n = ...)` to see more rows
```

13.3 Finding missing values

E030-missing_values.R

```
library(haven)
library(dplyr)
df <- read_dta('data/008-nlfs2.dta')
df <- df[c('psu','hhid','q13','q18')]

#Selecting observations with no missing values
df %>% filter(complete.cases(.) == T)

#Selecting observations with missing values
df %>% filter(complete.cases(.) == F)
```

Session 14: Using survey weights

```
library(survey) #for survey data
library(haven) #for importing Stata dta file
library(srvyr) # srvyr' for tidyverse-style coding (easier to read)

#function for directly accessing the data from the web
get_url <- function(x) {
  return(paste0("https://github.com/tempgita/training/raw/refs/heads/master/R%20trainin
g%202082-10-03/data/",x))
}

df <- read_stata(get_url('013-poverty.dta'))
```

```r
svy_design <- svydesign(
  ids = ~psu_number,
  strata = ~domain,
  weights = ~hhs_wt,
  data = df
)

svy_obj <- as_survey(svy_design)

#*=====================================================
#*Population size calculation
#*=====================================================
#The manual way
df %>% summarise(hh_pop = sum(hhs_wt),
                 ind_pop = sum(ind_wt))
```

```
# A tibble: 1 × 2
    hh_pop   ind_pop
     <dbl>     <dbl>
1 7185103. 28740504.
```

```r
df %>%
  group_by(prov) %>%
  summarise(hh_pop = sum(hhs_wt),
                 ind_pop = sum(ind_wt))
```

```
  prov                hh_pop  ind_pop
  <dbl+lbl>            <dbl>    <dbl>
1 1 [Koshi]         1216891. 4675310.
2 2 [Madhesh]       1347877. 6486147.
3 3 [Bagmati]       1622908. 5869481.
4 4 [Gandaki]        695063. 2396728.
5 5 [Lumbini]       1354403. 5447377.
6 6 [Karnali]        372795. 1473115.
7 7 [Sudurpaschim]   575165. 2392344.
```

```r
#using the survey object
svy_obj %>%
  summarise(pop_size = survey_total())
```

```
  pop_size pop_size_se
     <dbl>       <dbl>
1 7185103.     206632.
```

```r
svy_obj %>%
  group_by(prov) %>%
  summarise(pop_size = survey_total())
```

```
  prov              pop_size pop_size_se
  <dbl+lbl>            <dbl>       <dbl>
1 1 [Koshi]         1216891.      42103.
2 2 [Madhesh]       1347877.     166259.
3 3 [Bagmati]       1622908.      67348.
4 4 [Gandaki]        695063.      20815.
5 5 [Lumbini]       1354403.      84838.
6 6 [Karnali]        372795.      12288.
7 7 [Sudurpaschim]   575165.      31059.
```

```r
#*=====================================================
#* Mean calculation
#*=====================================================
# the manual way
df %>%
  summarise(pcep_mean = weighted.mean(pcep, hhs_wt))
```

```
  pcep_mean
      <dbl>
1   141032.
```

```r
df %>%
```

```
  group_by(prov) %>%
  summarise(pcep_mean = weighted.mean(pcep, hhs_wt))
```

prov	pcep_mean
<dbl+lbl>	<dbl>
1 [Koshi]	144113.
2 [Madhesh]	129734.
3 [Bagmati]	158703.
4 [Gandaki]	154778.
5 [Lumbini]	136939.
6 [Karnali]	130556.
7 [Sudurpaschim]	110946.

```
#using the survey object
svy_obj %>%
  summarise(mean_pcep = survey_mean(pcep))
```

mean_pcep	mean_pcep_se
<dbl>	<dbl>
141032.	2066.

```
svy_obj %>%
  group_by(prov) %>%
  summarise(mean_pcep = survey_mean(pcep))
```

prov	mean_pcep	mean_pcep_se
<dbl+lbl>	<dbl>	<dbl>
1 [Koshi]	144113.	4533.
2 [Madhesh]	129734.	5662.
3 [Bagmati]	158703.	4558.
4 [Gandaki]	154778.	4688.
5 [Lumbini]	136939.	4759.
6 [Karnali]	130556.	6919.
7 [Sudurpaschim]	110946.	4532.

```
#*=====================================================
#* Calculation of poor proportion
#*=====================================================
svy_obj %>%
  group_by(prov) %>%
  summarise(poverty_rate = survey_mean(poor,
                                       vartype = c("se", "ci"),
                                       level = 0.95)) # for CI at 5% level of
significance
```

prov	poverty_rate	poverty_rate_se	poverty_rate_low	poverty_rate_upp
<dbl+lbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 [Koshi]	0.145	0.0146	0.116	0.173
2 [Madhesh]	0.196	0.0171	0.162	0.229
3 [Bagmati]	0.106	0.0109	0.0850	0.128
4 [Gandaki]	0.0993	0.0146	0.0707	0.128
5 [Lumbini]	0.196	0.0196	0.157	0.234
6 [Karnali]	0.215	0.0213	0.173	0.257
7 [Sudurpaschim]	0.299	0.0241	0.252	0.347

```
#*=====================================================
#* Quantiles
#*=====================================================
# --- Using the Survey Object ---
# Calculate Median PCEP by Province
svy_obj %>%
  group_by(prov) %>%
  summarise(
    median_pcep = survey_median(pcep),
    q90 = survey_quantile(pcep, quantiles = 0.9)
  )
```

prov	median_pcep	median_pcep_se	q90_q90	q90_q90_se
<dbl+lbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 [Koshi]	121248.	3372.	244609.	10364.
2 [Madhesh]	109475.	3510.	221380.	10313.
3 [Bagmati]	133873.	3867.	266275.	9775.
4 [Gandaki]	135702.	3957.	258257.	8614.
5 [Lumbini]	114586.	4495.	236134.	10823.
6 [Karnali]	103770.	3530.	215739.	17857.
7 [Sudurpaschim]	93192.	3007.	188344.	12445.

```
#*=====================================================
#* Ratio Estimation
#*=====================================================
svy_obj %>%
  group_by(prov) %>%
  summarise(
    food_share = survey_ratio(numerator = pcep_food,
                              denominator = pcep)
  )
```

prov	food_share	food_share_se
<dbl+lbl>	<dbl>	<dbl>
1 [Koshi]	0.454	0.0108
2 [Madhesh]	0.434	0.0168
3 [Bagmati]	0.502	0.00821
4 [Gandaki]	0.501	0.0133
5 [Lumbini]	0.450	0.0101
6 [Karnali]	0.435	0.0166
7 [Sudurpaschim]	0.461	0.0107

```
#*=====================================================
#* t-test
#*=====================================================
svyttest(pcep ~ poor, design = svy_obj)
```

```
        Design-based t-test

data:  pcep ~ poor
t = -47.19, df = 784, p-value < 2.2e-16
alternative hypothesis: true difference in mean is not equal to 0
95 percent confidence interval:
 -104863.64  -96487.82
sample estimates:
difference in mean
         -100675.7
```

```
#OR
summary(svyglm(pcep ~ poor, design = svy_obj))
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   157834       2044   77.23   <2e-16 ***
poor         -100676       2133  -47.19   <2e-16 ***
```

Task: perform t-test between the mean pcep between Madhesh vs Karnali and Madhesh vs. Bagamati.

```
#*=====================================================
#* T-test (Comparing Means between two groups)
#*=====================================================
# --------- Madhesh vs Karnali -----------------
# Step 1: Create a subset design object for the two provinces
sub_obj <- svy_obj %>%
  filter(prov %in% c(2, 6)) %>%
  mutate(prov_binary =as.factor(prov)) #to make prov binary variable
```

```r
# Step 2: Run the test using the new binary variable
svyttest(pcep ~ prov_binary, design = sub_obj)
```

```
        Design-based t-test

data:  pcep ~ prov_binary
t = 0.091952, df = 195, p-value = 0.9268
alternative hypothesis: true difference in mean is not equal to 0
95 percent confidence interval:
 -16810.24  18454.41
sample estimates:
difference in mean
          822.0861
```

```r
#OR
summary(svyglm(pcep ~ prov_binary, design = sub_obj))
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  129734.0     5662.4  22.911   <2e-16 ***
prov_binary6    822.1     8940.4   0.092    0.927
```

```r
# --------- Madhesh vs Bagamati -----------------
# Step 1: Create a subset design object for the two provinces
sub_obj <- svy_obj %>%
  filter(prov %in% c(2, 3)) %>%
  mutate(prov_binary =as.factor(prov)) #to make prov binary variable

# Step 2: Run the test using the new binary variable
svyttest(pcep ~ prov_binary, design = sub_obj)
```

```
        Design-based t-test

data:  pcep ~ prov_binary
t = 3.9852, df = 279, p-value = 8.611e-05
alternative hypothesis: true difference in mean is not equal to 0
95 percent confidence interval:
 14659.87 43278.27
sample estimates:
difference in mean
          28969.07
```

```r
#OR
summary(svyglm(pcep ~ prov_binary, design = sub_obj))
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    129734       5662  22.911  < 2e-16 ***
prov_binary3    28969       7269   3.985 8.61e-05 ***
```