|  |  |  |
| --- | --- | --- |
|  | **European Programme**  **For Intervention**  **Epidemiology Training** | Logo_Epiet_Short_WhiteBG copie |

**Rapid Assessment and Survey methods**

**R guide for Childhood Vaccination coverage survey in Greece, 2006**

**May 8th -13th 2017**

**Athens, Greece**

**Authors of case study:** Kostas Danis, Dimitris Papamichail, Takis Panagiotopoulos

**Authors of R guides**: Patrick Keating and Alexander Spina

Copyright and License

**Source:**

The R guides for this case study were created by Patrick Keating and Alexander Spina

**Contributors to R code:**  
Alexandre Blake (Epicentre)

**You are free:**

* **to Share** — to copy, distribute and transmit the work
* **to Remix** — to adapt the work

**Under the following conditions:**

* **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). The best way to do this is to keep as it is the list of contributors: sources, authors and reviewers.
* **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. Your changes must be documented. Under that condition, you are allowed to add your name to the list of contributors.
* You cannot sell this work alone but you can use it as part of a teaching.

**With the understanding that:**

* **Waiver** — Any of the above conditions can be [**waived**](http://creativecommons.org/licenses/by-sa/3.0/) if you get permission from the copyright holder.
* **Public Domain** — Where the work or any of its elements is in the [**public domain**](http://wiki.creativecommons.org/Public_domain) under applicable law, that status is in no way affected by the license.
* **Other Rights** — In no way are any of the following rights affected by the license:
  + Your fair dealing or [**fair use**](http://wiki.creativecommons.org/Frequently_Asked_Questions#Do_Creative_Commons_licenses_affect_fair_use.2C_fair_dealing_or_other_exceptions_to_copyright.3F) rights, or other applicable copyright exceptions and limitations;
  + The author's [**moral**](http://wiki.creativecommons.org/Frequently_Asked_Questions#I_don.E2.80.99t_like_the_way_a_person_has_used_my_work_in_a_derivative_work_or_included_it_in_a_collective_work.3B_what_can_I_do.3F) rights;
  + Rights other persons may have either in the work itself or in how the work is used, such as [**publicity**](http://wiki.creativecommons.org/Frequently_Asked_Questions#When_are_publicity_rights_relevant.3F) or privacy rights.
* **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work by keeping together this work and the current license.**This licence is based on** [**http://creativecommons.org/licenses/by-sa/3.0/**](http://creativecommons.org/licenses/by-sa/3.0/)

# Session 4 - Descriptive analysis

# Prerequisites

Participants are expected to be familiar with data management and basic analysis in R

# An introduction to the R companion

This text was adapted from the introduction used at the 2016 TSA module.

R packages are bundles of functions which extend the capability of R. Thousands of add-on packages are available in the main online repository (known as CRAN) and many more packages in development can be found on GitHub. They may be installed and updated over the Internet.

We will mainly use packages which come ready installed with R (base code), but where it makes things easier we will use add-on packages. In addition, we have included a few extra functions to simplify the code required. All the R packages you need for the exercises can be installed over the Internet.

# Installing required packages for the week  
required\_packages <- c("foreign", "Hmisc", "dplyr", "survey")  
  
install.packages(required\_packages)

Run the following code at the beginning of the case study to make sure that you have made available all the packages and functions that you need. Be sure to include it in any scripts too.

# Loading required packages for the week  
required\_packages <- c("foreign", "Hmisc", "dplyr", "survey")  
  
for (i in seq(along = required\_packages))  
 library(required\_packages[i], character.only = TRUE)

# Function to make tables with counts and proportions   
big.table <- function(data) {  
 count <- table(data)  
 prop <- round(prop.table(count)\*100, digits = 2)  
 cbind(count,  
 prop)   
}  
  
# Function used to calculate weighted proportions, CI and design effect  
svy\_prop <- function(x, design) {  
p1 <- round(svyciprop(as.formula(paste0( "~" , x)), design, na.rm = T) \* 100, digits = 2)  
p2 <- round(confint(p1) \* 100, digits = 2)  
p3 <- deff(round(svymean(as.formula(paste0( "~" , x)), design, na.rm = T, deff = T) \* 100, digits = 2))  
p4 <- cbind("Proportion" = p1, p2, "Design effect" = p3)  
}

R and Stata have minor differences in default settings and methods. In this document we will follow the Stata analysis as closely as possible, but small and usually unimportant differences may be noted between the statistical findings in R and those in Stata. Despite the differences, the findings from each statistical package should be comparable. At some points additional steps (which would usually be optional in R) will be taken to produce output which is comparable to that of Stata.

The big.table function uses data directly and allows combining of counts, proportions and cumulative sums, thus reducing the number of lines of code required for descriptive analyses. The svy\_prop function is used to calculate proportions, CIs, and the design effect of weighted variables.

You will work with Stata.dta data sets which can be loaded into R with the "foreign" or "readstata13" packages. The appropriate functions to use will be indicated.

R can hold one or many data sets in memory simultaneously, so there is usually no need to save intermediate files or close and re-open datasets.

# Descriptive analysis

Start a new R script, name it, for example, **session4.r** and save it in your working directory. Write all commands in the R script so that you can run (and re-run) it when needed during the exercise.

Open the **vaccine4.dta** dataset

vaccine <- read.dta("vaccine4.dta", convert.factors = FALSE)

#### Describing your dataset

You can view the structure of your data set using the following commands:

str(vaccine)  
summary(vaccine)  
describe(vaccine)

# No. of school classes that participated  
describe(vaccine$school)

## vaccine$school   
## n missing distinct Info Mean Gmd .05 .10   
## 4387 0 342 1 206.6 118.5 19.3 45.0   
## .25 .50 .75 .90 .95   
## 121.0 220.0 297.0 331.0 351.0   
##   
## lowest : 1 2 3 4 6, highest: 366 367 368 369 371

# Proportion of records where a vaccination booklet was received  
big.table(vaccine$vaccrec)

## count prop  
## 0 509 11.6  
## 1 3878 88.4

# Perform Student's t-test for comparison of means (for numeric variables)  
ttestage <- t.test(age~vaccrec, var.equal = TRUE, data = vaccine)  
  
# Chi2 test for comparison of proportions:  
genderresponse <- table(vaccine$gender, vaccine$vaccrec)  
chisq.test(genderresponse)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: genderresponse  
## X-squared = 1.9268, df = 1, p-value = 0.1651

**See appendix for an example of code showing how to make table 4.1**

## Task 4.3 - Calculate sampling weights

#### Generate a new variable for sampling weights for each stratum in your dataset.

To create sampling weights for each stratum in your dataset, you may use the following:

# Create a 2-columned matrix that contains the population values per stratum  
popvar <- matrix(c(11,12,21,22,31,32,41,51,52,61,62,3282,1773,8609,3609,24311,6734,30345,12928,4813,3069,1132),  
 nrow = 11, ncol = 2, byrow = F)  
  
# Rename the column names for ease of merging  
colnames(popvar) <- c("strata", "population")  
  
# Merge the original data set with the matrix  
vaccine <- merge(vaccine, popvar[, c("strata", "population")], by = "strata", all.x = T)

We will now use the **dplyr package** to create a new variable called **sample** which is the total number of records (rows) per stratum. For more information on dplyr see https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html

We will use the **mutate** function of dplyr to add a new variable to the vaccine dataset.

# Using dplyr, we create the sample variable, for which each row is equal to the total number of rows in a specific stratum  
# The %>% can be read below as "then"  
  
# First, we take the data frame vaccine, then  
vaccine <- vaccine %>%  
 # We group vaccine data by strata, then   
 group\_by(strata) %>%  
 # We create a new variable, sample, using n(), to count the number of rows per stratum   
 mutate(sample = n())

# You can view this new variable along with other key variables  
View(vaccine[,c("id","school", "strata","population", "sample")])

We now create the sampling fraction variable (samplef) which is the sample/population and then, finally the weight variable.

vaccine$samplef <- vaccine$sample/vaccine$population  
  
# The weight variable is the inverse of the sampling fraction  
vaccine$weight1 <- 1/vaccine$samplef

# You can view these new variables  
View(vaccine[,c("id","school", "strata","population","sample", "samplef", "weight1")])

## 

## Task 4.4 - Calculate weighted proportions

You can think of a weighted proportion as:

* a weighted average of each observation of 0 and 1, where the weights for each stratum are equal to Ni/ni (i.e. inversely proportional to the sampling fraction of the ith stratum); where: Ni=total population of stratum i; and ni=sample size of stratum i; or
* as a weighted average of the stratum specific proportions, where the weights are equal to Ni (i.e. proportional to the stratum population sizes).

The researchers aimed to obtain separate estimates for the urban and rural sectors of the population in each region (strata) as well as of the whole country.

#### 

#### Calculate the vaccination coverage of MMR-2 (variable name mmr2yn) for each stratum

We will first make a new data set that only keeps records of children that had vaccination booklets.

vacc\_rec <- vaccine[vaccine$vaccrec == 1, ]

There are multiple ways that you could create table 4.3 including the vaccination coverage of MMR-2. Below is one way which involves a number of steps.

**Step 1:** Obtain the total number of records, total sample, sampling fraction, weight per stratum using the **distinct** function of the the **dplyr package**:

# creates a table with unique values of population, sample, samplef and weight1 by strata  
  
# We make table1 using 5 variables from the vacc\_rec data set, then  
table1 <- vacc\_rec[, c("strata", "population", "sample", "samplef", "weight1")] %>%  
   
# We group this data subset by strata, then   
 group\_by(strata) %>%  
   
# We extract distinct/unique rows for each of the following variables  
 distinct(population, sample, samplef, weight1)

table1

## Source: local data frame [11 x 5]  
## Groups: strata [11]  
##   
## population sample samplef weight1 strata  
## <dbl> <int> <dbl> <dbl> <dbl>  
## 1 3282 435 0.13254113 7.544828 11  
## 2 1773 247 0.13931190 7.178138 12  
## 3 8609 477 0.05540713 18.048218 21  
## 4 3609 188 0.05209199 19.196809 22  
## 5 24311 572 0.02352844 42.501748 31  
## 6 6734 109 0.01618652 61.779817 32  
## 7 30345 1029 0.03391003 29.489796 41  
## 8 12928 512 0.03960396 25.250000 51  
## 9 4813 158 0.03282776 30.462025 52  
## 10 3069 388 0.12642555 7.909794 61  
## 11 1132 272 0.24028269 4.161765 62

**Step 2:** Calculate the proportion of records that received MMR-2.

# Using dplyr, we calculate the number of rows by strata and by level of mmr2yn (0 and 1), and create the freq variable to calculate the % per strata and level of mmr2yn  
  
# We take the variables strata and mmr2yn from vacc\_rec, then  
table2 <- vacc\_rec[, c("strata", "mmr2yn")] %>%  
   
# We group them by strata and level of mmr2yn (0 and 1), then   
 group\_by(strata, mmr2yn) %>%  
   
# we create a variable that counts the number of rows by strata and level of mmr2yn, then  
 summarise(counts = n()) %>%

# We create a new variable freq which is the proportion of each level of mmr2yn per stratum  
 mutate(freq = round(counts/sum(counts) \* 100, digits = 2))

table2

## Source: local data frame [22 x 4]  
## Groups: strata [11]  
##   
## strata mmr2yn counts freq  
## <dbl> <int> <int> <dbl>  
## 1 11 0 123 32.03  
## 2 11 1 261 67.97  
## 3 12 0 82 36.12  
## 4 12 1 145 63.88  
## 5 21 0 99 22.86  
## 6 21 1 334 77.14  
## 7 22 0 48 27.91  
## 8 22 1 124 72.09  
## 9 31 0 116 21.76  
## 10 31 1 417 78.24  
## # ... with 12 more rows

**Step 3:** Merge tables 1 and 2 but only where mmr2yn = 1, using **strata** as the merging variable.

table3 <- merge(table1, table2[table2$mmr2yn == 1, c("strata", "freq")], by = "strata")  
  
table3

## strata population sample samplef weight1 freq  
## 1 11 3282 435 0.13254113 7.544828 67.97  
## 2 12 1773 247 0.13931190 7.178138 63.88  
## 3 21 8609 477 0.05540713 18.048218 77.14  
## 4 22 3609 188 0.05209199 19.196809 72.09  
## 5 31 24311 572 0.02352844 42.501748 78.24  
## 6 32 6734 109 0.01618652 61.779817 83.96  
## 7 41 30345 1029 0.03391003 29.489796 76.67  
## 8 51 12928 512 0.03960396 25.250000 72.14  
## 9 52 4813 158 0.03282776 30.462025 66.21  
## 10 61 3069 388 0.12642555 7.909794 82.57  
## 11 62 1132 272 0.24028269 4.161765 75.00

**Step 4:** Tidy up the table by rounding figures and adding in total values.

# Rounds value of samplef  
table3$samplef <- round(table3$samplef \* 100, digits = 2)  
  
## Adding additional row at end of table  
table3 <- table3[1:12,]  
  
# Label first cell in 12th row as Total  
table3[12,1] <- "Total"  
  
# Calculate sum of population and put value in 12th row, 2nd column  
table3[12,2] <- colSums(table3[1:11, c("population"),drop = F])  
  
# Obtain the unweighted proportion of MMR-2 vaccination coverage  
unweighted <- prop.table(table(vacc\_rec$mmr2yn))  
unweighted <- round(unweighted \* 100, digits = 2)  
unweighted <- as.data.frame(unweighted)   
  
# Incorporate the unweighted proportion of value into table 3  
table3[12,6] <- unweighted[2,2]

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | strata | population | sample | samplef | weight1 | freq |
| 1 | 11 | 3282 | 435 | 13.25 | 7.544828 | 67.97 |
| 2 | 12 | 1773 | 247 | 13.93 | 7.178138 | 63.88 |
| 3 | 21 | 8609 | 477 | 5.54 | 18.048218 | 77.14 |
| 4 | 22 | 3609 | 188 | 5.21 | 19.196808 | 72.09 |
| 5 | 31 | 24311 | 572 | 2.35 | 42.501748 | 78.24 |
| 6 | 32 | 6734 | 109 | 1.62 | 61.779817 | 83.96 |
| 7 | 41 | 30345 | 1029 | 3.39 | 29.489796 | 76.67 |
| 8 | 51 | 12928 | 512 | 3.96 | 25.250000 | 72.14 |
| 9 | 52 | 4813 | 158 | 3.28 | 30.462025 | 66.21 |
| 10 | 61 | 3069 | 388 | 12.64 | 7.909794 | 82.57 |
| 11 | 62 | 1132 | 272 | 24.03 | 4.161765 | 75.00 |
| NA | Total | 100605 | NA | NA | NA | 74.78 |

Suppose there are only three strata (stratum 11, 12, 21) in the country. Based on the information provided in Table 4.2, calculate the weighted proportion of children vaccinated with MMR-2 in the country (three strata only).

# Using formula 4.2 and information on Table 4.2, you should have:  
((3282\*0.6797) + (1773\*0.6388) + (8609\*0.7714))/(3282 + 1773 + 8609)

## [1] 0.7321685

# For the 11 strata in the study, you could type:  
((3282\*0.6797) + (1773\*0.6388) + (8609\*0.7714) + (3609\*0.7209) + (24311\*0.7824) + (6734\*0.8396) + (30345\*0.7667) + (12928\*0.7214) + (4813\*0.6621) + (3069\*0.8257) + (1132\*0.75)) / (100605)

## [1] 0.7598271

The above suggests that allowing for the sampling weights, 76% of children received MMR-2. This differs from the simple (not weighted) proportion, which is 75%.

## 

## Task 4.5 - Estimate vaccination coverage

You will now estimate the vaccination coverage of children for different vaccines and in different settings, using the **survey** package in R.

#### Calculate the proportion of children that were fully vaccinated (vacful) and the corresponding 95%CI

* as if simple random sampling were used
* allowing for the weights
* allowing for the weights and clustering
* allowing for the weights, clustering and stratification

Complete Table 4.4.1 Comment on the results.

## 

## Help, Task 4.5

You can use the survey package in R, which allows you to analyse survey data taking into account the sampling design (stratification, multistage sampling, cluster sampling) e.g. calculate proportions, allowing for the weights, stratification and clustering. These commands can be used for means, proportions, ratios and sums, but also for the estimation of regression coefficients (univariate and multivariate regressions).

You need to use the function **svydesign** to define the design to be applied to the survey.

More information can be found in R help, but the essentials are summarised below:

*svydesign(ids, strata = NULL, weights = NULL, data = NULL)*

* **ids**: designates the name of the variable that contains identifiers for the primary sampling units (e.g. clusters = school in this study). Ids take a value of **~0 or ~1** if there are no clusters
* **strata**: designates the name of the variable that contains identifiers for the strata (e.g. strata = urban/rural areas of each region in this study), to allow for stratification
* **weights**: designates the variable of the sampling weight used, i.e. allows for the sampling weight of each individual
* **data**: specifies the relevant data frame

You may use the function **svymean** or **svyciprop** to estimate proportions for **dichotomous variables**. Of note, **syvciprop** provides **more accurate confidence intervals** compared to svymean. However, it doesn't provide the design effect. We will use a combination of svyciprop and svymean to obtain all the elements we need. **svyby** can be used to estimate proportions in different subpopulations (see below for explanations of options for each function).

*svymean(x, design, na.rm = FALSE, deff = FALSE)*

The options for svymean include:

* **x**: a variable, a formula or matrix
* **design**: the object created using svydesign
* **na.rm**: should cases with missing values be dropped?
* **deff**: return the design effect

*svyciprop(formula, design, level = 0.95, method = c("logit", "likelihood", "asin", "beta", "mean", "xlogit"))*

The options for svyciprop include:

* **formula**: model formulat specifying a single binary variable
* **design**: the object created using svydesign
* **level**: confidence interval level
* **method**: method to be used

*svyby(formula, by, design, FUN, deff = FALSE, vartype = c("se","ci","cv","cvpct","var")*

The options for svyby include:

* **formula,x**: a variable, a formula or matrix to pass to the function
* **by**: a formula specifying factors that define subsets, or a list of factors
* **design**: the object created using svydesign
* **FUN**: a function taking a formula and survey design object as its first two arguments
* **deff**: request a design effect from FUN
* **vartype**: report variability in terms of standard error, confidence intervals and more

#### 

#### Calculate the simple proportion (unweighted) and 95% CIs

# Make vacful a factor and reorder in order to obtain proportion/ CI  
vacc\_rec$vacful <- factor(vacc\_rec$vacful, levels = c(1, 0))  
  
# Use prop.test to obtain the proportion and the CI  
simple <- prop.test(table(vacc\_rec$vacful))

# Extract the proportion, confidence intervals from the simple table and add 1 as the design effect for a simple proportion  
simple <- rbind(round(simple$estimate \* 100, digits = 2),  
 round(simple$conf.int[1] \* 100, digits = 2),   
 round(simple$conf.int[2] \* 100, digits = 2),  
 1)  
  
# Make simple a data frame and use the transpose function to switch rows to columns  
simple <- as.data.frame(t(simple))  
  
# Add column names to the simple data frame  
colnames(simple)[1:4] <- c("Proportion", "2.5%", "97.5%","Design effect")  
simple

## Proportion 2.5% 97.5% Design effect  
## p 60.99 59.43 62.53 1

# We must convert vacful back to a numeric variable for the subsequent calculations, as follows:  
# Factor variables must be first converted to characters and then to numeric  
vacc\_rec$vacful <- as.numeric(as.character(vacc\_rec$vacful))

#### 

#### Calculate a weighted proportion and 95% CIs (taking into account the sampling weights)

We first need to define the survey design. By specifying that ids = ~1, we ignore clusters in the data.

design <- svydesign(ids = ~1, weights = ~weight1, data = vacc\_rec)

We can obtain the proportion, confidence intervals and design in a multi-step process as below.

# NB.any records that contain NAs must be dropped to calculate the proportion using svymean  
# We use svyciprop to calculate proportions and specifically for 95% CI of proportions  
a <- round(svyciprop(~vacful, design, na.rm = T) \* 100, digits = 2)  
# Save the 95% CI separately  
b <- round(confint(a) \* 100, digits = 2)  
# Use svymean to calculate the design effect  
c <- deff(round(svymean(~vacful, design, na.rm = T, deff = T) \* 100, digits = 2))  
# Put them all together with correct labelling  
d <- cbind("Proportion" = a, b, "Design effect" = c)  
d

## Proportion 2.5% 97.5% Design effect  
## vacful 62.96 61.17 64.71 1.406724

Alternatively, we can use the **svy\_prop** function (defined at the start of the case study), which carries out the above 4 steps but keeps the code neater.

e <- svy\_prop("vacful", design = design)  
e

## Proportion 2.5% 97.5% Design effect  
## vacful 62.96 61.17 64.71 1.406724

#### To estimate the proportion of children that were fully vaccinated, allowing for the weights and clustering:

To include clustering in the design, we define the primary sampling unit/ids as school.

design2 <- svydesign(ids = ~school, weights = ~weight1, data = vacc\_rec)  
  
# Use the svy\_prop function specifying that design2 should be used  
f <- svy\_prop("vacful", design = design2)  
f

## Proportion 2.5% 97.5% Design effect  
## vacful 62.96 60.62 65.23 2.374761

#### 

#### To estimate the proportion of children that were fully vaccinated, allowing for the weights, clustering and stratification:

To include stratification in the design, we define that strata equals strata.

design3 <- svydesign(ids = ~school, weights = ~weight1, strata = ~strata, data = vacc\_rec)  
g <- svy\_prop("vacful", design = design3)  
g

## Proportion 2.5% 97.5% Design effect  
## vacful 62.96 60.63 65.23 2.35846

To create table 4.4.1, we need to combine all of these elements together.

# Combine simple with e, f, and g  
table4 <- rbind(simple,e,f,g)  
  
# Round the values  
table4[, 2:4] <- round(table4[, 2:4], digits = 3)

# Add in row names  
rownames(table4) <- c("Simple proportion",  
 "+ sampling weight",  
 "+ sampling weight + clustering",  
 "+ sampling weight + clustering + stratification")

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Proportion | 2.5% | 97.5% | Design effect |
| Simple proportion | 60.99 | 59.43 | 62.53 | 1.000 |
| + sampling weight | 62.96 | 61.17 | 64.71 | 1.407 |
| + sampling weight + clustering | 62.96 | 60.62 | 65.23 | 2.375 |
| + sampling weight + clustering + stratification | 62.96 | 60.63 | 65.23 | 2.358 |

Of note, the CIs and the design effects obtained from R and Stata are not exactly the same, however, their interpretation remains similar.

Estimations are modified when the sampling design is taken into account:

* allowing for the sample weight modifies the estimate of vaccination coverage
* allowing for the clustering (or multistage design) decreases the precision of the estimate (higher variance and design effect)
* allowing for the stratification improves the precision of the estimate (lower variance and design effect) Note that the point estimate depends on the weights only, while the 95% CIs depend on everything (weights, stratification and clustering).

#### 

#### Design effects and intraclass correlation coefficient

A design effect of **1.407** suggests that the variability (variance or the square of the standard error) of the estimate under the chosen design is **40.7%** larger than that of the same-sized simple random sampling. Similarly, a design effect of 2.358 suggests that the variability of the estimate allowing for clustering, stratification and sampling weights is 135.8% larger than that would come from the same sample size if simple random sampling were used.

Rearranging the formula: deff=1+(n-1) · rho ==> rho=(deff-1)/(n-1) where deff=2.358 and n=20, you can get intraclass correlation coefficient (rho)=0.071

You may use this value of rho in sample size calculations in other similar surveys in the future. As mentioned earlier, the rho is the proportion of the total variation in the outcome that is between clusters; this measures the degree of similarity or correlation between subjects within the same cluster. The larger the rho-that is the tendency for subjects within a cluster to be similar-the greater the size of the design effect and the larger the additional number of subjects required to achieve the same precision.

Note that in the sample size calculations in session 2, deff and rho were expected to be higher (2 and 0,05, respectively). Hence, the sample size actually achieved was larger and the estimates were more precise than originally expected.

#### Calculate the vaccination coverage and complete Table 4.4.2

For example, DTP-3:

# Weighted vaccine coverage of DTP-3  
dtp <- svy\_prop("dtp3yn", design = design3)  
dtp

## Proportion 2.5% 97.5% Design effect  
## dtp3yn 99.19 98.77 99.47 1.543883

# Weighted vaccine coverage of DTP-3 by minority group  
dtm <- svyby(~dtp3yn, ~minority,  
 design3,  
 svyciprop, vartype = "ci")  
dtm <- round(dtm[,2:4] \* 100, digits = 2)  
dtm

## dtp3yn ci\_l ci\_u  
## 0 99.52 99.12 99.74  
## 1 79.65 55.95 92.35  
## 2 96.31 91.77 98.39  
## 3 98.99 96.40 99.72

# Weighted vaccine coverage of DTP-3 by area  
dtu <- svyby(~dtp3yn, ~urban,   
 design3,  
 svyciprop, vartype = "ci")  
dtu <- round(dtu[,2:4] \* 100, digits = 2)  
dtu

## dtp3yn ci\_l ci\_u  
## 0 98.37 96.76 99.19  
## 1 99.39 98.93 99.65

Creating table 4.4.2 will take a few steps, as outlined below.

**Step 1:** Use a loop to estimate overall vaccination coverage for all vaccines.

vars <- c("dtp3yn", "dtp4yn", "dtp5yn", "mmr1yn", "mmr2yn", "hibprmyn", "hibfulyn", "hbv3yn", "mnc1yn", "pne1yn", "var1yn", "vacful", "vactime")  
  
# Create an empty list to store the output of the loop  
output <- list()  
  
# for each variable in vars  
for (var in vars) {  
# Calculate the proportion, 95% CI and deff  
 overall <- svy\_prop(var, design = design3)  
 output[[var]] <- overall  
}  
  
# Bind dataframes from the list (output) as rows below each other in a matrix  
output2 <- do.call(rbind, output[1:length(output)])   
  
# Transform output2 to a dataframe  
output2 <- as.data.frame(output2)

**Step 2:** Use a loop to estimate vaccination coverage for DTP-3, complete vaccination and timely vaccination **by minority.**

vars <- c("dtp3yn", "vacful", "vactime")  
  
# Create an empty list to store the output of the loop  
output3 <- list()  
   
for (var in vars) {  
# Calculate proportions, 95% CI and deff for vars by minority  
 a <- svyby(as.formula(paste0( "~" , var)),  
 by = ~minority, design3,   
 svyciprop, vartype = "ci")  
# Change the column names to facilitate merging  
 colnames(a)[2:4] <- c("Proportion", "2.5%","97.5%")  
 a <- round(a[,2:4]\*100, digits = 1)  
 output3[[var]] <- a  
}   
  
# Bind dataframes from the list (output3) as rows below each other in a matrix  
output3 <- do.call(rbind, output3[1:length(output3)])  
  
# Need to add an empty Design effect variable to be able to combine all of the dataframes later  
output3$`Design effect` <- ""

**Step 3:** Use a loop to estimate vaccination coverage for DTP-3, complete vaccination and timely vaccination **by area.**

vars <- c("dtp3yn", "vacful", "vactime")  
  
# Create an empty list to store the output of the loop  
output4 <- list()

for (var in vars) {  
# Calculate proportions, 95% CI and deff for vars by area  
 b <- svyby(as.formula(paste0( "~" , var)),  
 by = ~urban, design3,   
 svyciprop, vartype = "ci")  
# Change the column names to facilitate merging   
 colnames(b)[2:4] <- c("Proportion", "2.5%","97.5%")  
 b <- round(b[,2:4]\*100, digits = 1)  
 output4[[var]] <- b   
}   
  
output4 <- do.call(rbind, output4[1:length(output4)])  
  
# Create an empty Design effect variable to be able to combine all of the dataframes later  
output4$`Design effect` <- ""

**Step 4:** Combine the tables, while trying to follow the table structure in the manual as much as possible.

finaltable <- rbind(output2[1,],  
 output3[1:4,],  
 output4[1:2,],  
 output2[2:12,],  
 output3[5:8,],  
 output4[3:4,],  
 output2[13,],  
 output3[9:12,],  
 output4[5:6,])

**Step 5:** Tidy up the table with rownames and rounding of variables (found in the appendix).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Proportion | 2.5% | 97.5% | Design effect |
| DTP-3 (overall) | 99.19 | 98.77 | 99.47 | 1.54 |
| DTP-3 General population | 99.50 | 99.10 | 99.70 |  |
| DTP-3 Roma | 79.70 | 56.00 | 92.30 |  |
| DTP-3 Greek Muslims | 96.30 | 91.80 | 98.40 |  |
| DTP-3 Immigrants | 99.00 | 96.40 | 99.70 |  |
| DTP-3 Rural areas | 98.40 | 96.80 | 99.20 |  |
| DTP-3 Urban areas | 99.40 | 98.90 | 99.60 |  |
| DTP-4 (overall) | 98.35 | 97.70 | 98.81 | 1.91 |
| DTP-5 (overall) | 90.07 | 88.80 | 91.22 | 1.71 |
| MMR-1 (overall) | 98.29 | 97.67 | 98.75 | 1.75 |
| MMR-2 (overall) | 76.00 | 73.60 | 78.24 | 3.1 |
| HiB- primary (overall) | 90.00 | 88.83 | 91.06 | 1.45 |
| HiB- full (overall) | 85.43 | 84.04 | 86.72 | 1.51 |
| HepB-3 (overall) | 95.30 | 94.36 | 96.10 | 1.75 |
| MNC-1 (overall) | 71.82 | 69.45 | 74.07 | 2.76 |
| PCV7-1 (overall) | 4.58 | 3.80 | 5.52 | 1.74 |
| Var-1 (overall) | 12.98 | 11.38 | 14.76 | 2.65 |
| Complete vaccination (overall) | 62.96 | 60.63 | 65.23 | 2.36 |
| Comp Vacc General population | 66.60 | 64.20 | 68.90 |  |
| Comp Vacc Roma | 24.30 | 9.10 | 50.50 |  |
| Comp Vacc Greek Muslims | 51.20 | 40.60 | 61.80 |  |
| Comp Vacc Immigrants | 30.30 | 25.00 | 36.20 |  |
| Comp Vacc Rural areas | 57.10 | 50.60 | 63.30 |  |
| Comp Vacc Urban areas | 64.30 | 61.90 | 66.80 |  |
| Timely vaccination (overall) | 51.24 | 48.72 | 53.75 | 2.59 |
| Timely vacc General population | 55.40 | 52.80 | 57.90 |  |
| Timely vacc Roma | 2.20 | 0.20 | 17.80 |  |
| Timely vacc Greek Muslims | 30.50 | 23.30 | 38.80 |  |
| Timely vacc Immigrants | 17.00 | 12.40 | 22.80 |  |
| Timely vacc Rural areas | 47.50 | 40.40 | 54.60 |  |
| Timely vacc Urban areas | 52.10 | 49.50 | 54.70 |  |

* Summarizing the results
* Vaccination coverage was high for vaccines included in the NVP before 1990 (i.e. DTP, Po, MMR). However, the uptake of 2 doses of MMR was much lower than the WHO target of 95%.
* Vaccination coverage was satisfactory for Hib and HepB that were introduced in the NVP in 1998 and 2002, respectively.
* Uptake for vaccines introduced in the NVP in 2006 ranged from high (MNC that was available in the pharmacies earlier) to very low (PCV7 and Var)
* Vaccination uptake was lower among Greek Roma and to a lesser degree among immigrants
* There were no substantial differences in the vaccination coverage in urban and rural areas.

**Types of practices administering vaccines.**

# The investigators also estimated the weighted proportions of practices administering MMR-1 and other vaccines. Almost 70% of MMR-1 vaccines were administered in private practices. Approximately 90% of the first doses of MMR among Greek Roma were administered in state practices, with the corresponding proportions among immigrants, Greek Muslims and the general population being 53%, 45% and 26%, respectively. Similar patterns were observed for most, but not all, vaccines

# Appendix

### Making table 4.1

You could use the following code to make a table similar to table 4.1:

# List your variales of interest  
vars <- c("gender", "urban", "minority", "country1")  
  
# Make an empty list to store output  
output <- list()  
  
# Use the loop to obtain count & proportions of variables in the full sample and by vaccrec, and result of chisq test  
for (var in vars) {  
 full <- big.table(vaccine[,var])  
 combo <- table(vaccine[,var], vaccine$vaccrec)  
 prop <- round(prop.table(combo,2)\*100,digits = 2)  
 test <- chisq.test(combo)  
 output[[var]] <- cbind(total,  
 "Respondents (n)" = combo[,c(2)],  
 "% respondents" = prop[,c(2)],  
 "Non-respondents (n)" = combo[,c(1)],   
 "% non-respondents" = prop[,c(1)],  
 Pvalue = round(test$p.value, digits = 3))  
}  
  
output  
  
## Use the Do.call function which will loop over output with rbind to create a neater table  
output2 <- do.call(rbind, output[1:length(output)])   
  
# Add in appropriate variable names  
rownames(output2) <- c("Female", "Male", "Rural areas", "Urban areas", "General population", "Roma", "Greek Muslims", "Immigrants", "Other country", "Greece")  
  
# Note that the final order isn't exactly the same as table 4.1  
output2

### Tidying up table 4.4.2

# Add appropriate rownames  
rownames(finaltable) <- c("DTP-3 (overall)", "DTP-3 General population", "DTP-3 Roma", "DTP-3 Greek Muslims", "DTP-3 Immigrants", "DTP-3 Rural areas", "DTP-3 Urban areas", "DTP-4 (overall)","DTP-5 (overall)", "MMR-1 (overall)", "MMR-2 (overall)", "HiB- primary (overall)", "HiB- full (overall)", "HepB-3 (overall)", "MNC-1 (overall)", "PCV7-1 (overall)", "Var-1 (overall)", "Complete vaccination (overall)", "Comp Vacc General population","Comp Vacc Roma", "Comp Vacc Greek Muslims", "Comp Vacc Immigrants", "Comp Vacc Rural areas", "Comp Vacc Urban areas", "Timely vaccination (overall)", "Timely vacc General population", "Timely vacc Roma", "Timely vacc Greek Muslims", "Timely vacc Immigrants", "Timely vacc Rural areas", "Timely vacc Urban areas")  
  
# Design effect needs to be numeric to round the values  
finaltable$`Design effect` <- as.numeric(finaltable$`Design effect`)  
  
# round values  
finaltable[,2:4] <- round(finaltable[,2:4], digits = 2)  
  
# convert NA values to ""   
finaltable[is.na(finaltable) == T ] <- ""

### 

### Tidying up table output using ReporterRs package

# Functions used to format tables  
label\_table <- function(X){  
 setFlexTableBorders(X,inner.vertical = borderProperties(style = "none"),inner.horizontal = borderProperties(style = "none"),outer.vertical = borderProperties(style = "none"),outer.horizontal = borderProperties(width = 2),body = T,header = T)  
}   
  
label\_footer <- function(X){  
 setFlexTableBorders(X,inner.vertical = borderProperties(style = "none"),inner.horizontal = borderProperties(style = "none"),outer.vertical = borderProperties(style = "none"),outer.horizontal = borderProperties(style = "none"),footer = T)  
}

# To make publication standard tables through R, you can use the FlexTable function from the ReporteRs package  
table3 <- FlexTable(table3,header.columns = F)  
  
# this adds a new row with those headings and you specify over how many columns each heading should span  
table3 <- addHeaderRow(table3, text.properties = textBold(), value = c("Stratum", "Total number of 1st year pupils", "Number of pupils selected in sample", "Sampling fraction (%)", "Sampling weight", "MMR-2 vaccination coverage (%)"), colspan = c(1,1,1,1,1,1))   
  
 # removed the label around the footer  
table3 <- label\_footer(table3)  
  
 # formatted the table so that only the top and lower parts are neatly formatted  
table3 <- label\_table(table3)   
  
 # Can export the table to e.g. powerpoint  
doc1 <- pptx()  
  
doc1 <- addSlide( doc1, "Two Content")  
doc1 <- addTitle( doc1 , 'Table3', level = 1)  
doc1 <- addFlexTable(doc1, flextable = table3)  
writeDoc( doc1, 'N:/MED/IMED-VIE/INFE/Public/CC-INFE-Schmid/EPIET/Output/Table3.pptx')

# Session 5

# An introduction to the R companion

This text was adapted from the introduction used at the 2016 TSA module.

R packages are bundles of functions which extend the capability of R. Thousands of add-on packages are available in the main online repository (known as CRAN) and many more packages in development can be found on GitHub. They may be installed and updated over the Internet.

We will mainly use packages which come ready installed with R (base code), but where it makes things easier we will use add-on packages. In addition, we have included a few extra functions to simplify the code required. All the R packages you need for the exercises can be installed over the Internet.

# Installing required packages for the week  
required\_packages <- c("foreign", "survey", "broom")  
  
install.packages(required\_packages)

Run the following code at the beginning of the case study to make sure that you have made available all the packages and functions that you need. Be sure to include it in any scripts too.

# Loading required packages for the week  
required\_packages <- c("foreign", "survey", "broom")  
  
for (i in seq(along = required\_packages))  
 library(required\_packages[i], character.only = TRUE)

# Function to make tables with counts and proportions   
big.table <- function(data) {  
 count <- table(data)  
 prop <- round(prop.table(count)\*100, digits = 2)  
 cbind(count,  
 prop)   
}  
  
# Function used to calculate weighted proportions, CI and design effect  
svy\_prop <- function(x, design) {  
p1 <- round(svyciprop(as.formula(paste0( "~" , x)), design, na.rm = T) \* 100, digits = 2)  
p2 <- round(confint(p1) \* 100, digits = 2)  
p3 <- deff(round(svymean(as.formula(paste0( "~" , x)), design, na.rm = T, deff = T) \* 100, digits = 2))  
p4 <- cbind("Proportion" = p1, p2, "Design effect" = p3)  
}  
  
# Function used to calculate weighted proportions, CI and design effect for categorical variables  
svy\_prop\_cat <- function(x, dataset, designids, designweight, designstrata){  
   
 #create a list to save outputs in   
   
 outer <- list()   
   
 #change each of the variables of interest to factor  
 dataset[,x] <- as.factor(dataset[,x])  
   
 #For each level of the categorical variable of interest  
 for (i in levels(dataset[,x])){  
   
 #create a new binary variable called var temp for each level  
 dataset[,paste0(x, "temp")] <- ifelse(dataset[,x] == i, 1, 0)  
   
 #define your design  
 design <- svydesign(ids = as.formula(paste0("~",designids)),   
 weights = as.formula(paste0("~",designweight)),   
 strata = as.formula(paste0("~",designstrata)),   
 data = dataset)  
   
 #use the svy\_prop function on binary variable  
 overall <- svy\_prop(paste0(var ,"temp"), design = design)  
   
 #label the rows of your output with the variable and category level  
 rownames(overall) <- paste0(var,": ", i)  
   
 #save within the list   
 outer[[var]][[i]] <- overall  
 }  
   
 #bind the output from the list in to a dataframe  
 do.call(rbind, outer[[1]][1:length(outer[[1]])])  
}

R and Stata have minor differences in default settings and methods. In this document we will follow the Stata analysis as closely as possible, but small and usually unimportant differences may be noted between the statistical findings in R and those in Stata. Despite the differences, the findings from each statistical package should be comparable. At some points additional steps (which would usually be optional in R) will be taken to produce output which is comparable to that of Stata.

The big.table function uses data directly and allows combining of counts, proportions and cumulative sums, thus reducing the number of lines of code required for descriptive analyses. The svy\_prop function is used to calculate proportions, CIs, and the design effect of weighted variables. The svy\_prop\_cat function does the same as svy\_prop, but specifically for each level of a categorical variable.

You will work with Stata.dta data sets which can be loaded into R with the "foreign" or "readstata13" packages. The appropriate functions to use will be indicated.

R can hold one or many data sets in memory simultaneously, so there is usually no need to save intermediate files or close and re-open datasets.

# Crude analysis

Start a new R script, name it, for example, **session5.r** and save it in your working directory. Write all commands in the R script so that you can run (and re-run) it when needed during the exercise.

Open the **vaccine5.dta** dataset

vaccine <- read.dta("vaccine5.dta", convert.factors = FALSE)

## 

## Task 5.1 - Describe parental characteristics, beliefs towards vaccination

In the original database, 175 parental questionnaires were excluded from further analysis, as some of their answers on beliefs and attitudes were found inconsistent. For simplicity reasons, you will now use vaccine5.dta for further analysis.

* Recode the following variables and add labels:
  + Mother's age (mage; <25, 25-29, ???30 years)
  + Number of other siblings in the family (osib1; ???3, 1-2, 0 other siblings)
* Calculate proportions (and corresponding 95%CI) allowing for the weights, stratification and clustering. Complete Table 5.1.

## Help, Task 5.1

Firstly, recoding the variables.

vaccine$mage2 <- cut(vaccine$mage,  
 breaks = c(0,24, 29, 100),   
 labels = c("0-24", "25-29", ">30"))  
  
  
vaccine$osib2 <- cut(vaccine$osib1,   
 breaks = c(-1,0, 2,100),  
 labels = c("0", "1-2", ">3"))

Calculate weighted proportions (and corresponding 95%CI).

# Specify the design to include weights, clustering and stratification  
design <- svydesign(ids = ~school, weights = ~weight, strata = ~strata, data = vaccine)  
  
  
# To complete table 5.1, select the following variables  
varsbin <- c("uncritical", "a1x1g", "a1x2g", "a1x5g", "a3x1g", "a3x2g", "a3x5g")  
  
# Create a list to store the output  
output <- list()  
  
# Apply the svy\_prop function to each variable  
for (var in varsbin) {  
 overall <- svy\_prop(var, design = design)  
 output[[var]] <- overall  
}  
  
# Convert the list to a matrix to condense all the output  
outputa <- do.call(rbind, output[1:length(output)])   
  
# Convert the output to a data frame  
outputa <- as.data.frame(outputa)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Proportion | 2.5% | 97.5% | Design effect |
| uncritical | 48.24 | 46.15 | 50.34 | 1.57274 |
| a1x1g | 99.84 | 99.50 | 99.95 | 1.84916 |
| a1x2g | 3.28 | 2.62 | 4.11 | 1.45755 |
| a1x5g | 19.43 | 17.95 | 21.00 | 1.28477 |
| a3x1g | 20.10 | 17.38 | 23.12 | 4.57977 |
| a3x2g | 20.86 | 19.09 | 22.75 | 1.78642 |
| a3x5g | 39.41 | 37.22 | 41.65 | 1.82605 |

Repeat the above steps, but this time for categorical variables.

# List your categorical variables of interest  
varscat <- c("minority", "mage2", "osib2", "educf", "a1posg")  
  
# Create a list to store the output  
output2 <- list()  
  
# Apply svy\_prop\_cat to each variable of interest  
for (var in varscat) {  
 overall <- svy\_prop\_cat(var, dataset = vaccine, designids = "school", designweight = "weight", designstrata = "strata")  
 output2[[var]] <- overall  
}  
  
# Convert the list to a matrix to condense all the output  
output2a <- do.call(rbind, output2[1:length(output2)])   
  
# Convert the output to a data frame  
output2a <- as.data.frame(output2a)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Proportion | 2.5% | 97.5% | Design effect |
| minority: 0 | 89.75 | 87.98 | 91.28 | 2.71053 |
| minority: 1 | 0.77 | 0.35 | 1.69 | 4.48133 |
| minority: 2 | 1.76 | 1.23 | 2.50 | 2.08138 |
| minority: 3 | 7.73 | 6.43 | 9.26 | 2.57053 |
| mage2: 0-24 | 0.78 | 0.51 | 1.21 | 1.37661 |
| mage2: 25-29 | 13.21 | 11.71 | 14.87 | 1.97564 |
| mage2: >30 | 86.00 | 84.28 | 87.57 | 2.03164 |
| osib2: 0 | 15.73 | 14.19 | 17.41 | 1.76429 |
| osib2: 1-2 | 77.10 | 75.34 | 78.77 | 1.51398 |
| osib2: >3 | 7.17 | 6.03 | 8.51 | 2.07090 |
| educf: 2 | 14.15 | 12.33 | 16.20 | 2.63789 |
| educf: 3 | 17.30 | 15.69 | 19.03 | 1.66713 |
| educf: 4 | 37.74 | 35.55 | 39.99 | 1.80139 |
| educf: 6 | 30.81 | 27.82 | 33.97 | 3.80339 |
| a1posg: 0 | 1.54 | 1.10 | 2.16 | 1.56882 |
| a1posg: 1 | 17.28 | 15.81 | 18.84 | 1.40569 |
| a1posg: 2 | 81.18 | 79.55 | 82.71 | 1.43179 |

* Summarising the findings on parents' opinions towards vaccination and perceived barriers to vaccination.

Parents/guardians indicated a high degree of perceived necessity of immunization with the vast majority agreeing that vaccination is beneficial for their children. In addition, 76% considered that vaccines are completely safe and never harm child health and 45% felt that children should be immediately immunized with every newly introduced vaccine (data not shown here), reflecting an uncritical positive opinion on vaccination. Only 3% feared that vaccines may expose their child to a substantial risk, suggesting minimal concerns over the side effects of vaccines. Nevertheless, one fifth reported that natural childhood disease may be preferable to vaccination, reflecting perceptions of less susceptibility to and severity of vaccine preventable diseases. More skepticism was observed regarding new vaccines and accuracy of recommendations with approximately one third feeling confused due to conflicting physicians' opinions, almost half stating the need to cross-check doctors' recommendations and about 60% believing that new vaccines are developed for financial profit (data not shown here). Despite these statements, however, parents/guardians indicated a high degree of compliance with the current official recommendations on vaccination, with almost all stating that they would keep their child's immunization up-to-date, following the current scientific advice (data not shown).

Among all respondents, cost (including the cost of the visit to an immunization provider, the fee for vaccine administration and the cost of the vaccine itself) was the barrier most commonly identified. About one fifth of respondents cited long distance to vaccination location and/or inconvenient opening hours of the immunization sites as important barriers to immunization. Other obstacles less commonly reported (not shown here) included lack of accurate information regarding immunization, unfriendly behavior of immunization staff or bad organization of services, lack of confidence in social insurance doctors, lack of paediatricians and long waiting times for appointments.

## Task 5.2 - Identify unadjusted determinants of vaccination coverage

Continue with the previous dataset.

* Calculate the appropriate measures of association (and corresponding 95%CI) between factors presented in table 5.2 and complete vaccination. Complete Table 5.2. The measures of association should allow for the weights, stratification and clustering. Interpret the results. You may use svyglm function.

## Help, Task 5.2

Calculate prevalence ratios for each of the variables in table 5.2. To adjust for the sampling design, we must specify the design prior to running any regression models.

design <- svydesign(ids = ~school, weights = ~weight, strata = ~strata, data = vaccine)

To calculate weighted proportions and PRs from the weighted proportions, you may use the **svyglm** function.

More information can be found in R help, but the essentials are summarised below:

*svyglm(formula, design, family = NULL, data = NULL)*

* **formula**: designates the model formula (outcome and predictor variables to include)
* **design**: the object created using svydesign
* **family**: specifies the distribution of dependent variable,the link specifies the link function; the default is the canonical link for the family() specified.
* **data**: specifies the relevant data frame

For binomial and Poisson families use **family=quasibinomial() and family=quasipoisson()** to avoid a warning about non-integer numbers of successes. The 'quasi' versions of the family objects give the same point estimates and standard errors and do not give the warning.

Below is an example using a3x4g as a predictor variable.

model <- svyglm(vacful ~a3x4g,   
 data = vaccine,  
 family = quasibinomial(link = "log"),  
 design = design)  
  
# The model is exponentiated in order to obtain the PRs  
 model1 <- tidy(model, exponentiate = TRUE, conf.int = TRUE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.64633 | 0.02417 | -18.05616 | 0.00000 | 0.61642 | 0.67769 |
| a3x4g | 0.98846 | 0.02918 | -0.39793 | 0.69095 | 0.93351 | 1.04663 |

PR=0.98 suggests that the complete vaccination uptake was almost equal among children whose parents/guardians perceived cost of the vaccines as a major barrier to immunization and those who did not.

#### Calculate the appropriate measures of association (and corresponding 95% CI) between factors presented in table 5.2 and complete vaccination status.

This can be done using a loop as below:

# Specify the variables of interest  
vars <- c("minority", "mageg", "educf", "osibg",   
 "a1posyn", "a1posg", "uncritical",   
 "a1x1g", "a1x2g", "a1x5g", "a3x1g", "a3x2g", "a3x5g")  
  
# Create a loop to store the output  
output3 <- list()  
  
# Run svglm for each variable  
for (var in vars) {  
   
 form <- formula(paste0("vacful ~ factor(",var,")"))  
   
 model2 <- svyglm(form,  
 data = vaccine,  
 family = quasibinomial(link = "log"),  
 design = design)  
   
 model3 <- tidy(model2, exponentiate = TRUE, conf.int = TRUE)  
   
 output3[[var]] <- model3  
}

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.671 | 0.019 | -20.708 | 0.000 | 0.646 | 0.696 |
| factor(minority)1 | 0.597 | 0.250 | -2.057 | 0.041 | 0.366 | 0.976 |
| factor(minority)2 | 0.787 | 0.118 | -2.034 | 0.043 | 0.625 | 0.991 |
| factor(minority)3 | 0.446 | 0.094 | -8.564 | 0.000 | 0.371 | 0.537 |
| (Intercept) | 0.386 | 0.230 | -4.143 | 0.000 | 0.246 | 0.606 |
| factor(mageg)2 | 1.367 | 0.242 | 1.294 | 0.196 | 0.851 | 2.195 |
| factor(mageg)3 | 1.703 | 0.231 | 2.304 | 0.022 | 1.083 | 2.680 |
| (Intercept) | 0.559 | 0.051 | -11.365 | 0.000 | 0.506 | 0.618 |
| factor(educf)3 | 1.046 | 0.065 | 0.684 | 0.494 | 0.920 | 1.189 |
| factor(educf)4 | 1.155 | 0.055 | 2.624 | 0.009 | 1.037 | 1.286 |
| factor(educf)6 | 1.260 | 0.056 | 4.103 | 0.000 | 1.128 | 1.407 |
| (Intercept) | 0.486 | 0.083 | -8.727 | 0.000 | 0.413 | 0.571 |
| factor(osibg)1 | 1.343 | 0.086 | 3.429 | 0.001 | 1.135 | 1.589 |
| factor(osibg)3 | 1.347 | 0.086 | 3.449 | 0.001 | 1.137 | 1.595 |
| (Intercept) | 0.523 | 0.153 | -4.236 | 0.000 | 0.387 | 0.706 |
| factor(a1posyn)1 | 1.233 | 0.154 | 1.360 | 0.175 | 0.912 | 1.669 |
| (Intercept) | 0.523 | 0.153 | -4.236 | 0.000 | 0.387 | 0.706 |
| factor(a1posg)1 | 1.140 | 0.157 | 0.835 | 0.404 | 0.838 | 1.552 |
| factor(a1posg)2 | 1.253 | 0.155 | 1.458 | 0.146 | 0.925 | 1.697 |
| (Intercept) | 0.652 | 0.021 | -19.922 | 0.000 | 0.625 | 0.680 |
| factor(uncritical)1 | 0.967 | 0.029 | -1.168 | 0.244 | 0.914 | 1.023 |
| (Intercept) | 0.511 | 0.561 | -1.198 | 0.232 | 0.170 | 1.533 |
| factor(a1x1g)1 | 1.253 | 0.562 | 0.402 | 0.688 | 0.417 | 3.768 |
| (Intercept) | 0.644 | 0.021 | -21.196 | 0.000 | 0.619 | 0.671 |
| factor(a1x2g)1 | 0.851 | 0.102 | -1.584 | 0.114 | 0.697 | 1.039 |
| (Intercept) | 0.656 | 0.021 | -20.537 | 0.000 | 0.630 | 0.683 |
| factor(a1x5g)1 | 0.893 | 0.040 | -2.810 | 0.005 | 0.825 | 0.966 |
| (Intercept) | 0.660 | 0.020 | -20.750 | 0.000 | 0.634 | 0.686 |
| factor(a3x1g)1 | 0.849 | 0.039 | -4.156 | 0.000 | 0.786 | 0.917 |
| (Intercept) | 0.653 | 0.021 | -20.662 | 0.000 | 0.627 | 0.680 |
| factor(a3x2g)1 | 0.912 | 0.044 | -2.102 | 0.036 | 0.838 | 0.994 |
| (Intercept) | 0.644 | 0.023 | -19.201 | 0.000 | 0.616 | 0.673 |
| factor(a3x5g)1 | 0.991 | 0.031 | -0.273 | 0.785 | 0.932 | 1.055 |

**Summarizing the results**

* Socioeconomic factors  
  Children were less likely to be completely vaccinated if they belonged to a minority group, were born to a younger mother, had many siblings or their parents/guardians were less educated (Table 1). The weighted proportions of children with complete vaccination among households with 3 or more children were 26% and 38% lower, respectively, compared with those of households with only one child.
* Beliefs and attitudes towards immunization  
  Children whose parents had a positive opinion regarding immunization had 23% and 16% higher complete vaccination coverage, respectively. However, these associations were not statistically significant. Complete vaccination coverage was 11% lower in children of parents/guardians reporting that natural childhood disease may be preferable to vaccination.
* Barriers to immunization  
  Complete vaccination was lower in children of parents/guardians who perceived long distance to immunization site, as a major barrier to immunization. Children whose parents/guardians reported inconvenient opening hours as a barrier to vaccination were less likely to be completely vaccinated.

# Session 6 - Multivariable analysis

# An introduction to the R companion

This text was adapted from the introduction used at the 2016 TSA module.

R packages are bundles of functions which extend the capability of R. Thousands of add-on packages are available in the main online repository (known as CRAN) and many more packages in development can be found on GitHub. They may be installed and updated over the Internet.

We will mainly use packages which come ready installed with R (base code), but where it makes things easier we will use add-on packages. In addition, we have included a few extra functions to simplify the code required. All the R packages you need for the exercises can be installed over the Internet.

# Installing required packages for the week  
required\_packages <- c("foreign", "broom", "leaps", "survey")  
  
install.packages(required\_packages)

Run the following code at the beginning of the case study to make sure that you have made available all the packages and functions that you need. Be sure to include it in any scripts too.

# Loading required packages for the week  
required\_packages <- c("foreign", "broom", "leaps", "survey")  
  
for (i in seq(along = required\_packages))  
 library(required\_packages[i], character.only = TRUE)

# Function to make tables with counts and proportions   
big.table <- function(data) {  
 count <- table(data)  
 prop <- round(prop.table(count)\*100, digits = 2)  
 cbind(count,  
 prop)   
}

R and Stata have minor differences in default settings and methods. In this document we will follow the Stata analysis as closely as possible, but small and usually unimportant differences may be noted between the statistical findings in R and those in Stata. Despite the differences, the findings from each statistical package should be comparable. At some points additional steps (which would usually be optional in R) will be taken to produce output which is comparable to that of Stata.

The big.table function uses data directly and combines counts and proportions, thus reducing the number of lines of code required for descriptive analyses.

You will work with Stata.dta data sets which can be loaded into R with the "foreign" or "readstata13" packages. The appropriate functions to use will be indicated.

R can hold one or many data sets in memory simultaneously, so there is usually no need to save intermediate files or close and re-open datasets.

# Session 6 - Multivariable analysis

Start a new R script, name it, for example, **session6.r** and save it in your working directory. Write all commands in the R script so that you can run (and re-run) it when needed during the exercise.

Open the **vaccine6.dta** dataset

vacc <- read.dta("vaccine6.dta", convert.factors = FALSE)

## 

## Task 6.1 Logistic regression

#### Treat minority as a categorical variable and change the reference group

As seen in the MVA module, for logit and logistic regression, we use glm with a "logit" link. We obtain ORs by exponentiating the output of the logit model. Minority is treated as a factor variable in the model by writing **factor** before the variable.

model <- glm(vacful~factor(minority),  
 data = vacc,  
 family = binomial(link = "logit"))  
  
modelop <- tidy(model, exponentiate = TRUE, conf.int = TRUE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 1.94683 | 0.03934 | 16.93411 | 0e+00 | 1.80292 | 2.10357 |
| factor(minority)1 | 0.19262 | 0.39284 | -4.19260 | 3e-05 | 0.08444 | 0.40180 |
| factor(minority)2 | 0.54387 | 0.13374 | -4.55379 | 1e-05 | 0.41844 | 0.70725 |
| factor(minority)3 | 0.23684 | 0.14326 | -10.05455 | 0e+00 | 0.17804 | 0.31242 |

The level of reference for minority is set to **0** by R.

# Here, we see the 4 levels of minority  
big.table(vacc$minority)

## count prop  
## 0 2896 84.58  
## 1 33 0.96  
## 2 245 7.16  
## 3 250 7.30

We can change the reference level of minority in a number of ways. See below for an example.

# Create a new variable where the reference is the 4th level of the minority variable  
vacc$minority2 <- relevel(factor(vacc$minority), ref = 4)

#### 

#### Use the anova test to see if independent variables (e.g. minority groups (minority), gender (gender), other siblings in the family (osibling) contribute significantly to the model.

If we want to compare two models, we will use the **anova** test, which tests for the difference in the residual deviances between the models. This is equivalent to the likelihood ratio test in Stata. An important point when comparing two models is that they should both contain the same number of observations.

First run a model with the outcome (vacful) and three independent variables (minority, osibling and gender).

# Note use of the minority2 variable here   
model2 <- glm(vacful~factor(minority2) + osibling + gender,  
 data = vacc,  
 family = binomial(link = "logit"))  
  
model2op <- tidy(model2, exponentiate = TRUE, conf.int = TRUE)  
  
# Number of observations in the model:   
nobs(model2)

## [1] 3299

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.37163 | 0.16280 | -6.08039 | 0.00000 | 0.26879 | 0.50919 |
| factor(minority2)0 | 4.49609 | 0.15019 | 10.00886 | 0.00000 | 3.36366 | 6.06550 |
| factor(minority2)1 | 0.74181 | 0.45806 | -0.65202 | 0.51439 | 0.28190 | 1.74248 |
| factor(minority2)2 | 2.64229 | 0.19869 | 4.89034 | 0.00000 | 1.79495 | 3.91365 |
| osibling | 1.13930 | 0.04673 | 2.79072 | 0.00526 | 1.04015 | 1.24936 |
| gender | 1.02130 | 0.09309 | 0.22640 | 0.82089 | 0.85164 | 1.22684 |

Then, we run a reduced model (removing one of the independent variables, e.g. gender). The reduced model (i.e. the model without the dropped variable) will contain more observations than the full model (model2). To overcome this, we subset our dataset so that it has the same number of observations as in model2. See https://theesspreckelsen.wordpress.com/2016/08/10/estimation-sample-information-from-linear-regression-in-r-using-lm-aka-statas-esample/ for more information.

# This creates a matrix out of the residuals of each observation of model2  
esample <- rownames(as.matrix(resid(model2)))  
  
# We then subset the vacc dataset with esample to have the same number of observations as model2  
model3 <- glm(vacful~factor(minority2) + osibling,  
 data = vacc[esample,],  
 family = binomial(link = "logit"))  
  
model3op <- tidy(model3, exponentiate = TRUE, conf.int = TRUE)  
  
# Number of observations in the model:   
nobs(model3)

## [1] 3299

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.37411 | 0.16009 | -6.14149 | 0.00000 | 0.27200 | 0.50985 |
| factor(minority2)0 | 4.48343 | 0.14966 | 10.02520 | 0.00000 | 3.35769 | 6.04231 |
| factor(minority2)1 | 0.74070 | 0.45802 | -0.65534 | 0.51225 | 0.28150 | 1.73972 |
| factor(minority2)2 | 2.65946 | 0.19663 | 4.97447 | 0.00000 | 1.81399 | 3.92341 |
| osibling | 1.13916 | 0.04673 | 2.78840 | 0.00530 | 1.04003 | 1.24919 |

Finally, perform the **anova** test to compare the 2 models.

anova(model2, model3, test = "Chisq")

## Analysis of Deviance Table  
##   
## Model 1: vacful ~ factor(minority2) + osibling + gender  
## Model 2: vacful ~ factor(minority2) + osibling  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 3293 4220.4   
## 2 3294 4220.5 -1 -0.051313 0.8208

The p-value of 0.8208 suggests that the 2 models are not significantly different (p-value > 0.05). So, we can drop the variable gender, as it does not contribute significantly to the model and keep the simpler (reduced) model with only minority and osibling as the independent variables.

#### Perform logistic regression using the weights and allowing for the clustering, using the command svyglm and regTermTest to perform Wald tests.

The logistic regression you have done so far ignores the sampling weights and the clustering in the sample. To allow for the weights and the clustering you can use the svyglm command in R. However, if you use weights and/or allow for the clustering any anova test will be invalid. In this case, you just have to use the approximate significance Wald tests that R gives in the regression table. Alternatively, to test variables with more than one parameter after the regression (svyglm), you can use **regTermTest**. This provides Wald test and working likelihood ratio (Rao-Scott) test of the hypothesis that all coefficients associated with a particular regression term are zero (or have some other specified values).

The use of the **quasibinomial** family with svyglm is recommended rather than binomial in order to avoid obtaining the error message about non-integer numbers of successes. The “quasi” versions of the family objects give the same point estimates and standard errors and do not give the warning.

For example:

# First,set the design to include clustering, stratification and weights  
design <- svydesign(ids = ~school, weights = ~weight, strata = ~strata, data = vacc)

model4 <- svyglm(vacful~factor(minority2) + factor(osibling) + factor(mageg) + factor(educf),  
 data = vacc,  
 family = quasibinomial(link = "logit"),  
 design = design)  
model4op <- tidy(model4, exponentiate = T, conf.int = T)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.05914 | 0.50867 | -5.55914 | 0.00000 | 0.02182 | 0.16028 |
| factor(minority2)0 | 4.49021 | 0.17117 | 8.77454 | 0.00000 | 3.21048 | 6.28005 |
| factor(minority2)1 | 2.35404 | 0.43916 | 1.94948 | 0.05215 | 0.99541 | 5.56708 |
| factor(minority2)2 | 3.59220 | 0.27332 | 4.67865 | 0.00000 | 2.10238 | 6.13777 |
| factor(osibling)1 | 1.96972 | 0.17805 | 3.80728 | 0.00017 | 1.38946 | 2.79230 |
| factor(osibling)3 | 2.21759 | 0.19215 | 4.14487 | 0.00004 | 1.52169 | 3.23173 |
| factor(mageg)2 | 2.72137 | 0.46948 | 2.13242 | 0.03376 | 1.08433 | 6.82990 |
| factor(mageg)3 | 3.38342 | 0.44514 | 2.73818 | 0.00654 | 1.41400 | 8.09588 |
| factor(educf)3 | 1.03131 | 0.15944 | 0.19334 | 0.84682 | 0.75452 | 1.40964 |
| factor(educf)4 | 1.23840 | 0.13985 | 1.52893 | 0.12731 | 0.94150 | 1.62892 |
| factor(educf)6 | 1.50828 | 0.14790 | 2.77872 | 0.00579 | 1.12873 | 2.01546 |

Now assess those categorical variables using regTermTest.

regTermTest(model4, "factor(minority2)")

## Wald test for factor(minority2)  
## in svyglm(formula = vacful ~ factor(minority2) + factor(osibling) +   
## factor(mageg) + factor(educf), data = vacc, family = quasibinomial(link = "logit"),   
## design = design)  
## F = 25.74959 on 3 and 308 df: p= 6.8625e-15

The p-value suggests that **minority** contributes significantly to the model and cannot be dropped.

regTermTest(model4, "factor(osibling)")

## Wald test for factor(osibling)  
## in svyglm(formula = vacful ~ factor(minority2) + factor(osibling) +   
## factor(mageg) + factor(educf), data = vacc, family = quasibinomial(link = "logit"),   
## design = design)  
## F = 9.257961 on 2 and 308 df: p= 0.00012461

regTermTest(model4, "factor(mageg)")

## Wald test for factor(mageg)  
## in svyglm(formula = vacful ~ factor(minority2) + factor(osibling) +   
## factor(mageg) + factor(educf), data = vacc, family = quasibinomial(link = "logit"),   
## design = design)  
## F = 5.239839 on 2 and 308 df: p= 0.005784

regTermTest(model4, "factor(educf)")

## Wald test for factor(educf)  
## in svyglm(formula = vacful ~ factor(minority2) + factor(osibling) +   
## factor(mageg) + factor(educf), data = vacc, family = quasibinomial(link = "logit"),   
## design = design)  
## F = 3.985442 on 3 and 308 df: p= 0.0082942

#### 

#### Work your way towards a final model (the simplest adequate model). Include in your initial model all the variables of Table 5.2 (session 5). Use vacful as the outcome variable.

Your final model should look like this:

finalmodel <- svyglm(vacful~factor(minority2) + factor(osibling) + factor(mageg) + a1x5g + a3x1g,  
 data = vacc,  
 family = quasibinomial(link = "logit"),  
 design = design)  
  
finalmodelop <- tidy(finalmodel, exponentiate = TRUE, conf.int = TRUE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.06668 | 0.58157 | -4.65613 | 0.00000 | 0.02133 | 0.20846 |
| factor(minority2)0 | 4.64546 | 0.17513 | 8.77011 | 0.00000 | 3.29579 | 6.54784 |
| factor(minority2)1 | 1.62013 | 0.37652 | 1.28150 | 0.20098 | 0.77456 | 3.38879 |
| factor(minority2)2 | 3.01624 | 0.26549 | 4.15838 | 0.00004 | 1.79258 | 5.07517 |
| factor(osibling)1 | 2.20004 | 0.18504 | 4.26117 | 0.00003 | 1.53083 | 3.16181 |
| factor(osibling)3 | 2.38825 | 0.20419 | 4.26345 | 0.00003 | 1.60056 | 3.56359 |
| factor(mageg)2 | 2.64095 | 0.54302 | 1.78840 | 0.07469 | 0.91105 | 7.65561 |
| factor(mageg)3 | 3.69059 | 0.52119 | 2.50541 | 0.01275 | 1.32880 | 10.25013 |
| a1x5g | 0.76154 | 0.10189 | -2.67365 | 0.00790 | 0.62369 | 0.92987 |
| a3x1g | 0.83025 | 0.11077 | -1.67934 | 0.09410 | 0.66822 | 1.03157 |

## 

## Task 6.2 Binomial regression

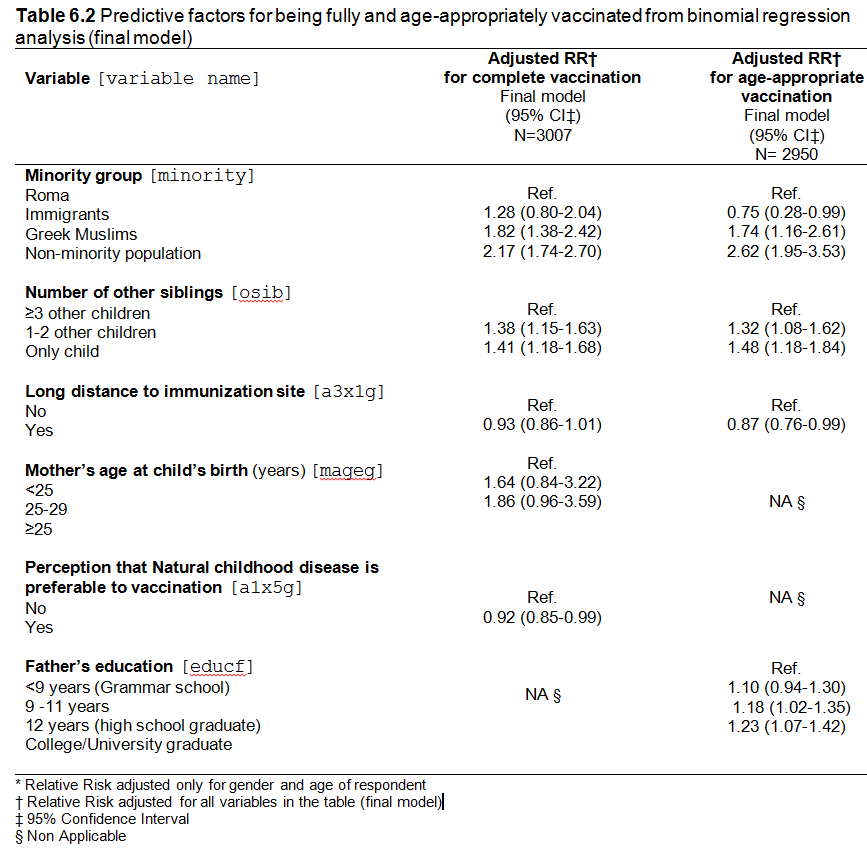
#### Generalised linear model to estimate relative risks using sampling weights

The measure of effect of choice in this study was PR and not OR. To estimate adjusted PR and using sampling weights, you should use a generalized linear model (with a log link function). You may use the svglm command with log link and vacful as the outcome variable.

# Use the previously defined final model  
bin1 <- svyglm(vacful~factor(minority2) + factor(osibling) + factor(mageg) + a1x5g + a3x1g,  
 data = vacc,  
 family = quasibinomial(link = "log"),  
 design = design)  
  
bin1op <- tidy(bin1, exponentiate = TRUE, conf.int = TRUE)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| term | estimate | std.error | statistic | p.value | conf.low | conf.high |
| (Intercept) | 0.12956 | 0.36784 | -5.55573 | 0.00000 | 0.06300 | 0.26643 |
| factor(minority2)0 | 2.16929 | 0.11406 | 6.78959 | 0.00000 | 1.73473 | 2.71270 |
| factor(minority2)1 | 1.28010 | 0.23688 | 1.04247 | 0.29801 | 0.80466 | 2.03646 |
| factor(minority2)2 | 1.82649 | 0.14473 | 4.16228 | 0.00004 | 1.37539 | 2.42554 |
| factor(osibling)1 | 1.37432 | 0.08936 | 3.55810 | 0.00043 | 1.15351 | 1.63740 |
| factor(osibling)3 | 1.40673 | 0.09082 | 3.75779 | 0.00020 | 1.17736 | 1.68079 |
| factor(mageg)2 | 1.64636 | 0.34533 | 1.44377 | 0.14982 | 0.83673 | 3.23942 |
| factor(mageg)3 | 1.85732 | 0.33751 | 1.83444 | 0.06755 | 0.95852 | 3.59892 |
| a1x5g | 0.91609 | 0.03702 | -2.36716 | 0.01854 | 0.85197 | 0.98504 |
| a3x1g | 0.93142 | 0.04213 | -1.68638 | 0.09273 | 0.85760 | 1.01159 |

The completed table 6.2 should look like this:



# Conclusions

* **Summarising the results**

This study identified several independent determinants of complete and age-appropriate vaccination status among 6-year old school children in Greece. Both complete and age-appropriate vaccination coverage were significantly lower among children who belonged to a minority group, who had siblings in the household and whose parents/guardians perceived long distance to the place of immunization as a barrier to vaccination. In addition, maternal age <30 years and the perception that natural disease is preferable to vaccination were significantly associated with lower rates of complete vaccination. Furthermore, low paternal education was an additional predictor of delayed vaccination. These findings indicate that socioeconomic factors may be more important determinants of immunization coverage than parental perceptions.

* **Recommendations**
* Based on the findings of this study, the following identified groups merit increased attention in future interventions designed to improve immunization delivery in Greece:
  + minority groups, especially Roma and immigrants,
  + families with many children,
  + young mothers, and
  + households headed by fathers with low educational level, possible reflecting low socioeconomic status of the families.
* Interventions aimed at those high-risk families, although difficult to deliver, may have the greatest effects on community immunization rates.
* In addition, parents/guardians must be educated about the hazards of diseases that vaccinations prevent and the likely susceptibility of their children to these diseases.
* There is a need for policies to overcome structural and health care system barriers to immunization, and identify effective and comprehensive approaches for improving the immunization levels of children in high-risk groups.