

Survey Data: Design and Examples

Ehsan Karim

2020-09-14

Contents

1	Outline	5
2	Model-based Approach	7
2.1	Example	7
2.2	Research question	7
2.3	Data	8
2.4	Checking assumptions	11
2.5	Analysis	13
2.6	Verdict	19
2.7	Exercises (Optional)	20
3	Design-based Approach	21
3.1	Sampling	21
3.2	Statistical inference	23
3.3	Complex surveys	25
3.4	Other ideas	28
3.5	Further readings	28
3.6	Exercise	28
4	Potential Data Sources	29
4.1	Survey data with features	29
4.2	Others	29

5	Importing CCHS to R	31
5.1	Downloading CCHS data from UBC	31
5.2	Reading and Formatting the data	38
5.3	Processing data in R	53
6	Demystifying NHANES	57
6.1	Overview	57
6.2	Survey history	57
6.3	NHANES datafile and documents	58
6.4	Exercise (web)	61
7	Importing NHANES to R	63
7.1	NHANES Dataset	63
7.2	Accessing NHANES Data	63

Chapter 1

Outline

- Review of Model-based Approach
- Introduction to Design-based Approach
- Complex survey design examples
- Canadian Community Health Survey - Annual Component (CCHS)
 - Data import to R
- National Health and Nutrition Examination Survey (NHANES)
 - Understanding NHANES data and documentation structure
 - Data import to R

Chapter 2

Model-based Approach

Review of regression analysis and ANOVA from pre-requisites (+ some extra concepts). Below we see an example of a random data generating process that depends on specification of a probability model. We assume that the population data was generated from a **Normal distribution**, and we are merely dealing with a sample. All our inferences (point estimate or hypothesis testing) will depend on how closely the data fulfill such assumption. We call such approach as ‘**model-based**’ approach.

2.1 Example

Does plant weight increase with added nutrition?

The following problem was taken from Exercise set 2.5 (2.1) from Dobson and Barnett (2008):

“Genetically similar seeds are randomly assigned to be raised in either a nutritionally enriched environment (treatment group) or standard conditions (control group) using a completely randomized experimental design. After a predetermined time all plants are harvested, dried and weighed.”

2.2 Research question

We want to test whether there is any difference in yield (weight) between the two groups

- plants from nutritionally enriched environment (treatment group) and
- plants from standard conditions (control group)

2.2.1 Notations

1. Let k be the index of each plant, and $k = 1, \dots, 20$ for both groups.
2. Let j be the index for groups. Here, $j = 1$ for the treatment group (**Trt**), $j = 2$ for the control group (**Ctl**).
3. Let Y_{jk} denote the k th observation of weights in the j th group.

2.2.2 Assumptions

1. Assume that the Y_{jk} 's are independent random variables with $Y_{jk} \sim N(\mu_j, \sigma^2)$.
2. We also assume that the variances are homogenous, that is, σ_1^2 and σ_2^2 are not very different (and could be pooled to one single value of σ^2).

2.2.3 Hypothesis

The null hypothesis $H_0 : \mu_1 = \mu_2 = \mu$, that there is no difference, is to be compared with the alternative hypothesis $H_1 : \mu_1 \neq \mu_2$.

2.3 Data

2.3.1 Data table

```
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
length(ctl);length(trt)
```

```
## [1] 10
```

```
## [1] 10
```

```
group <- rep(c("Ctl","Trt"), each = length(ctl))
group
```

```
## [1] "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Ctl" "Trt" "Trt"
## [13] "Trt" "Trt" "Trt" "Trt" "Trt" "Trt" "Trt" "Trt" "Trt"
```



```
mode(group)
```

```
## [1] "character"
```

```
weight <- c(ctl, trt)
weight
```

```
## [1] 4.17 5.58 5.18 6.11 4.50 4.61 5.17 4.53 5.33 5.14 4.81 4.17 4.41 3.59 5.87
## [16] 3.83 6.03 4.89 4.32 4.69
```

```
mode(weight)
```

```
## [1] "numeric"
```

```
Plant.Weight.Data <- data.frame(group=group, weight = c(ctl, trt))
mode(Plant.Weight.Data)
```

```
## [1] "list"
```

```
dim(Plant.Weight.Data)
```

```
## [1] 20  2
```

```
str(Plant.Weight.Data)
```

```
## 'data.frame': 20 obs. of  2 variables:
## $ group : chr  "Ctl" "Ctl" "Ctl" "Ctl" ...
## $ weight: num  4.17 5.58 5.18 6.11 4.5 4.61 5.17 4.53 5.33 5.14 ...
```

The results, expressed in grams, for 20 plants in each group are shown in the following Table.

```
library(DT)
```

```
## Warning: package 'DT' was built under R version 4.0.2
```

```
datatable(Plant.Weight.Data)
```

Show 10 entries Search

	group	id	weight
1	Ctl		4.17
2	Ctl		5.58
3	Ctl		5.18
4	Ctl		6.11
5	Ctl		4.5
6	Ctl		4.61
7	Ctl		5.17
8	Ctl		4.53
9	Ctl		5.33
10	Ctl		5.14

Showing 1 to 10 of 20 entries Previous 1 2 Next

2.3.2 Visualization

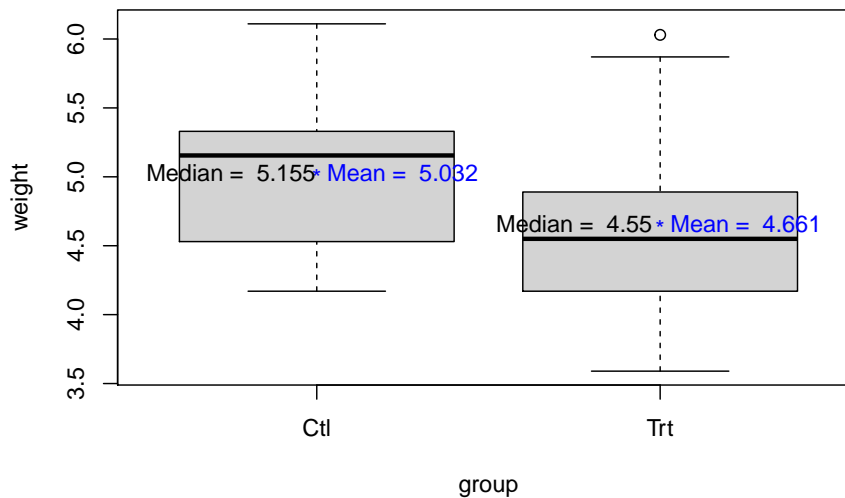
```
boxplot(weight ~ group, data=Plant.Weight.Data)
weight.means <- aggregate(weight ~ group, data=Plant.Weight.Data, FUN=mean)
weight.means
```

```
##   group weight
## 1   Ctl  5.032
## 2   Trt  4.661
```

```
weight.medians <- aggregate(weight ~ group, data=Plant.Weight.Data, FUN=median)
weight.medians
```

```
##   group weight
## 1   Ctl  5.155
## 2   Trt  4.550
```

```
points(1:2, weight.means$weight, pch = "*", col = "blue")
text(c(1:2)+0.25, weight.means$weight, labels =
      paste("Mean = ", weight.means$weight), col = "blue")
text(c(1:2)-0.25, weight.means$weight, labels =
      paste("Median = ", weight.medians$weight), col = "black")
```



Wait: so, plant weight reduces as we add nutrition? How confidently can we say that this added nutrition harmful for the plants (e.g., so that the weight will be reduced)?

2.4 Checking assumptions

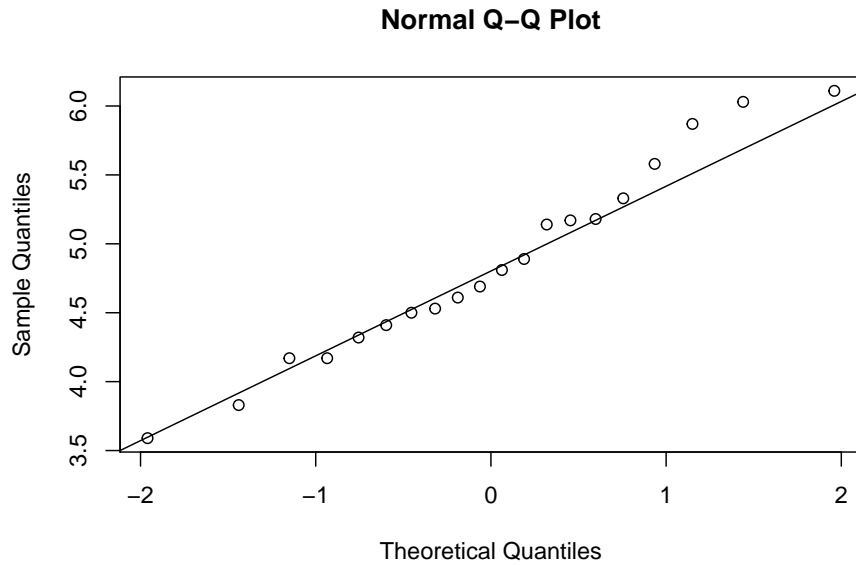
Test of normality of the outcomes (Shapiro-Wilk normality test):

```
shapiro.test(Plant.Weight.Data$weight)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Plant.Weight.Data$weight
## W = 0.97311, p-value = 0.8187
```

Therefore, we cannot reject the null hypothesis that samples come from a population which has a normal distribution. Also check a normal quantile-quantile plot:

```
qqnorm(Plant.Weight.Data$weight)
qqline(Plant.Weight.Data$weight)
```



Test of homogeneity of variances, that tests $H_0 : \sigma_1 = \sigma_2$ vs. $H_1 : \sigma_1 \neq \sigma_2$:

```
# SD from each groups
tapply(Plant.Weight.Data$weight,
       INDEX = Plant.Weight.Data$group, FUN = sd)
```

```
##          Ctl          Trt
## 0.5830914 0.7936757
```

```
bartlett.test(weight ~ group, data = Plant.Weight.Data) # Bartlett's test
```

```
##
## Bartlett test of homogeneity of variances
##
## data: weight by group
## Bartlett's K-squared = 0.79805, df = 1, p-value = 0.3717
```

```
# leveneTest(weight ~ group, data = Plant.Weight.Data) # Levene's test
```

2.5 Analysis

2.5.1 Two-sample t-test

A two-sample (independent) t-test compares the weights of control and treatment group as follows (assuming equal variance; judging from the IQR from the boxplots or the above Bartlett test):

```
ttest<- t.test(weight ~ group, data = Plant.Weight.Data,
               paired = FALSE, var.equal = TRUE)
ttest

##
## Two Sample t-test
##
## data: weight by group
## t = 1.1913, df = 18, p-value = 0.249
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.2833003 1.0253003
## sample estimates:
## mean in group Ctl mean in group Trt
##           5.032           4.661
```

Here, we test $H_0 : \mu_1 = \mu_2 = \mu$ vs. $H_1 : \mu_1 \neq \mu_2$.

```
ttest$statistic
```

```
##      t
## 1.19126
```

2.5.2 Regression

A simple linear model exploring the relationship between the plant weight and the group status can be fitted as follows:

```
lm.group.including.intercept <- lm(weight ~ 1 + group, data = Plant.Weight.Data)
lm.group.including.intercept

##
## Call:
## lm(formula = weight ~ 1 + group, data = Plant.Weight.Data)
```

```
##
## Coefficients:
## (Intercept)      groupTrt
##          5.032        -0.371

lm.group <- lm(weight ~ group, data = Plant.Weight.Data)
lm.group

##
## Call:
## lm(formula = weight ~ group, data = Plant.Weight.Data)
##
## Coefficients:
## (Intercept)      groupTrt
##          5.032        -0.371

confint(lm.group)

##                2.5 %    97.5 %
## (Intercept)  4.56934  5.4946602
## groupTrt    -1.02530  0.2833003
```

2.5.2.1 Interpretation

Note that the variable `group` is dummy coded. R generally chooses the first category as the reference category.

```
levels(as.factor(Plant.Weight.Data$group))
```

```
## [1] "Ctl" "Trt"
```

1. In this case, the intercept 5.032 tells us the predicted mean value for the plant weights for the control group (reference category of the group variable).
2. On the other hand, the slope is interpreted as the expected difference in the mean of the plant weights for that treatment group as compared to the control group. On average, weight is 0.371 units (lb?) lower in plants who are in the treatment condition compared to those in the control condition.

2.5.2.2 Summary of the regression fit

The complete summary of the results is as follows:

```
summary(lm.group)
```

```
##
## Call:
## lm(formula = weight ~ group, data = Plant.Weight.Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.0710 -0.4938  0.0685  0.2462  1.3690
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.0320     0.2202  22.850 9.55e-15 ***
## groupTrt       -0.3710     0.3114  -1.191   0.249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6964 on 18 degrees of freedom
## Multiple R-squared:  0.07308, Adjusted R-squared:  0.02158
## F-statistic: 1.419 on 1 and 18 DF,  p-value: 0.249
```

This is testing a different hypothesis (from the table): $H_0 : \alpha = 0$ vs. $H_1 : \alpha \neq 0$ (α being the intercept) and $H_0 : \beta = 0$ vs. $H_1 : \beta \neq 0$ (β being the slope). At the bottom of the `summary` output, the **F-statistic** tests $H_0 : \beta = 0$ vs. $H_1 : \beta \neq 0$. This is an overall, and could accomodate more slopes if the regression had more slopes. E.g., for 2 slopes, this would have tested $H_0 : \beta_1 = \beta_2 = 0$.

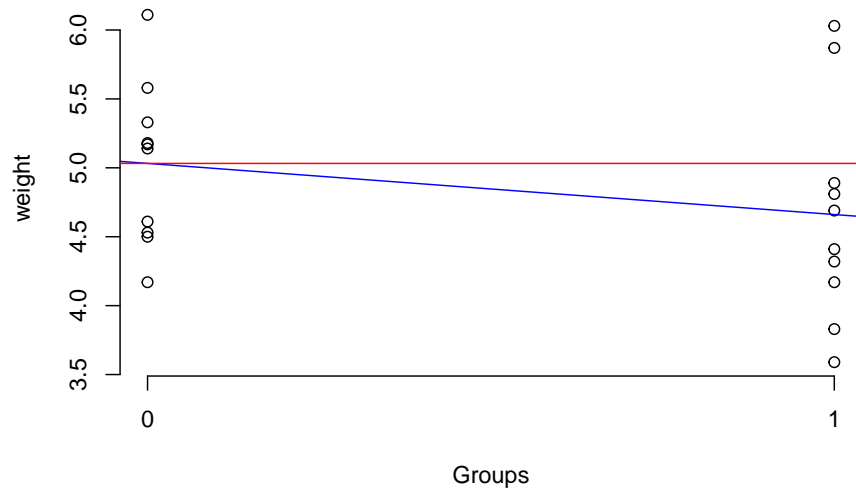
2.5.2.3 Regression plot

Let us visualize the scatter plot and the regression line:

```
Plant.Weight.Data$group.code <-
  ifelse(Plant.Weight.Data$group == "Trt", 1, 0)
Plant.Weight.Data$group.code

## [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1

lm.code <- lm(weight ~ group.code, data = Plant.Weight.Data)
plot(weight ~ group.code, data = Plant.Weight.Data,
      axes = FALSE, xlab = "Groups")
axis(1, 0:1, levels(Plant.Weight.Data$group))
axis(2)
abline(lm.code, col = "blue") # regression line
abline(h=coef(lm.code)[1], col = "red") # intercept
```



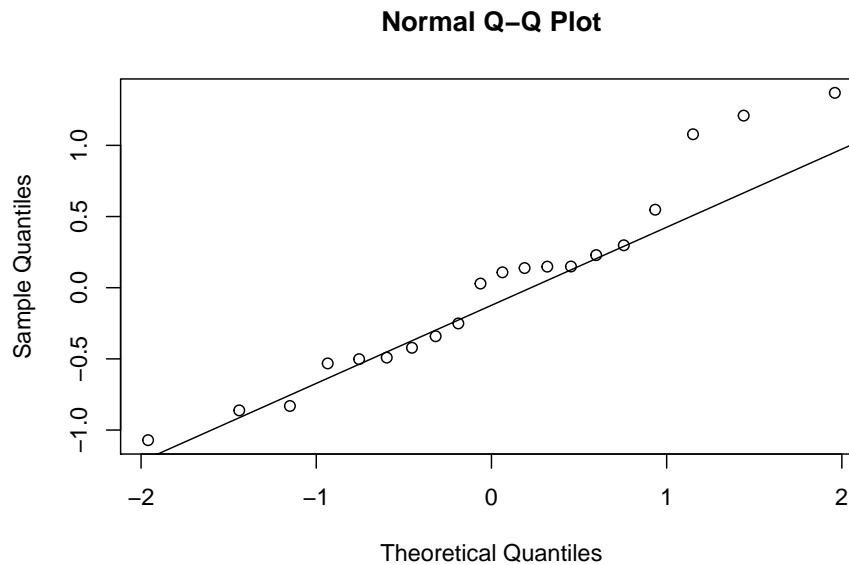
2.5.2.4 Assumption checking for the residuals

Checking normality of the residuals:

```
lm.residual <- residuals(lm.group)
shapiro.test(lm.residual)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  lm.residual
## W = 0.94744, p-value = 0.3299
```

```
qqnorm(lm.residual)
qqline(lm.residual)
```

2.5.2.5 Null model

A null model (with only intercept):

```
lm.null <- lm(weight ~ 1, data = Plant.Weight.Data) # Including just the intercept
summary(lm.null)
```

```
##
## Call:
## lm(formula = weight ~ 1, data = Plant.Weight.Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.2565 -0.4590 -0.0965  0.3710  1.2635
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.8465     0.1574   30.79  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.704 on 19 degrees of freedom
```

2.5.3 ANOVA

For testing for the significance of the group membership, we can compare the current model to the null model (is adding the variable `group` in the model useful?).

```
anova(lm.null, lm.group)
```

```
## Analysis of Variance Table
##
## Model 1: weight ~ 1
## Model 2: weight ~ group
##   Res.Df    RSS Df Sum of Sq    F Pr(>F)
## 1      19  9.4175
## 2      18  8.7292   1   0.68821 1.4191  0.249
```

Or, we could directly test $H_0 : \mu_1 = \mu_2 = \mu$ vs. $H_1 : \mu_1 \neq \mu_2$ under the homogeneity of variances assumption:

```
anova(lm.group)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1  0.6882  0.68820   1.4191  0.249
## Residuals 18  8.7292  0.48496
```

```
# Alternate ways to do the same
# car::Anova(lm.group, type="II")
aov.fit <- aov(lm.group)
summary(aov.fit)
```

```
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1   0.688   0.6882   1.419  0.249
## Residuals 18   8.729   0.4850
```

```
# Multiple pairwise-comparison:
# (compare with t-test; same p-value?)
TukeyHSD(aov.fit)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
```

```
##
## Fit: aov(formula = lm.group)
##
## $group
##      diff      lwr      upr      p adj
## Trt-Ctl -0.371 -1.0253 0.2833003 0.2490232
```

Checking normality of the residuals (not run; same as above):

```
# aov.residual <- residuals(aov.fit)
# shapiro.test(aov.residual)
# qqnorm(aov.residual)
# qqline(aov.residual)
```

ANOVA is basically a generalization of the two-sample t-test (verify that the calculated $F = t^2$):

```
ttest$statistic^2
```

```
##      t
## 1.419101
```

An alternative non-parametric version of this independent 2-sample test is as follows (a Kruskal-Wallis rank sum test):

```
# Assuming groups come from similar shaped populations:
kruskal.test(weight ~ group, data = Plant.Weight.Data)
```

```
##
## Kruskal-Wallis rank sum test
##
## data: weight by group
## Kruskal-Wallis chi-squared = 1.7513, df = 1, p-value = 0.1857
```

2.6 Verdict

2.6.1 Informal conclusion

With added nutrition, plant weights generally decrease (judging from the point estimate), but such trend could be due to sampling fluctuation (e.g., as the 95% confidence interval includes the null value of 0) and we can not confidently (not at least with 95% confidence) say that adding nutrition will cause plant weights to go down.

2.6.2 A word of caution

Note that, we are inherently trying to infer ‘causality’ out of a statistical analysis, even though our hypothesis is not about ‘cause’ explicitly. Unfortunately, correlation does not imply causation, and we need to know more about the subject-area and study-design before we make such inference or interpretation.

2.7 Exercises (Optional)

1. What is the difference between a regression analysis with a dummy coded predictor variable vs. an ANOVA?
2. Was multiple pairwise-comparison (**TukeyHSD**) necessary in the above example?
3. Which R package includes the **leveneTest** function? (hint: use **help.search()** function.)
4. Is ‘multicollinearity’ an issue in the above example?
5. In the current example, can we interpret the slope as follows: **the change in Y for a 1-unit change in X** where, Y being the outcome and X being the predictor? Why, or why not?

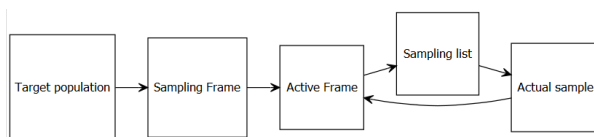
Chapter 3

Design-based Approach

Before discussing design-based approach, let us review some of concepts related to **sampling**.

3.1 Sampling

3.1.1 Steps of generalization



Example: Let us consider CCHS.

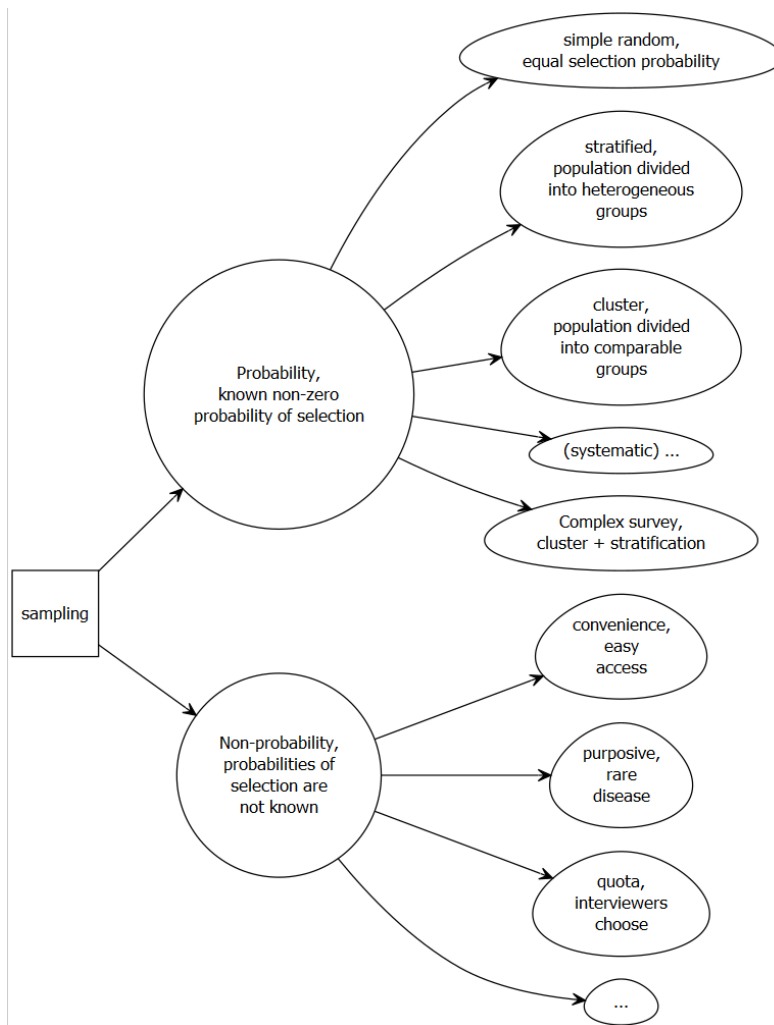
- Target population: You think about a **target population** in you PICOT.
 - Canadian population 12 years of age and over
- Sampling Frame: But all of your target population may not belong to a **sampling frame** compiled by a government.
 - Canadian population 12 years of age and over excluding about 3% population (e.g., aboriginal settlements, canadian Forces, institutionalized, foster care, 2 selected Quebec health regions)
- Active Frame: People that are still reachable
 - E.g., not dead or have not moved
- Sampling list

- Prepared from a specific sampling technique (SRS, stratified, cluster, complex)
- Actual sample: people that have responded
 - some don't respond

Note that, results from 'actual sample' are generalized to the 'active frame'. An inference from a sample may not really be generalizable to the target population (strictly speaking).

3.1.2 Types of sampling techniques

- Probability
- Non-probability

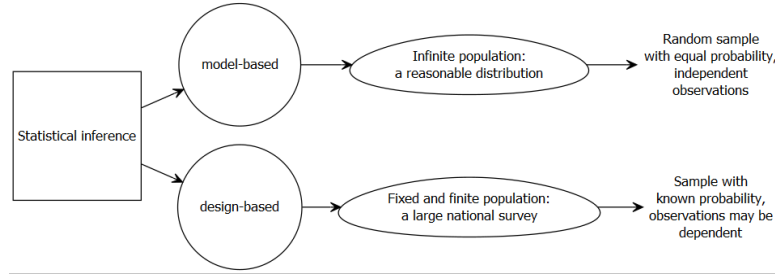


3.2 Statistical inference

3.2.1 Model-based

Most of the statistical techniques we have seen in our pre-requisite courses (SPPH 400, 500) generally assumed that we are dealing with a sample that was obtained from an infinite population. We usually assume that a random process can approximate such data generation process, and the data was collected by a simple random sampling or SRS (everyone has equal opportunity to be selected in the sample). All our conclusions are based on such assumptions. If we

are wrong in specifying correct distribution to approximate the data generating process, our subsequent inferences may not be valid anymore.



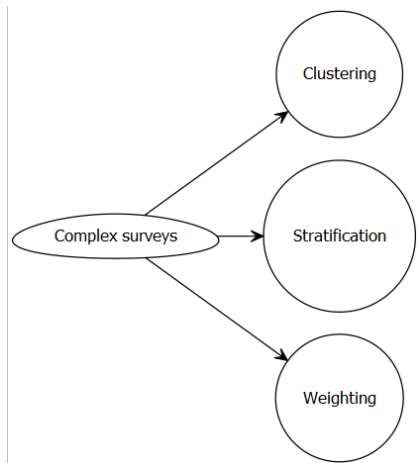
3.2.2 Design-based

Generally, when wide-scale surveys are designed, simple random sampling or SRS may not be feasible for various practical considerations. Maybe researchers and policy-makers want that a special but small sub-group subjects should be included in our sample (e.g., people suffering from a rare disease), but it is possible that by a SRS scheme, none of the subject from that small subgroup will be included. For convenience of sampling, and for controlling variance, researchers may have to make decisions regarding how the survey needs to collect sample. Researchers may resort to cluster or stratified sampling; or a mix of both (trade-off between cost and precision). Unfortunately, in these cases, equal probability of being selected in the sample is not there anymore. Lumley (2011) discussed the following properties for making design-based inference:

- properties needed to get valid estimates
 - non-zero probability ($P_i > 0$ for subject i) of being selected in the sample
 - every subject has a known probability (P_i) of being selected
- properties needed to achieve accuracy of those estimates
 - Every pair of subjects must have a non-zero probability ($P_{ij} > 0$ for subjects i and j) of being selected in the sample and
 - that probability (P_{ij}) must be known as well.

3.3 Complex surveys

3.3.1 Design features

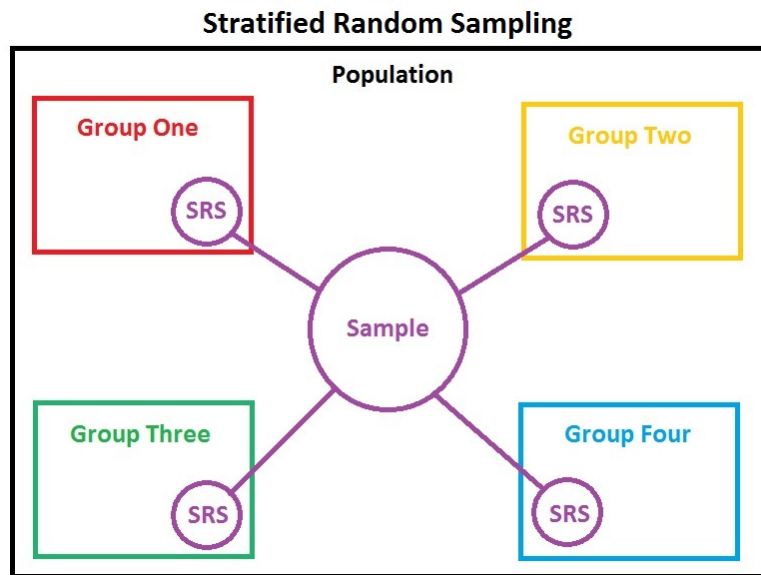


3.3.1.1 Stratification

Considering sub-groups that are sufficiently different from each other with respect to characteristics. Usual examples:

- different geographical location: Manitoba vs. Nunavut
- high income vs. low income
- gender

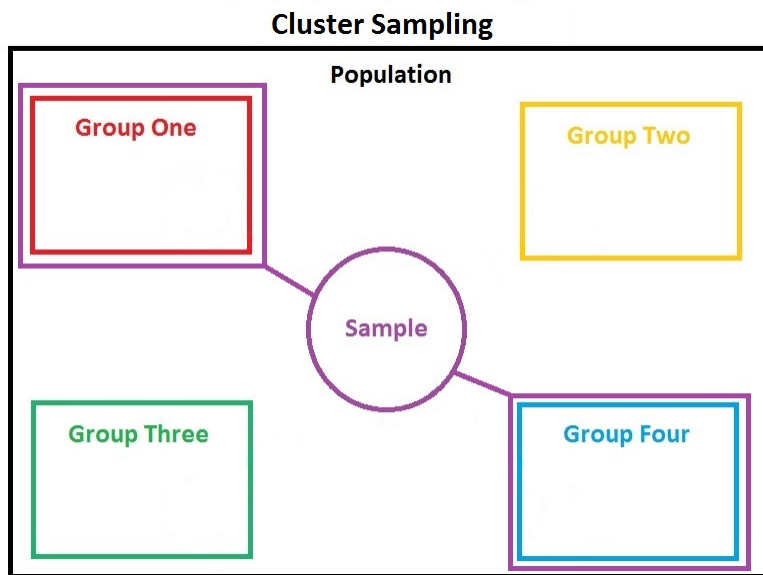
For each stratum (single unit), sampling is done separately. As we can select sample size from each stratum, we are able to control for variability of the estimates (SE) from each strata as well.



Source: [link](#)

3.3.1.2 Clustering

Clustering is done for convenience of data collection, generally. In a nationwide survey, researchers may choose to collect more samples from selected geographic locations. This is generally the case for cost considerations. In doing so, the surveyers don't have to travel too far, as they could essentially get many neighboring subjects at a much lower cost. An obvious consequence could be that the neighboring subjects may be more **correlated with each other** compared to subjects who are selected by randomness. This may cause the observations not being **independent** anymore.



Source: [link](#)

3.3.1.3 Weighting

Assume that, in a SRS, a subject is selected in a sample with a probability of $p_i = 0.04$. This means that person is representing $(1/p_i) = (1/0.04) = 25$ subjects in the population. We call this the **sampling weight** ($w_i = 25$). There are other types of weight:

- precision weight
- frequency weight

but we are not really interested about those in this course in general.

In a complex survey, where we have stratification and clustering, this weight is not as straightforward because, then, it is coming from an unequal probability sampling. As a consequence, not all subjects in the population will have the same probability (p_i) of being included in the sample, and the sampling weights (w_i) will vary as well (but the probability or weight is known for each subject).

3.3.2 Design effect

Compared to a SRS, all of the design features of a complex survey, such as, stratification, cluster sampling, and weighting generally influence the SEs of the

estimates. Survey researchers use a ratio called design effect, to account for the difference in SEs between a complex survey versus a SRS:

$$DE^2 = \frac{SE_{Complex.Survey}^2}{SE_{SRS}^2}.$$

3.4 Other ideas

- Oversampling

3.5 Further readings

Available via UBC library:

- Chapter 2 of Heeringa et al. (2017)
- Chapter 1 of Lumley (2011)
- Section 6.3 of Bilder and Loughin (2014)
- Chapter 12 of Vittinghoff et al. (2011)

3.6 Exercise

- Skim through the first chapter (from the further readings list). Should be easier to read most of it after this lecture.
- If any terminology remains unfamiliar, please discuss on Canvas.

Chapter 4

Potential Data Sources

4.1 Survey data with features

- Canadian Community Health Survey - Annual Component CCHS
 - Download link UBC library
- National Health and Nutrition Examination Survey NHANES
 - R packages to download data: nhanesA, RNHANES
- National Longitudinal Study of Adolescent to Adult Health [Add Health], 1994-2008 ICPSR 21600
- European Social Survey ESS
 - R package to download data: essurvey
- Behavioral Risk Factor Surveillance System BRFSS
- Bureau of Economic Analysis BEA
- US National Vital Statistics System NVSS

4.2 Others

- Vanderbilt Biostatistics Datasets link
- World Bank Open Data WBOD
 - R packages to download data: wbstats, WDI

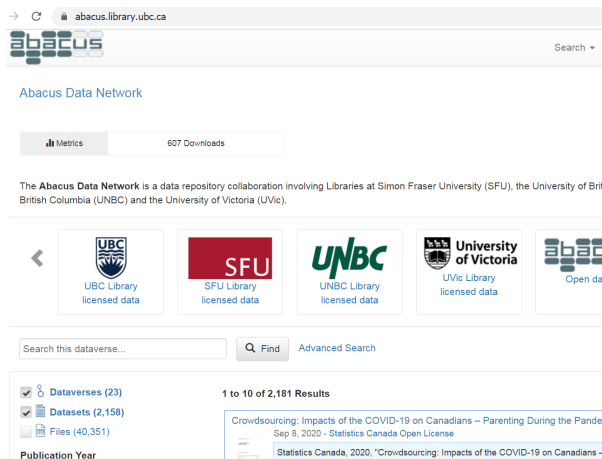
Chapter 5

Importing CCHS to R

This is a short instruction document of how to get CCHS dataset from the UBC library site to your RStudio environment. Once we bring the dataset into RStudio, the next step is to think about creating analytic dataset.

5.1 Downloading CCHS data from UBC

- **Step 1:** Go to [dvn.library.ubc.ca](https://abacus.library.ubc.ca), and press ‘log-in’



- **Step 2:** Select ‘UBC’ from the dropdown menu

Search ▾ About User Guide Support **Log In**

Log In

Your Institution

actions

Simon Fraser University

University of British Columbia

University of Northern British Columbia

Admin login

- **Step 3:** Enter your CWL or UBC library authentication information

EZproxy Login

CWL Library Login

To log into EZproxy using the UBC Campus-Wide Login (CWL) fa

Standard Library Login

To log into EZproxy using your UBC/Library card barcode and PIN


Barcode:


PIN:


- **Step 4:** Once you log-in, search the term 'cchs' in the search-box

abacus

<

 UBC Library
licensed data

 SFU Library
licensed data

☒  **Dataverses (23)**

1 to 10 of 2,181 Re

- **Step 5:** For illustrative purposes, let us work with the Cycle 3.1 of the CCHS dataset from the list of results. In that case, type ‘cchs 3.1’

Abacus Data Network

Metrics 607 Downloads

cchs 3.1 [Advanced Search](#)

☒ Datasets (0)
☒ Datasets (42)
☒ Files (133)

Publication Year
2020 (175)

Producer Name
Statistics Canada (74)

1 to 10 of 175 Results

Canadian Community Health Survey, Cycle 3.1, 2005 [2006]
 Aug 29, 2020 - Statistics Canada Open License

Statistics Canada, 2009, "Canadian Community Health Survey, C
 Abacus Data Network, V1

... for CCHS Cycle 3.1 between January 2005 and December 2005. The CCI
 Series Name: CCHS

- **Step 6:** CCHS Cycle 3.1 information

Statistics Canada Open License (Public)

Abacus Data Network > Statistics Canada Open License > Canadian Community Health Survey, Cycle 3.1, 2005 [2006]

[Share](#) [Link](#)

Canadian Community Health Survey, Cycle 3.1, 2005 [2006]

Version 1.0

Statistics Canada, 2009, "Canadian Community Health Survey, Cycle 3.1, 2005 [2006]", <https://hdl.handle.net/11272/1AN2/GBVYDV>,
 Abacus Data Network, V1

[Cite Dataset](#) [Learn about Data Citation Standards](#)

Description

The Canadian Community Health Survey (CCHS) is a cross-sectional survey that collects information related to health status, health care utilization and health determinants for the Canadian population. The CCHS operates on a two-year collection cycle. The first year of the survey cycle "1" is a large sample, general population health survey designed to provide reliable estimates at the health region level. The second year of the survey cycle "2" has a smaller sample and is designed to provide provincial level results on specific focused health topics. This Public Use Microdata File (PUMF) contains data collected for CCHS Cycle 3.1 between January 2005 and December 2005. The CCHS Cycle 3.1 collects responses from persons aged 12 or older, living in private occupied dwellings in 122 health regions covering all provinces and territories. Excluded from the sampling frame are individuals living on Indian Reserves and on Crown Lands, institutional residents, full-time members of the Canadian Forces, and residents of certain remote regions. The CCHS covers approximately 98% of the Canadian population aged 12 and over.

Keyword Health

[Files](#) [Metadata](#) [Terms](#) [Versions](#)

Change View [Table](#) [Tree](#)

Search this dataset

Filter by
 File Type: All Access: All File Tag: All

1 to 10 of 70 Files

[Download](#)

CCHS 2015-Enquête
 Documentation
 Adobe PDF - 352.0 KB - Aug 20, 2020 - 1 Download
 MD5: 776e114ae48c3269592691c1c044

[Explore](#) [Download](#)

- **Step 7:** Choose the ‘Data: CD’ from the menu

Keyword

Health

Files

Metadata

Terms

Versions

Change View

Table

Tree

Search this dataset...

Find

Filter by

File Type: All

Access: All

File Tag: All

1 to 10 of 70 Files

	CCHS 2015-Errata Documentation/ Adobe PDF - 592.0 M MD5: 77ae13f4ae4 Errata for all CCHS <div>Documentation</div>
	CCHS 2000-2001 Documentation/ Adobe PDF - 1.3 M MD5: 272bebee537 Errata for all CCHS <div>Documentation</div>
	cchs_cycle3-1C1 Data CD/ Unknown - 159.0 M MD5: d444be61715 Data: CD - Comple <div>Data: CD</div>

All

Command Files: SAS (17)

Command Files: SPSS (17)

Documentation: General (6)

Documentation (5)

Documentation: HS (5)

Documentation: HSS1 (5)

Documentation: HSS2 (5)

Documentation: HSS3 (5)

Data: CD (1)

Data: HS (1)

Data: HSS1 (1)

Data: HSS2 (1)

Data: HSS3 (1)

- **Step 8:** Download the entire data (about 159 MB) as a zip file

Keyword Health


Files Metadata Terms Versions

Change View

Filter by
File Type: All Access: All File Tag: **Data: CD**

1 file is currently selected. Select all 1 files in this dataset. Clear selection.

1 File

<input checked="" type="checkbox"/>	 <p>cdm_cyclo3-ICD zip Data CD Uploaded - 11:51 AM - Aug 25, 2020 - 13 Downloads URL: dataset/775b042043a24690704e Data CD - Complete CCHS Cycle 3 1 CD (zip) Download</p>	<input type="button" value="Download"/>
-------------------------------------	--	---

- **Step 9:** Accept the ‘terms of use’

Dataset Terms

Please confirm and/or complete the information needed below in order to continue.

Terms of Use

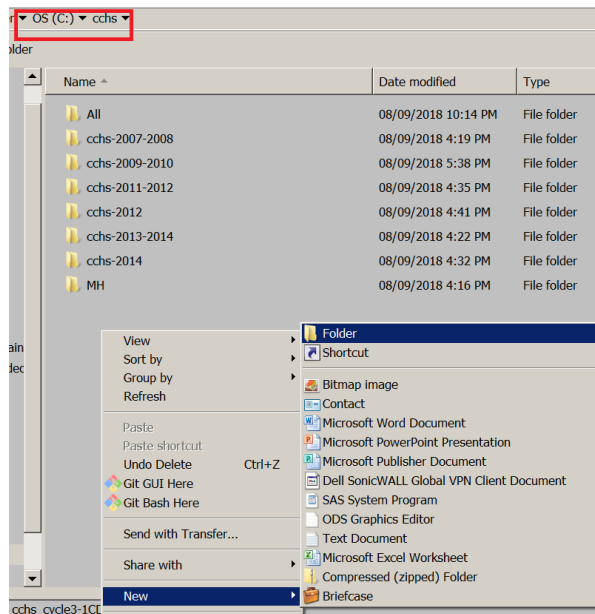
Statistics Canada Open Licence

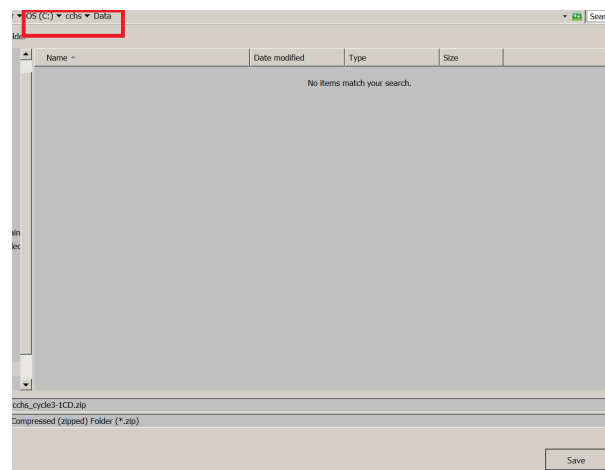
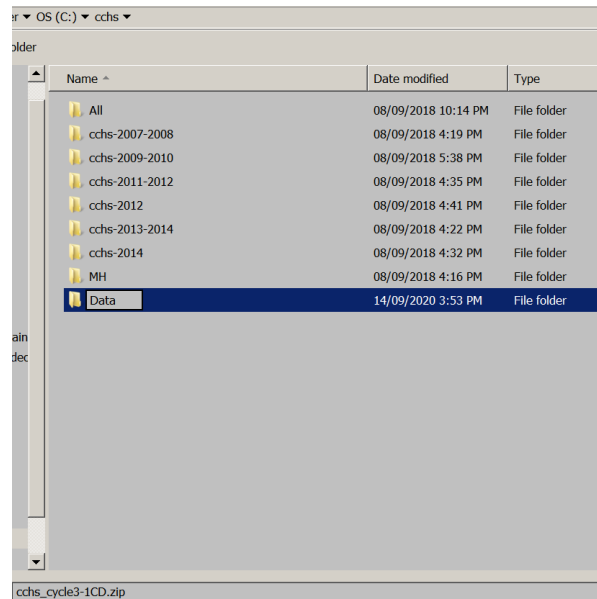
This licence is issued on behalf of Her Majesty the Queen in Right of Canada represented by the Minister for Statistics Canada ("Statistics Canada") to the individual or a legal entity that you are authorized to represent).

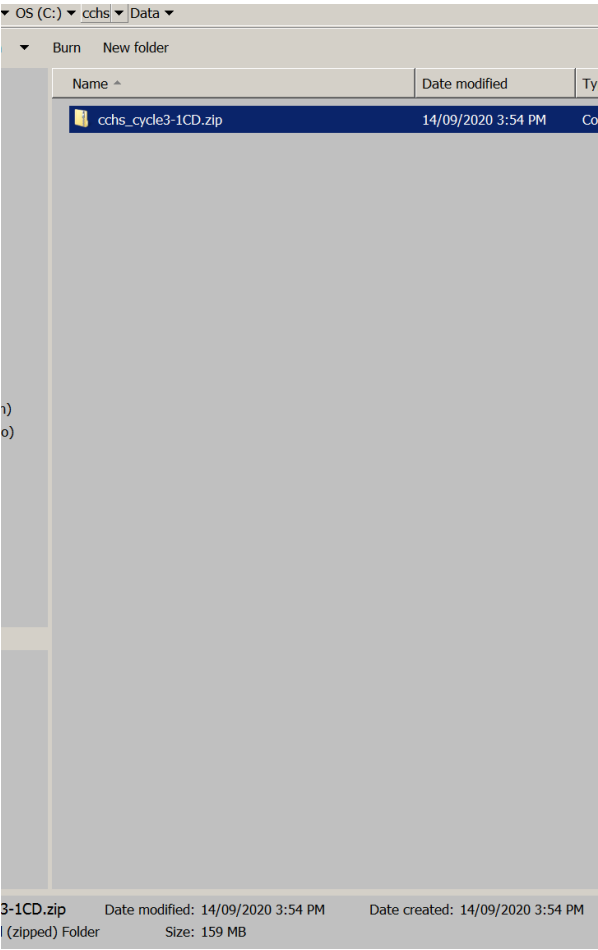
The following are terms governing your use of the Information. Your use of the Information indicates your understanding and acceptance of the terms. If you do not agree to these terms, you may not use the Information.

Accept Cancel

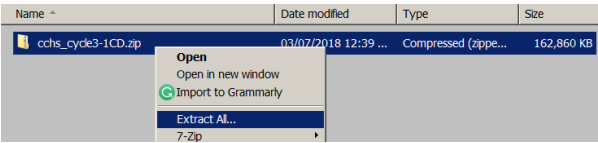
- **Step 10:** Select a directory to download the zip file. The path of the download directory is important (we need to use this **path** exactly later). For example, below we are in "C:\CCHS\" folder, but we will create a "Data" folder there, so that the download path is "C:\CCHS\Data\".



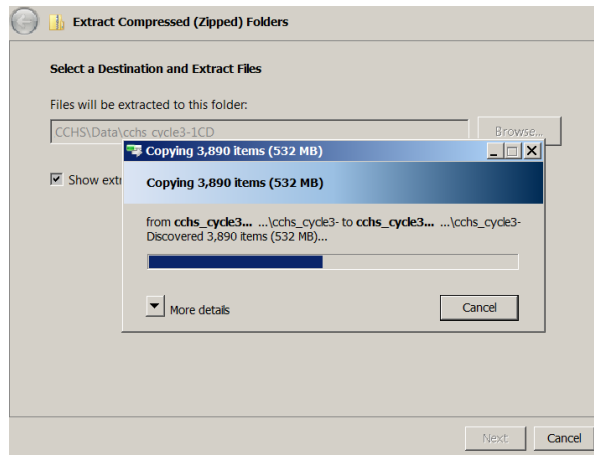




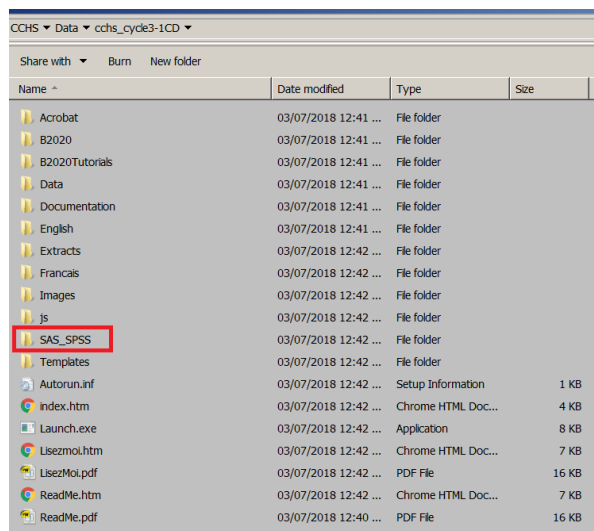
- **Step 11:** Extract the zip file



- **Step 12:** Be patient with the extraction



- **Step 13:** Once extraction is complete, take a look at the folders inside. You will see that there is a folder named 'SAS_SPSS'



5.2 Reading and Formatting the data

5.2.1 Option 1: Processing data using SAS

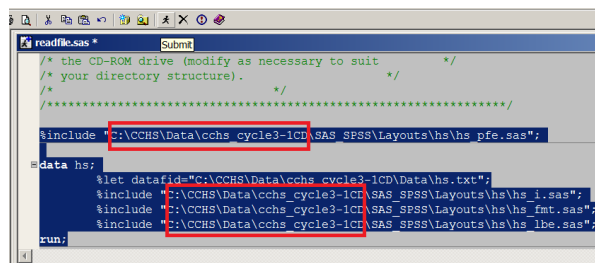
SAS is a commercial software. You may be able to get access to educational version. In case you don't have access to it, later we outline how to use free packages to read these datasets.

- **Step 1:** Inside that 'SAS_SPSS' folder, find the file *hs_pfe.sas*. It is a long file, but we are going to work on part of it. First thing we want to do is to change all the directory names to where you have unzipped the downloaded file (for example, here the zip file was extracted to C:/CCHS/Data/cchs_cycle3-1CD/). We only need the first part of the code (as shown below; only related to data 'hs'). Delete the rest of the codes for now. The resulting code should look like this:

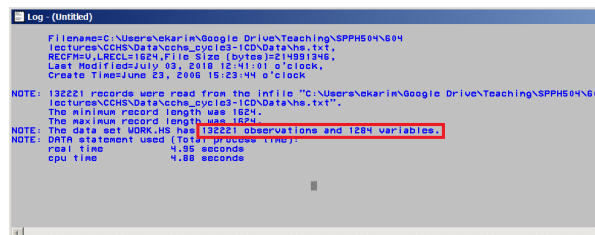
```
%include "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_pfe.sas";

data hs;
    %let datafid="C:\CCHS\Data\cchs_cycle3-1CD\Data\hs.txt";
    %include "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_i.sas";
    %include "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_fmt.sas";
    %include "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_lbe.sas";
run;
```

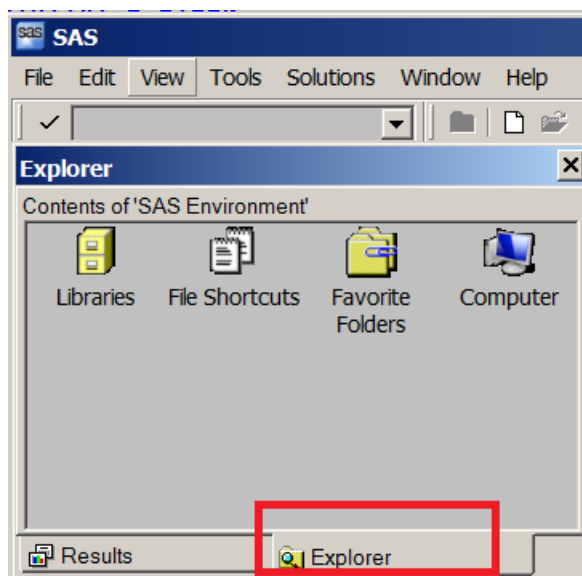
Once the modifications are done, submit the codes in SAS. Note that, the name of the data is 'hs'.



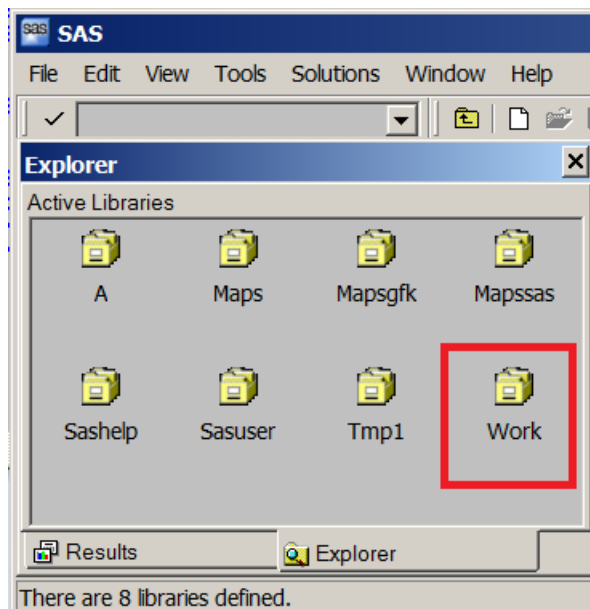
- **Step 2:** Once you submit the code, you can check the log window in SAS to see how the code submission went. It should tell you how many observations and variables were read.



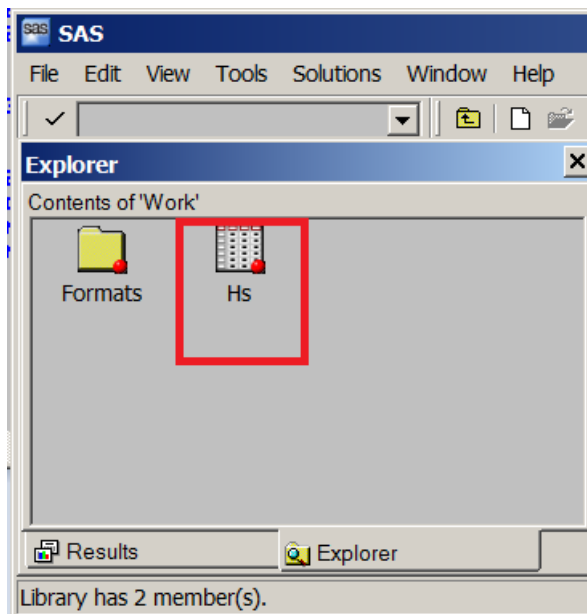
- **Step 3:** If you one to view the dataset, you can go to 'Explorer' window within SAS.



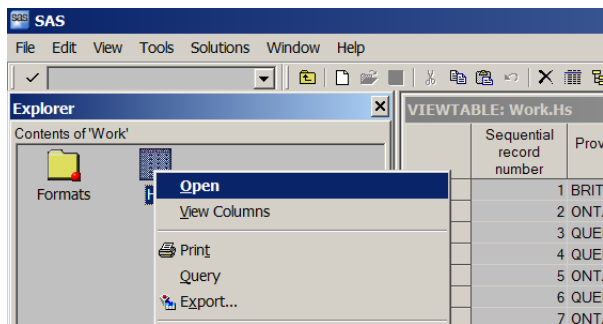
- **Step 4:** Generally, if you haven't specified where to load the files, SAS will by default save the data into a library called 'Work'



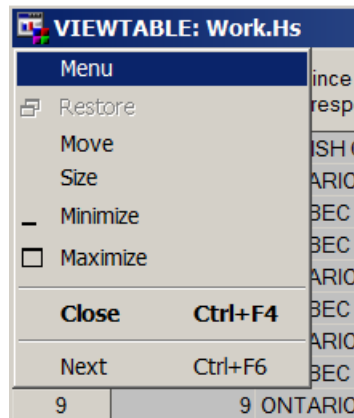
- **Step 5:** Open that folder, and you will be able to find the dataset 'Hs'.



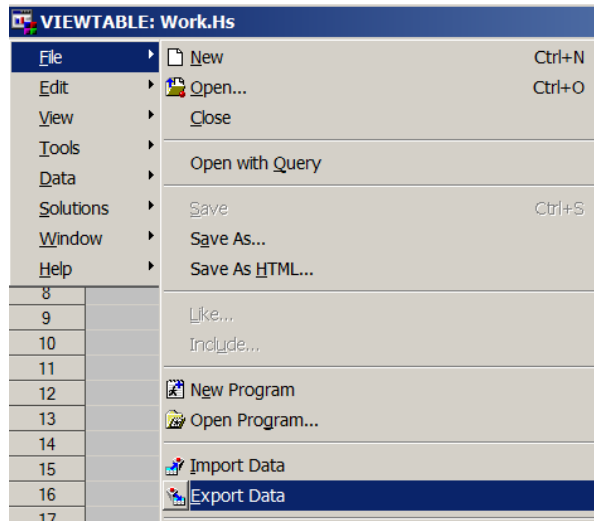
- **Step 6:** Right click on the data, and click 'open' to view the datafile.



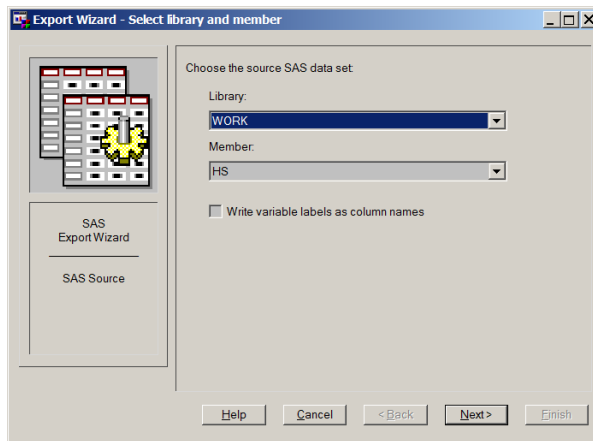
- **Step 7:** To export the data into a CSV format data (so that we can read this data into other software packages), click 'Menu'.



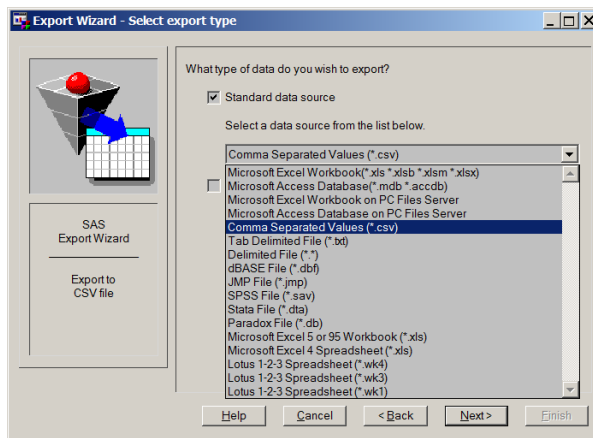
- **Step 8:** then press 'Export Data'.



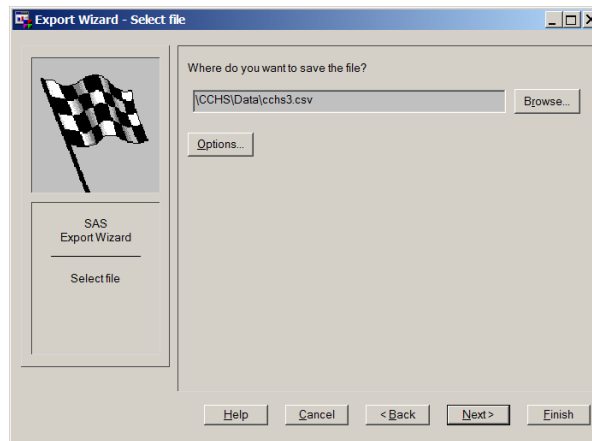
- **Step 9:** choose the library and the data.



- **Step 10:** choose the format in which you may want to save the existing data.



- **Step 11:** also specify where you want to save the csv file and the name of that file (e.g., cchs3.csv).



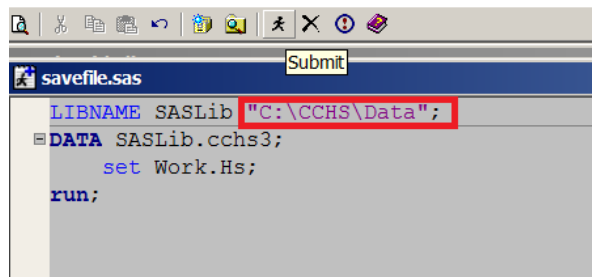
- **Step 12:** go to that directory to see the file cchs3.csv

Name	Date modified	Type	Size
cchs_cycle3-1CD	03/07/2018 12:42 ...	File folder	
cchs3.csv	03/07/2018 1:06 PM	Microsoft Excel Co...	2,103,015 ...
readfile.sas	03/07/2018 1:18 PM	SAS System Program	2 KB
readfile.sps	03/07/2018 1:33 PM	SPS File	2 KB

- **Step 13:** If you want to save the file in SAS format, you can do so by writing the following sas code into the 'Editor' window. Here we are saving the data Hs within the Work library in to a data called cchs3 within the SASLib library. Note that, the directory name has to be where you want to save the output file.

```
LIBNAME SASLib "C:\CCHS\Data";
DATA SASLib.cchs3;
    set Work.Hs;
run;
```

Submit these codes into SAS:



- **Step 13:** go to that directory to see the file cchs3.sas7dbat

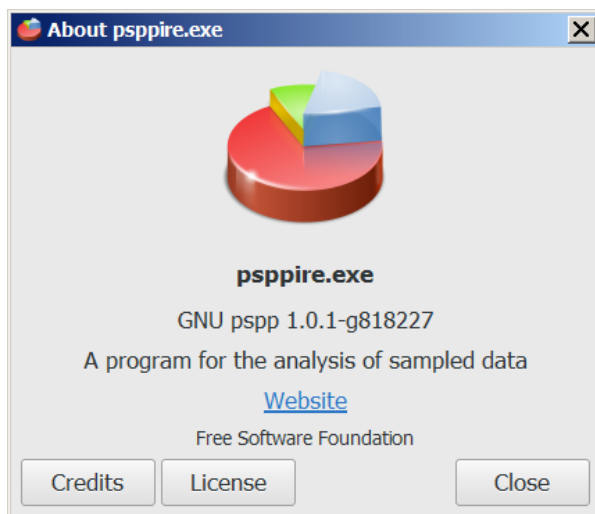
Name	Date modified	Type	Size
cchs_cycle3-1CD	03/07/2018 12:42 ...	File folder	
cchs3.csv	03/07/2018 1:06 PM	Microsoft Excel Co...	2,103,015 ...
cchs3.sas7bdat	03/07/2018 2:11 PM	SAS Data Set	1,333,332 ...
readfile.sas	03/07/2018 1:18 PM	SAS System Program	2 KB
readfile.sps	03/07/2018 1:33 PM	SPS File	2 KB

5.2.2 Option 2: Processing data using PSPP (Free)

PSPP is a free package; alternative to commercial software SPSS. We can use the same SPSS codes to read the datafile into PSPP, and save.

- **Step 1:** Get the free PSPP software from the website: www.gnu.org/software/pspp/

PSPP is available for GNU/Hurd, GNU/Linux, Darwin (Mac OS X), OpenBSD, NetBSD, FreeBSD, and Windows



For windows, download appropriate version.

pspp.awardspace.info ☆

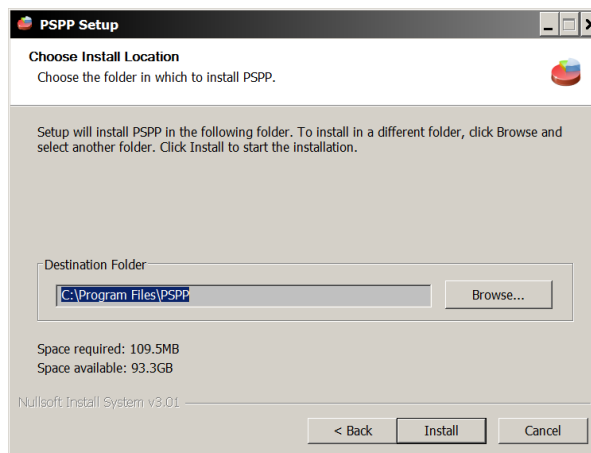
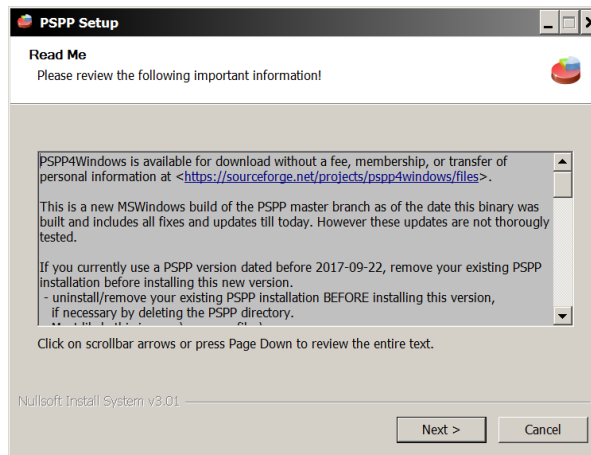
Highlights of the current PSPP-for-MSWindows setup			
PSPP info:		Package info:	
Current version:	Build from daily snapshot source, not fully tested	MSWindows version:	MSWindows7 and newer
Questions/Suggestions:	pspp-users@gnu.org	Package Size:	40 Mb
Information about PSPP:	https://www.gnu.org/software/pspp	Size on disk:	80 Mb
PSPP Manual:	PDF or HTML (will be installed on your PC by the installer package)	Technical:	MinGW based Cross-comp on openSUSE leap 42.3
View in latest build:	NEWS (open downloaded file with notepad)		

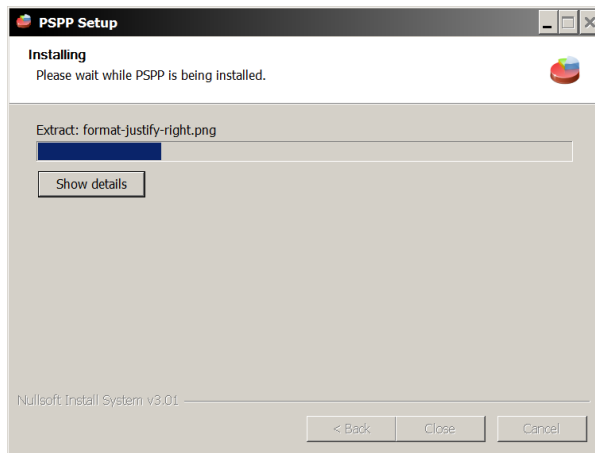
Downloads:			
Installer for 32bits version Will work on 32 and 64bits MSWindows	Installer for 64bits version Works only on 64bits MSWindows	Source version close to a released version	Build generation
PSPP_2018-11-09_daily_32bits	PSPP_2018-11-09_daily_64bits	Yes 1.2.0-g0fb84db	42.2.3
PSPP_2017-09-09_daily_32bits	PSPP_2017-09-09_daily_64bits	Yes 1.0.1-g818227	42.2.3
PSPP_2017-07-30_daily_32bits	PSPP_2017-07-30_daily_64bits	Yes 0.10.5-pre3-g9a68ff	42.2.3
PSPP_2016-09-27_daily_32bits	PSPP_2016-09-27_daily_64bits	No 0.10.4-g50f7b7	42.1.4

Download the file

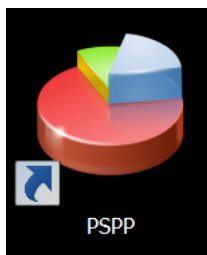


Install

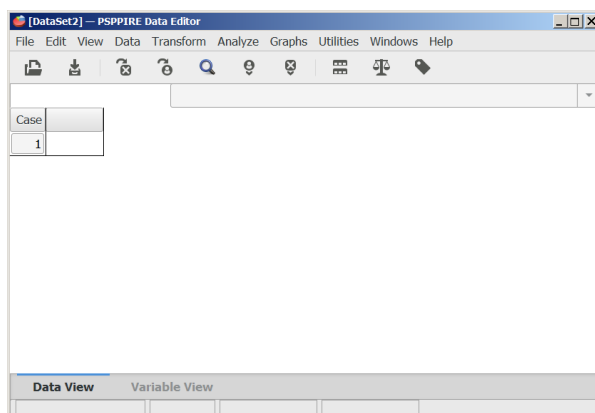




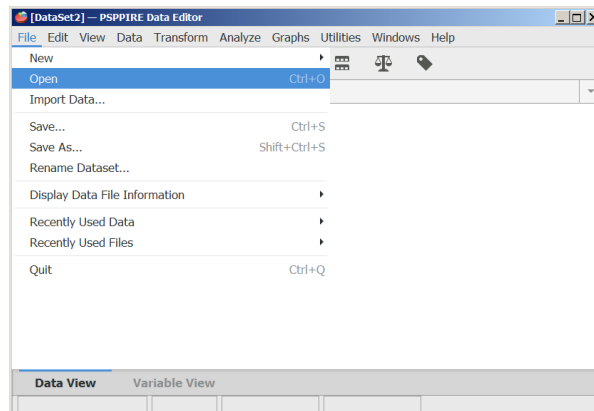
Click the icon shortcut after installing



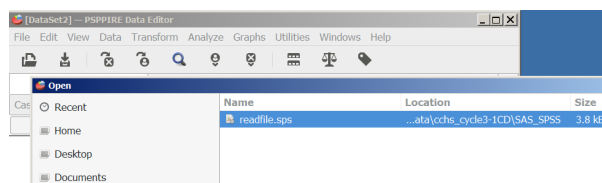
- **Step 2:** Open PSPP



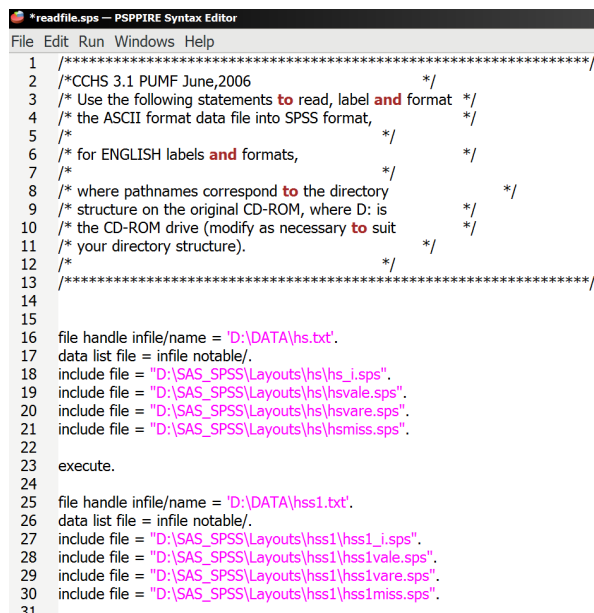
- **Step 3:** Go to 'file' menu and click 'open'



- **Step 4:** Specify the *readfile.sps* file from the 'SAS_SPSS' folder.



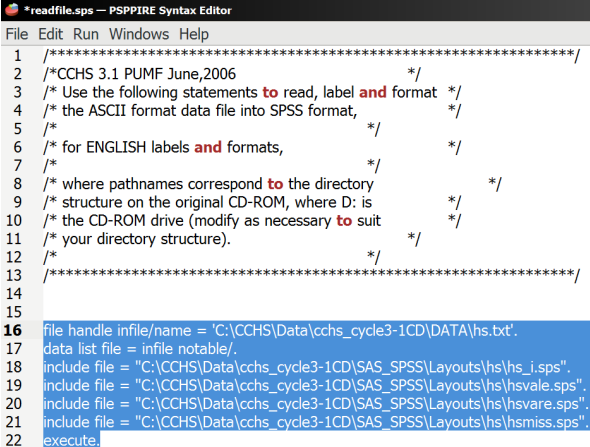
You will see the following file:



- **Step 5:** Similar to before, change the directories as appropriate. Get rid

of the extra lines of codes. Resulting codes are as follows (you can copy and replace the code in the file with the following codes):

```
file handle infile/name = 'C:\CCHS\Data\cchs_cycle3-1CD\DATA\hs.txt'.
data list file = infile notable/.
include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_i.sps".
include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsvale.sps".
include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsvare.sps".
include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsmisss.sps".
execute.
```



```
*readfile.sps — PSPPIRE Syntax Editor
File Edit Run Windows Help
1  /*****
2  /*CCHS 3.1 PUMF June,2006          */
3  /* Use the following statements to read, label and format */
4  /* the ASCII format data file into SPSS format,          */
5  /* */
6  /* for ENGLISH labels and formats,                        */
7  /* */
8  /* where pathnames correspond to the directory            */
9  /* structure on the original CD-ROM, where D: is          */
10 /* the CD-ROM drive (modify as necessary to suit         */
11 /* your directory structure).                             */
12 /* */
13 /*****/
14
15
16 file handle infile/name = 'C:\CCHS\Data\cchs_cycle3-1CD\DATA\hs.txt'.
17 data list file = infile notable/.
18 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_i.sps".
19 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsvale.sps".
20 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsvare.sps".
21 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hsmisss.sps".
22 execute.
```

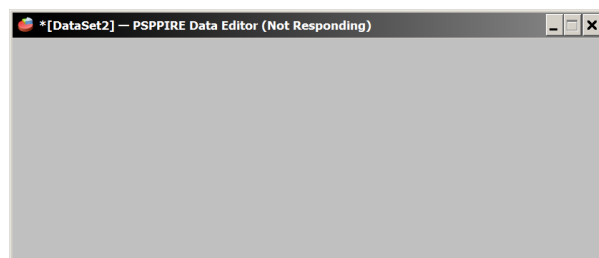
- **Step 6:** Run the codes.

```

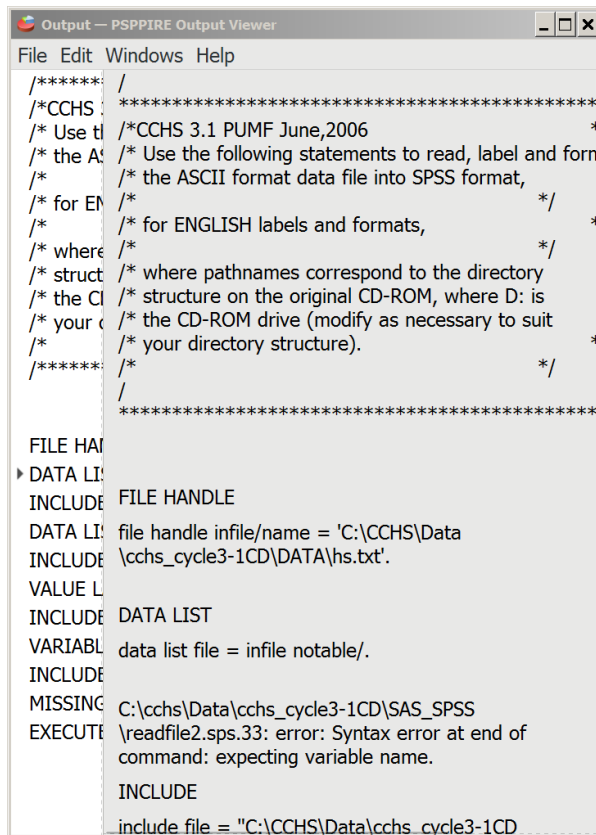
1  /*** All *****/
2  /*CC Selection :s to read, label and format */
3  /* U: Current Line Ctrl+R to SPSS format, */
4  /* th To End */
5  /* for evolution labels and formats, */
6  /* */
7  /* where pathnames correspond to the directory */
8  /* structure on the original CD-ROM, where D: is */
9  /* the CD-ROM drive (modify as necessary to suit */
10 /* your directory structure). */
11 /* */
12 /* *****/
13
14
15
16 file handle infile/name = 'C:\CCHS\Data\cchs_cycle3-1CD\DATA\hs.txt'.
17 data list file = infile notable/.
18 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_i.sps".
19 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_vale.sps".
20 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_vare.sps".
21 include file = "C:\CCHS\Data\cchs_cycle3-1CD\SAS_SPSS\Layouts\hs\hs_miss.sps".
22
23 execute.

```

- **Step 7:** This is a large data, and will take some time to load the data into the PSPP data editor. **Be patient.**



Once loading is complete, it will show the 'output' and 'data view'.



```

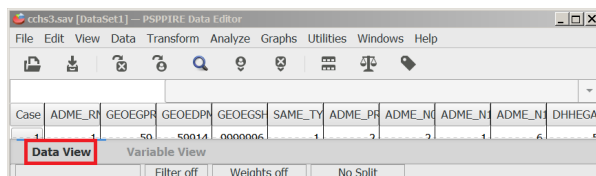
Output — PSPPIRE Output Viewer
File Edit Windows Help

/*****
/*CCHS
/* Use the following statements to read, label and format
/* the ASCII format data file into SPSS format,
/* for ENGLISH labels and formats,
/* where pathnames correspond to the directory
/* structure on the original CD-ROM, where D: is
/* the CD-ROM drive (modify as necessary to suit
/* your directory structure).
*****/

FILE HANDLE
DATA LIST
INCLUDE FILE HANDLE
DATA LIST file handle infile/name = 'C:\CCHS\Data
INCLUDE \cchs_cycle3-1CD\DATA\hs.txt'.
VALUE LABELS
INCLUDE DATA LIST
VARIABLES data list file = infile notable/.
INCLUDE MISSING
EXECUTE C:\cchs\Data\cchs_cycle3-1CD\SAS_SPSS
\readfile2.sps.33: error: Syntax error at end of
command: expecting variable name.

INCLUDE
include file = "C:\CCHS\Data\cchs_cycle3-1CD

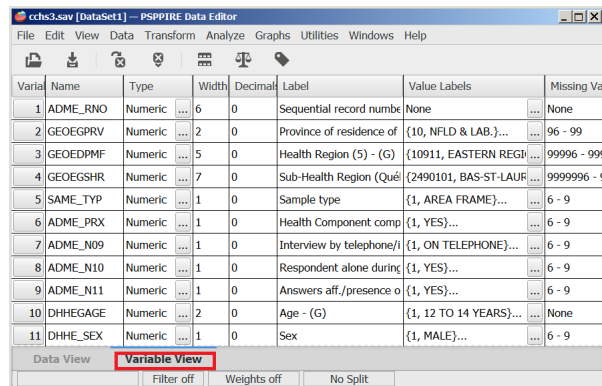
```



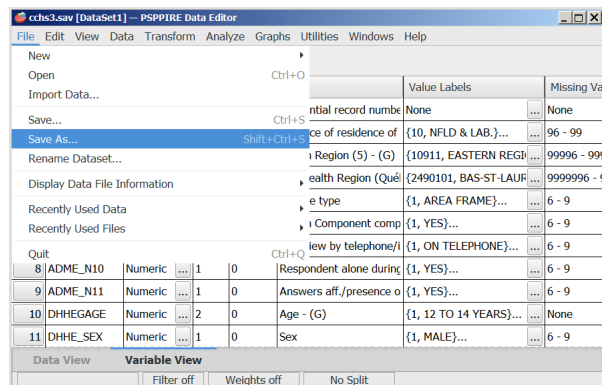
Case	ADME_RN	GEOEGPR	GEOEGSH	SAME_TY	ADME_PR	ADME_N	ADME_N	ADME_N	DHHEGAC
1	50	50014	0000006	1	2	2	1	6	5

Note that, you will get error message, if your files were not in the correct path. In our example, the path was "C:\CCHS\Data\" for the zip file content (see the previous steps).

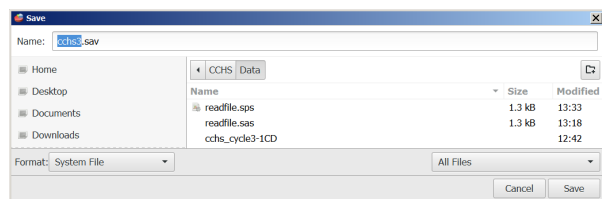
- **Step 7:** You can also check the 'variable view'.



- **Step 8:** Save the data by clicking 'File' and then 'save as ...'



- **Step 9:** Specify the name of the datafile and the location / folder to save the data file.



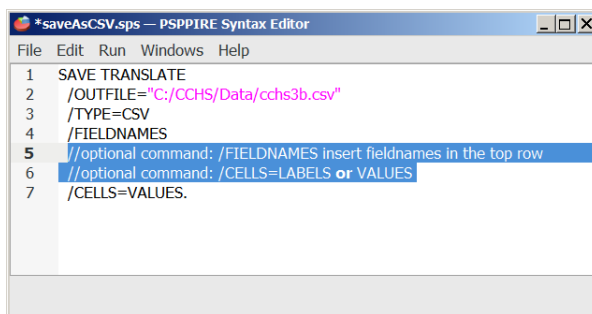
- **Step 10:** See the SAV file saved in the directory.

Name	Date modified	Type	Size
cchs_cycle3-1CD	03/07/2018 12:42 ...	File folder	
cchs3.csv	03/07/2018 1:06 PM	Microsoft Excel Co...	2,103,015 ...
cchs3.sav	03/07/2018 1:35 PM	SAV File	252,073 KB
readfile.sas	03/07/2018 1:18 PM	SAS System Program	2 KB
readfile.sps	03/07/2018 1:33 PM	SPS File	2 KB

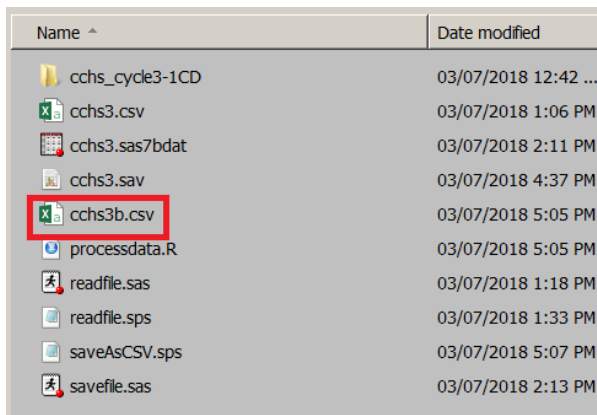
- **Step 11:** To save CSV format data, use the following syntax.

```
SAVE TRANSLATE
/OUTFILE="C:/CCHS/Data/cchs3b.csv"
/TYPE=CSV
/FIELDNAMES
/CELLS=VALUES.
```

Note that, for categorical data, you can either save values or labels. For our purpose, we prefer values, and hence saved with values here.



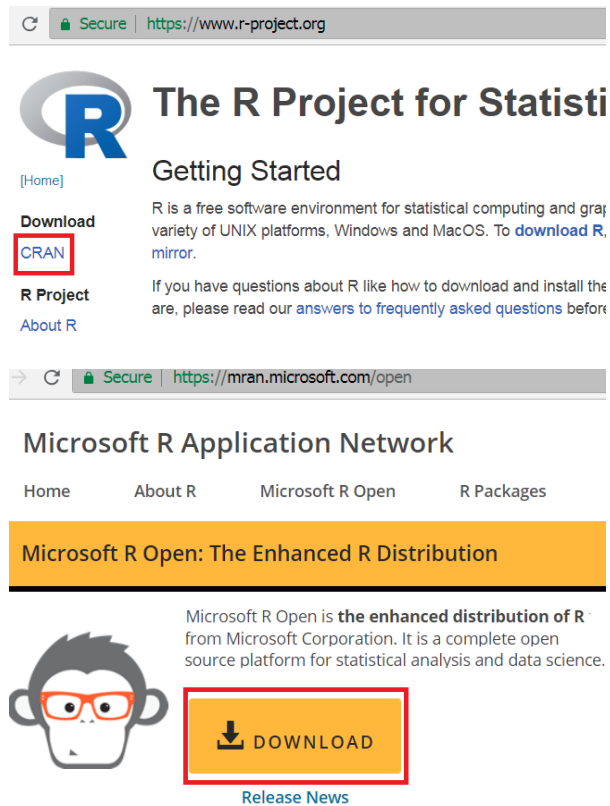
- **Step 12:** See the CSV file saved in the directory extracted from PSPP.



5.3 Processing data in R

5.3.1 Download software

- **Step 1:** Download either 'R' from CRAN www.r-project.org or 'R open' from Microsoft mran.microsoft.com/open



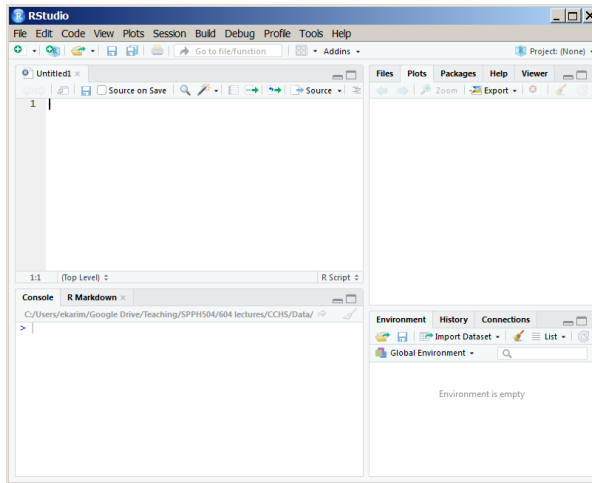
The screenshot shows the R Project for Statistical Computing website. The browser address bar displays "Secure | https://www.r-project.org". The page features the R logo and the title "The R Project for Statistical Computing". Below the logo, there is a "Getting Started" section. Under the "Download" heading, the "CRAN" link is highlighted with a red box. The text describes R as a free software environment for statistical computing and graphics, available on various UNIX platforms, Windows, and MacOS. It also mentions that users should read frequently asked questions before downloading. Below this, there is a "Microsoft R Application Network" section. The browser address bar shows "Secure | https://mran.microsoft.com/open". The page title is "Microsoft R Application Network". The navigation bar includes links for "Home", "About R", "Microsoft R Open", and "R Packages". A yellow banner reads "Microsoft R Open: The Enhanced R Distribution". Below the banner, there is a cartoon monkey wearing glasses. To the right of the monkey, the text states: "Microsoft R Open is the enhanced distribution of R from Microsoft Corporation. It is a complete open source platform for statistical analysis and data science." A yellow button with a download icon and the word "DOWNLOAD" is highlighted with a red box. Below the button is a link for "Release News".

- **Step 2:** Download RStudio from www.rstudio.com/



The screenshot shows the RStudio website. The browser address bar displays "Secure | https://www.rstudio.com". The page features the RStudio logo and the title "RStudio". Below the logo, there is a large blue banner with the text "RStudio" in large white letters. Underneath the banner, it says "Open source and enterprise-ready professional software for R".

- **Step 3:** Open RStudio



5.3.2 Import, export and load data into R

- **Step 1:** Set working directory

```
setwd("C:/CCHS/Data/") # or something appropriate
```

- **Step 2:** Read the dataset created from PSPP with cell values. We can also do a small check to see if the cell values are visible. For example, we choose a variable 'CCCE_05A', and tabulate it.

```
Hs <- read.csv("cchs3b.csv", header = TRUE)
table(Hs$CCCE_05A)
```

```
Console R Markdown x
C:/CCHS/Data/
> table(Hs$CCCE_05A)
 1      2      3      4      6      7      8      9
5098 14141 2096 2236 103781 4609  41    219
> |
```

- **Step 3:** Save the RData file from R into a folder SurveyData:

```
save(Hs, file = "SurveyData/cchs3.RData")
```

- **Step 4:** See the RData file saved in the directory extracted from R.

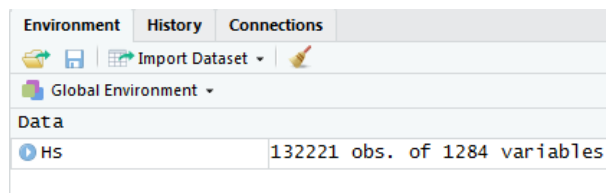
Name	Date modified
cchs_cycle3-1CD	03/07/2018 12:42 ..
processdata.R	03/07/2018 5:25 PM
cchs3.RData	03/07/2018 5:23 PM
.Rhistory	03/07/2018 5:31 PM
readfile.sas	03/07/2018 1:18 PM
savefile.sas	03/07/2018 2:13 PM
CCHS3.1 - Shortcut	03/07/2018 5:24 PM
readfile.sps	03/07/2018 1:33 PM
saveAsCSV.sps	03/07/2018 5:07 PM

- **Step 5:** Close R / RStudio and restart it. Environment window within RStudio should be empty.



- **Step 6:** Load the saved RData into R. Environment window within RStudio should have 'Hs' dataset.

```
load("SurveyData/cchs3.RData")
```



Chapter 6

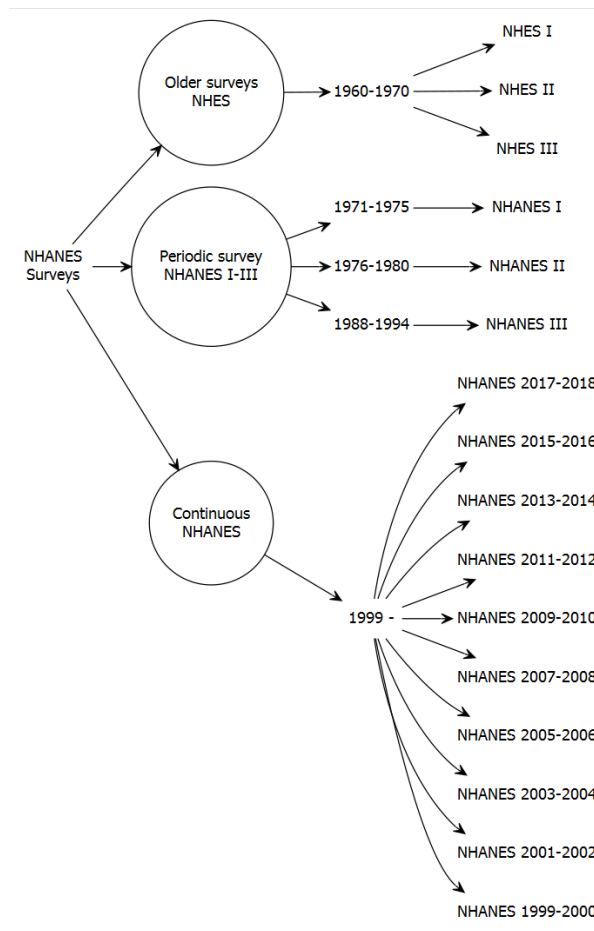
Demystifying NHANES

6.1 Overview

National Center for Health Statistics (NCHS) has been conducting surveys combining interviews with health/laboratory and physical examination studies since 1959. The end-product, recently known as, National Health and Nutrition Examination Surveys (NHANES) provide cross-sectional data of the health and nutrition of the United States population. This information source has been central to formulating nationwide public health policies and practices.

6.2 Survey history

Overall NHANES survey history



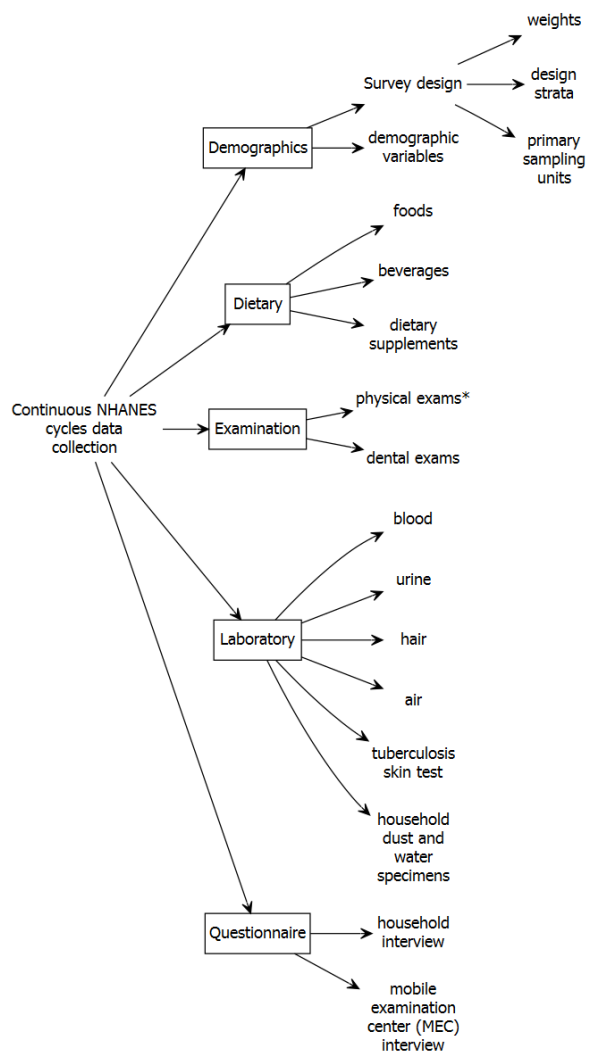
6.3 NHANES datafile and documents

6.3.1 File format

The Continuous NHANES files are stored in the NHANES website as SAS transport file formats (.xpt). You can import this data in any statistical package that supports this file format.

6.3.2 Continuous NHANES Components

Continuous NHANES components separated to reduce the amount of time to download and documentation size:



6.3.3 Public release excludes

The following data have not been released on the NHANES website as public release files due to confidentiality concerns:

- adolescent data on alcohol use,
- smoking,
- sexual behavior,
- reproductive health and drug use

6.3.4 Combining data

6.3.4.1 Different cycles

It is possible to combine datasets from different years/cycles together in NHANES. However, NHANES is a cross-sectional data, and identification of the same person across different cycles is not possible in the public release datasets. For appending data from different cycles, please make sure that the variable names/labels are the same/identical in years under consideration (in some years, names and labels do change).

6.3.4.2 Within the same cycle

Within NHANES datasets in a given cycle, each sampled person has a unique identifier sequence number (variable `SEQN`) and therefore various data components:

- demographics,
- dietary,
- examination,
- laboratory and
- questionnaire

within same cycle can be merged.

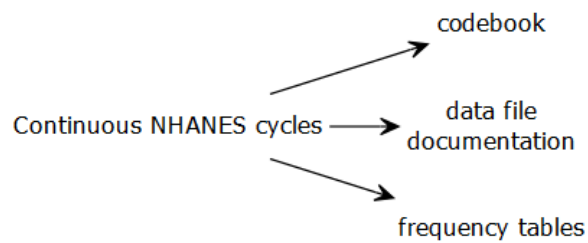
6.3.5 Missing data and outliers

CDC (2018) recommends:

1. “As a general rule, if 10% or less of your data for a variable are missing from your analytic dataset, it is usually acceptable to continue your analysis without further evaluation or adjustment. However, if more than 10% of the data for a variable are missing, you may need to determine whether the missing values are distributed equally across socio-demographic characteristics, and decide whether further imputation of missing values or use of adjusted weights are necessary.”
2. “If you fail to identify ‘refusal’ or ‘do not know’ as types of missing data, and treat the assigned values for ‘refused’ or ‘do not know’ as real values, you will get distorted results in your statistical analyses. Therefore, it is important to recode ‘refused’ or ‘don’t know’ responses as missing values (either as a period (.) for numeric variables or as a blank for character variables).”

3. “Outliers with extremely large weights could have an influential impact on your estimates. You will have to decide whether to keep these influential outliers in your analysis or not. It is up to the analysts to make that decision.”

6.3.6 NHANES documents



6.4 Exercise (web)

- More information about NHANES design
- Visit US CDC and do a variable keyword search based on your research interest (e.g., arthritis).

Chapter 7

Importing NHANES to R

This is a short instruction document of how to get NHANES dataset from the US CDC site to your RStudio environment. Once we bring the dataset into RStudio, the next step is to think about creating analytic dataset.

7.1 NHANES Dataset

National Center for Health Statistics (NCHS) conducts National Health and Nutrition Examination Survey (NHANES) (CDC,NCHS (2018)). These surveys are designed to evaluate the health and nutritional status of U.S. adults and children. These surveys are being administered in two-year cycles or intervals starting from 1999-2000. Prior to 1999, a number of surveys were conducted (e.g., NHANES III), but in our discussion, we will mostly restrict our discussions to ‘continuous NHANES’ (e.g., NHANES 1999-2000 to NHANES 2017-2018).

Witin the CDC website, continuous NHANES data are available in 5 categories:

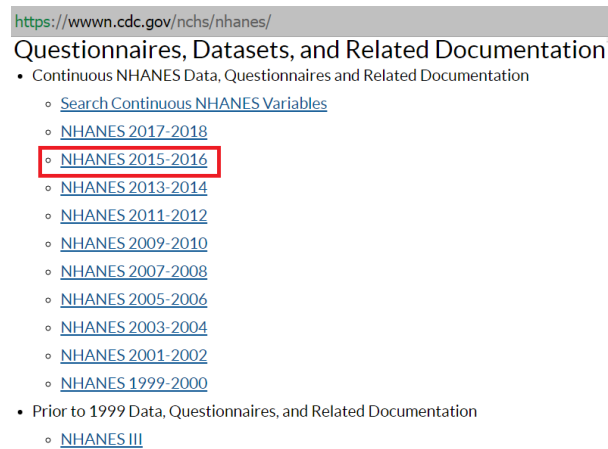
- Demographics
- Dietary
- Examination
- Laboratory
- Questionnaire

7.2 Accessing NHANES Data

In the following example, we will see how to download ‘Demographics’ data, and check associated variable in that data.

7.2.1 Accessing NHANES Data Directly from the CDC website

NHANES 1999-2000 and onward survey datasets are publicly available at www.cdc.gov/nchs/nhanes/.



- **Step 1:** Say, for example, we are interested about NHANES 2015-2016 surveys. Clicking the associated link in the above Figure gets us to the page for the cirresponding cycle (see below).

NHANES 2015-2016

Contents in Detail

- [Survey Questionnaires](#)
- [Examination and Laboratory Procedure Manuals](#)
- [Brochures and Consent Documents](#)

Using the Data

- [NHANES 2015-2016 Overview](#)
- [Technical Notes for Data Release](#)
- [Survey Methods and Analytic Guidelines](#)
- [Response Rates and Population Totals](#)
- [NHANES Web Tutorial](#)

Data, Documentation, Codebooks, SAS Code

- [Demographics](#)
- [Dietary](#)
- [Examination](#)
- [Laboratory](#)
- [Questionnaire](#)
- [Limited Access](#)

- **Step 2:** There are various types of data available for this survey. Let's explore the demographic information from this cycle. These data are mostly available in the form of SAS 'XPT' format (see below).

NHANES 2015-2016 Demographics Data

Data File Name	Doc File	Data File	Date Published
Demographic Variables and Sample Weights	DEMO_1 Doc	DEMO_1 Data (XPT - 3.6 MB)	September, 2017

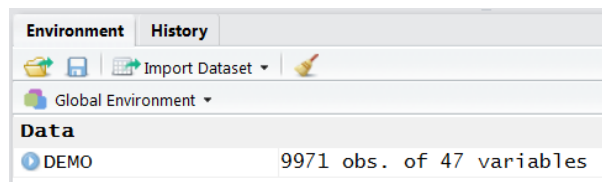
- **Step 3:** We can download the XPT data in the local PC folder and read the data into R as follows:

```
# install.packages("SASxport")
require(SASxport)
library(foreign)
DEMO <- read.xport("SurveyData\\DEMO_1.XPT")
```

```
##
## Attaching package: 'foreign'
```

```
## The following objects are masked from 'package:SASxport':
##
##      lookup.xport, read.xport
```

- **Step 4:** Once data is imported in RStudio, we will see the DEMO object listed under data window (see below):



- **Step 5:** We can also check the variable names in this DEMO dataset as follows:

```
names(DEMO)
```

```
## [1] "SEQN"      "SDDSRVYR" "RIDSTATR" "RIAGENDR" "RIDAGEYR" "RIDAGEMN"
## [7] "RIDRETH1" "RIDRETH3" "RIDEXMON" "RIDEXAGM" "DMQMILIZ" "DMQADFC"
## [13] "DMDBORN4" "DMDCITZN" "DMDYRSUS" "DMDEDUC3" "DMDEDUC2" "DMDMARTL"
## [19] "RIDEXPRG" "SIALANG" "SIAPROXY" "SIAINTRP" "FIALANG" "FIAPROXY"
## [25] "FIAINTRP" "MIALANG" "MIAPROXY" "MIAINTRP" "AIALANG" "DMDHHSIZ"
## [31] "DMDFMSIZ" "DMDHHSZA" "DMDHHSZB" "DMDHHSZE" "DMDHRGND" "DMDHRAGE"
## [37] "DMDHRBR4" "DMDHREDU" "DMDHRMAR" "DMDHSEDU" "WTINT2YR" "WTMEC2YR"
## [43] "SDMVPSU"  "SDMVSTRA" "INDHHIN2" "INDFMIN2" "INDFMPIR"
```

- **Step 6:** We can open the data in RStudio in the dataview window (by clicking the DEMO data from the data window). The next Figure shows only a few columns and rows from this large dataset. Note that there are some values marked as “NA”, which represents missing values.

DMDHSEDU HHS ref person's spouse's education level	WTINT2YR Full sample 2 year interview weight	WTMEC2YR Full sample 2 year MEC exam weight	SDMVPSU Masked variance pseudo-PSU	SDMVSTRA Masked variance pseudo-stratum
NA	9964.725	9860.625	1	120
5	44749.890	46173.307	2	124
NA	9891.944	10963.314	1	119
5	37043.087	39353.307	2	128
4	22744.355	23557.163	1	125
4	18526.180	18249.326	2	122
NA	20395.535	20068.663	2	126
NA	24788.723	25399.385	2	126
4	10998.012	11273.998	2	129
NA	34513.078	35673.964	1	121
NA	10988.317	11184.295	2	131
4	60125.441	63059.813	2	131
5	96194.928	97001.988	1	125
2	14862.011	14802.214	1	128

- **Step 7:** There is a column name associated with each column, e.g., DMDHSEDU in the first column in the above Figure. To understand what the column names mean in this Figure, we need to take a look at the codebook. To access codebook, click the 'DEMO|Doc' link (in step 2). This will show the data documentation and associated codebook (see the next Figure).

TABLE OF CONTENTS	
•	Component Description
•	Eligible Sample
•	Interview Setting and Mode of Administration
•	Quality Assurance & Quality Control
•	Data Processing and Editing
•	Analytic Notes
•	References
•	Codebook
•	SEQN - Respondent sequence number
•	SDDSRVYR - Data release cycle
•	RIDSTATR - Interview/Examination status
•	RIAGENDR - Gender
•	RIDAGEYR - Age in years at screening
•	DMDHRMAR - HH ref person's marital status
•	DMDHSEDU - HH ref person's spouse's education level
•	WTINT2YR - Full sample 2 year interview weight
•	WTMEC2YR - Full sample 2 year MEC exam weight

- **Step 8:** We can see a link for the column or variable DMDHSEDU in the table of content (in the above Figure). Clicking that link will provide us further information about what this variable means (see the next Figure).

DMDHSEDU - HH ref person's spouse's education level				
Variable Name:	DMDHSEDU			
SAS Label:	HH ref person's spouse's education level			
English Text:	HH reference person's spouse's education level			
Target:	Both males and females 0 YEARS - 150 YEARS			
Code or Value	Value Description	Count	Cumulative	Skip to Item
1	Less Than 9th Grade	619	619	
2	9-11th Grade (Includes 12th grade with no diploma)	511	1130	
3	High School Grad/GED or Equivalent	980	2110	
4	Some College or AA degree	1462	3572	
5	College Graduate or above	1629	5201	
7	Refused	2	5203	
9	Don't Know	23	5226	
.	Missing	4745	9971	

- **Step 9:** We can assess if the numbers reported under count and cumulative (from the above Figure) matches with what we get from the DEMO data we just imported (particularly, for the DMDHSEDU variable):

```
table(DEMO$DMDHSEDU)
```

```
##
##      1      2      3      4      5      7      9
## 619  511  980 1462 1629      2     23
```

```
cumsum(table(DEMO$DMDHSEDU))
```

```
##      1      2      3      4      5      7      9
## 619 1130 2110 3572 5201 5203 5226
```

```
length(is.na(DEMO$DMDHSEDU))
```

```
## [1] 9971
```

7.2.2 Accessing NHANES Data Using R Packages

7.2.2.1 nhanesA

`nhanesA` provides a convenient way to download and analyze NHANES survey data.

```
#install.packages("nhanesA")
library(nhanesA)
```

- **Step 1:** Within the CDC website, NHANES data are available in 5 categories
 - Demographics (DEMO)
 - Dietary (DIET)
 - Examination (EXAM)
 - Laboratory (LAB)
 - Questionnaire (Q)

To get a list of available variables within a datafile, we run the following command (e.g., we check variable names within DEMO data):

```
library(nhanesA)
```

```
## Warning: package 'nhanesA' was built under R version 4.0.2
```

```
nhanesTables(data_group='DEMO', year=2015)
```

```
##   FileName                                Description
## 1   DEMO_I Demographic Variables and Sample Weights
```

- **Step 2:** We can obtain the summaries of the downloaded data as follows (see below):

```
demo <- nhanes('DEMO_I')
```

```
## Processing SAS dataset DEMO_I    ..
```

```
names(demo)
```

```
## [1] "SEQN"      "SDDSRVYR" "RIDSTATR" "RIAGENDR" "RIDAGEYR" "RIDAGEMN"
## [7] "RIDRETH1" "RIDRETH3" "RIDEXMON" "RIDEXAGM" "DMQMILIZ" "DMQADFC"
## [13] "DMDBORN4" "DMDCITZN" "DMDYRSUS" "DMDEDUC3" "DMDEDUC2" "DMDMARTL"
## [19] "RIDEXPRG" "SIALANG" "SIAPROXY" "SIAINTRP" "FIALANG" "FIAPROXY"
## [25] "FIAINTRP" "MIALANG" "MIAPROXY" "MIAINTRP" "AIALANGA" "DMDHHSIZ"
## [31] "DMDFMSIZ" "DMDHHSZA" "DMDHHSZB" "DMDHHSZE" "DMDHRGND" "DMDHRAGE"
## [37] "DMDHRBR4" "DMDHREDU" "DMDHRMAR" "DMDHSEDU" "WTINT2YR" "WTMEC2YR"
## [43] "SDMVPSU"  "SDMVSTRA" "INDHHIN2" "INDFMIN2" "INDFMPPIR"
```

```
table(demo$DMDHSEDU)
```

```
##
##      1      2      3      4      5      7      9
## 619  511  980 1462 1629      2     23
```

```
cumsum(table(demo$DMDHSEDU))
```

```
##      1      2      3      4      5      7      9  
## 619 1130 2110 3572 5201 5203 5226
```

```
length(is.na(demo$DMDHSEDU))
```

```
## [1] 9971
```

7.2.2.2 RNHANES

RNHANES (Susmann (2016)) is another packages for downloading the NHANES data easily. Try yourself.

Bibliography

- Bilder, C. R. and Loughin, T. M. (2014). *Analysis of categorical data with R*. CRC Press.
- CDC (2018). Nhanes web tutorial frequently asked questions (faqs). <https://www.cdc.gov/nchs/tutorials/NHANES/FAQs.htm>. Accessed 2018-08-06.
- CDC,NCHS (2018). National health and nutrition examination survey data. <https://wwwn.cdc.gov/nchs/nhanes/>. [Online; accessed 11-April-2018].
- Dobson, A. and Barnett, A. (2008). *An introduction to generalized linear models, third edition*. Chapman and Hall/CRC.
- Heeringa, S. G., West, B. T., and Berglund, P. A. (2017). *Applied survey data analysis*. Chapman and Hall/CRC.
- Lumley, T. (2011). *Complex surveys: a guide to analysis using R*, volume 565. John Wiley & Sons.
- Susmann, H. (2016). *RNHANES: Facilitates Analysis of CDC NHANES Data*. R package version 1.1.0.
- Vittinghoff, E., Glidden, D. V., Shiboski, S. C., and McCulloch, C. E. (2011). *Regression methods in biostatistics: linear, logistic, survival, and repeated measures models*. Springer Science & Business Media.