UN WOMEN | WOMEN COUNT

COUNTED & VISIBLE TOOLKIT

Computation of selected gender-specific and gender-relevant Sustainable Development Goals indicators in selected countries using R

THE INTER-SECRETARIAT
WORKING GROUP ON
HOUSEHOLD SURVEYS

About the Report

The Counted and Visible: Toolkit to Better Utilize Existing Data from Household Surveys to Generate Disaggregated Gender Statistics (Counted and Visible Toolkit) provides a compilation of tools and mechanisms used by several countries to produce evidence to inform gender-responsive policies and catalyze actions to leave no one behind. This publication was developed by UN Women, in collaboration with the Inter-Secretariat Working Group on Household Surveys (ISWGHS). From the technical to the strategic, the publication also takes holistic approach to production of disaggregated gender statistics showcasing good country practices – from the commitment of national statistical systems to uptake and use. The digital version of the publication can be found at: https://data.unwomen.org/resources/counted-and-visible-toolkit.

This publication, a complementary technical resource of the Toolkit, is a comprehensive compilation of statistical computing exercises via R to produce disaggregated gender statistics covering 13 Sustainable Development Goals indicators using publicly available datasets of select countries

Acknowledgements

Chief of the Research and Data Section: Papa Seck

Report lead: Jessamyn Encarnacion

Report coordinators: Alper Gucumengil, Rea Jean Tabaco

Statistics: Zita Albacea, Ramya Emandi, Ghida Ismail, Rea Jean Tabaco

Production, communication and outreach: Lauren Billi, Jacqueline Gordillo, Mika Mansukhani, Samrat Maskey, Svenja Siemonsen

Cover Design: Ouissal Hmazzou

The views expressed in this publication are those of the authors and do not necessarily represent the views of UN Women, the United Nations or any of its affiliated organizations. For a list of any errors or omissions found subsequent to the report launch please visit our website.

NOTE TO USER

Software

This technical resource provides a step-by-step guide in producing and assessing disaggregated gender statistics using R. The scripts were produced using R version 4.1.1. However, the commands used were basic codes and will work in earlier or later versions of the statistical software.

Indicators

Disaggregated gender statistics were produced for 13 Sustainable Development Goals (SDG) indicators – one for each Goal and one for each SDG 5 target – data permitting. In some cases where no gender-specific or gender-relevant indicators are available, a related indicator was computed instead.

Additional codes are provided to disaggregate the indicator by wealth index and/or by type of location (urban/rural).

Dataset

The disaggregated gender statistics were computed using the latest Demographic and Health Survey (DHS) data of selected countries, mainly using the individual data file for women. DHS data are publicly available upon request via https://dhsprogram.com/. Weights in DHS data files need to be divided by 1,000,000 as they were computed to six decimal points but presented in the standard recode files without the decimal point.

In one indicator, the data for Multiple Indicator Survey (MICS) was used. MICS datasets can be downloaded for free via https://mics.unicef.org/surveys after registration.

Country selection

In generating relevant disaggregated gender statistics, mainly the datasets and variables of UN Women's Women Count programme pathfinder countries or other supported countries by the programme have been used.

Standard error (SE) and Coefficient of Variation (CV)

The codes will directly produce SE and CV which are used to assess the quality (particularly precision and reliability) of disaggregated gender statistics produced.

Low value of SE is preferred. For CV, there are no internationally agreed standards or recommendation as thresholds vary country to country and surveys to surveys. The Counted and Visible Toolkit suggests this classification of estimates:

VALUE OF CV	SUGGESTED CLASSIFICATION OF ESTIMATES (x)
x ≤ 10%	Highly reliable
10% > x ≥ 20%	Sufficiently reliable
20% > x ≥ 33%	Still acceptable but should be used with caution.
x > 33%	Caveats should be provided in terms of the level of reliability of the estimate.

LIST OF INDICATORS

Goal 1. End poverty in all its forms everywhere & Goal 10. Reduce inequality within and among countries

(Related Indicator) Proportion of women who belong to the poorest 20% of the population, by age and persons with disability

Goal 2. End hunger, achieve food security and improved nutrition and promote sustainable agriculture

2.2.3 Prevalence of anaemia in women aged 15 to 49 years, by pregnancy status

Goal 3. Ensure healthy lives and promote well-being for all at all ages

3.1.2 Proportion of births attended by skilled health personnel

3.7.1 Proportion of women of reproductive age (aged 15–49 years) who have their need for family planning satisfied with modern methods

Goal 4. Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all

(Related Indicator) Proportion of women with primary or less education

Goal 5. Achieve gender equality and empower all women and girls

5.2.1 Proportion of ever-partnered women and girls aged 15 years and older subjected to physical, sexual or psychological violence by a current or former intimate partner in the previous 12 months, by form of violence and by age

5.3.1 Proportion of women aged 20–24 years who were married or in a union before age 15 and before age 18

5.6.1 Proportion of women aged 15–49 years who make their own informed decisions regarding contraceptive use and reproductive health care1

5.b.1 Proportion of women who own a mobile telephone[2]

Goal 7. Ensure access to affordable, reliable, sustainable and modern energy for all

7.1.2 Proportion of women with primary reliance on clean fuels and technology[3]

Goal 8. Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

8.10.2 Proportion of women (15 years and older) with an account at a bank or other financial institution or with a mobile-money-service provider[4]

Goal 16. Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels

16.2.3 Proportion of young women aged 18–29 years who experienced sexual violence by age 18

Goal 17. Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development

17.8.1 Proportion of women using the Internet in the last 12 months[5]

[1] Full SDG indicator: Proportion of women aged 15–49 years who make their own informed decisions regarding sexual relations, contraceptive use and reproductive health care.

[2] Full SDG indicator: Proportion of individuals who own a mobile telephone, by sex

[3] Full SDG indicator: Proportion of population with primary reliance on clean fuels and technology

[4] Full SDG indicator: Proportion of adults (15 years and older) with an account at a bank or other financial institution or with a mobile-money-service provider

[5] Full SDG indicator: Proportion of individuals using the Internet in the last three months.

COUNTRIES AND DATA SOURCES

SDG	COUNTRY	DATASET	DATA FILE
1.x	Albania	Demographic and Health Survey 2017-18	ALIR71FL
2.2.3	Nepal	Demographic and Health Survey 2016	NPIR7HFL
3.1.2	Sierra Leone	Demographic and Health Survey 2019	SLBR7AFL
3.7.1	Uganda	Demographic and Health Survey 2016	UGIR7BFL
4.x	Mongolia	Multiple Indicator Cluster Survey 2018	wm
5.2.1	Tajikistan	Demographic and Health Survey 2017	TJIR71FL
5.3.1	Egypt	Demographic and Health Survey 2014	EGIR61FL
5.6.1	Kyrgyz Republic	Demographic and Health Survey 2012	KYIR61FL
5.b.1	Jordan	Demographic and Health Survey 2017-2018	JOIR73FL
7.1.2	Bangladesh	Demographic and Health Survey 2017-2018	BDIR7RFL
8.10.2	Tanzania	Demographic and Health Survey 2015-2016	TZIR7BFL
16.2.3	Senegal	Demographic and Health Survey 2019	SNIR8BFL
17.8.1	Cameroon	Demographic and Health Survey 2018	CMIR71FL

R CODES

Goal 1. End poverty in all its forms everywhere & Goal 10. Reduce inequality within and among countries

(Related Indicator) 1.x Proportion of women who belong to the poorest 20% of the population, by age and persons with disability

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    Albania_SDG1 <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/ALIR71FL.dta")
    Albania_SDG1$v005 <- Albania_SDG1$v005/1000000

###Step 2: Generate needed variables
#Poorest
    Albania_SDG1$poorest <- ifelse(Albania_SDG1$v190==1,1,0)
    Albania_SDG1$richest <- ifelse(Albania_SDG1$v190==5,1,0)

#Disability
    attach(Albania_SDG1)
    Albania_SDG1$disability[s1105==1] <- 1
    Albania_SDG1$disability[s1105==0] <- 0
    detach(Albania_SDG1)

#Interactive term

    attach(Albania_SDG1)
    Albania_SDG1$disability_poorest[disability==1 & poorest==0] <- 0
    Albania_SDG1$disability_poorest[poorest==1 & disability==1] <- 1
    detach(Albania_SDG1)

    attach(Albania_SDG1)
    Albania_SDG1$urban_poorest[v025==1 & poorest==0] <- 0
    Albania_SDG1$urban_poorest[poorest==1 & v025==1] <- 1
    detach(Albania_SDG1)

    attach(Albania_SDG1)
    Albania_SDG1$rural_poorest[v025==2 & poorest==0] <- 0
    Albania_SDG1$rural_poorest[poorest==1 & v025==2] <- 1
    detach(Albania_SDG1)
```

```
Albania_SDG1$age1 <- ifelse(Albania_SDG1$v013==1,1,0)
attach(Albania_SDG1)
Albania_SDG1$age1_poorest[age1==1 & poorest==0] <- 0
Albania_SDG1$age1_poorest[poorest==1 & age1==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age2 <- ifelse(Albania_SDG1$v013==2,1,0)
attach(Albania_SDG1)
Albania_SDG1$age2_poorest[age2==1 & poorest==0] <- 0
Albania_SDG1$age2_poorest[poorest==1 & age2==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age3 <- ifelse(Albania_SDG1$v013==3,1,0)
attach(Albania_SDG1)
Albania_SDG1$age3_poorest[age3==1 & poorest==0] <- 0
Albania_SDG1$age3_poorest[poorest==1 & age3==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age4 <- ifelse(Albania_SDG1$v013==4,1,0)
attach(Albania_SDG1)
Albania_SDG1$age4_poorest[age4==1 & poorest==0] <- 0
Albania_SDG1$age4_poorest[poorest==1 & age4==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age5 <- ifelse(Albania_SDG1$v013==5,1,0)
attach(Albania_SDG1)
Albania_SDG1$age5_poorest[age5==1 & poorest==0] <- 0
Albania_SDG1$age5_poorest[poorest==1 & age5==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age6 <- ifelse(Albania_SDG1$v013==6,1,0)
attach(Albania_SDG1)
Albania_SDG1$age6_poorest[age6==1 & poorest==0] <- 0
Albania_SDG1$age6_poorest[poorest==1 & age6==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age7 <- ifelse(Albania_SDG1$v013==7,1,0)
attach(Albania_SDG1)
Albania_SDG1$age7_poorest[age7==1 & poorest==0] <- 0
Albania_SDG1$age7_poorest[poorest==1 & age7==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age8 <- ifelse(Albania_SDG1$v013==8,1,0)
attach(Albania_SDG1)
Albania_SDG1$age8_poorest[age8==1 & poorest==0] <- 0
Albania_SDG1$age8_poorest[poorest==1 & age8==1] <- 1
detach(Albania_SDG1)

Albania_SDG1$age9 <- ifelse(Albania_SDG1$v013==9,1,0)
attach(Albania_SDG1)
```

```
    Albania_SDG1$age9_poorest[age9==1 & poorest==0] <- 0
    Albania_SDG1$age9_poorest[poorest==1 & age9==1] <- 1
    detach(Albania_SDG1)

# Label Variables
    library(sjlabelled)
    Albania_SDG1$poorest <- to_factor(Albania_SDG1$poorest)
    levels(Albania_SDG1$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Albania_SDG1$richest <- to_factor(Albania_SDG1$richest)
    levels(Albania_SDG1$richest) <- list("Richest" = "1", "Not richest"="0")
    Albania_SDG1$v025 <- to_factor(Albania_SDG1$v025)
    levels(Albania_SDG1$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c(“disability_poorest”, "urban_poorest", "rural_poorest", “age1_poorest”, “age2_poorest”,
    “age3_poorest”, “age4_poorest”, “age5_poorest”, “age6_poorest”, “age7_poorest”, “age8_poorest”,
    “age9_poorest”)

     for (var in varlist4) {
        Albania_SDG1[[var]] <- to_factor(Albania_SDG1[[var]])
        levels(Albania_SDG1[[var]]) <- list("Yes" = "1", "No"="0")
        }
```

***Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = Albania_SDG1, variable = poorest, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Albania_SDG1, x = s1105, y = poorest, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Albania_SDG1, x = v025, y = poorest, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Albania_SDG1, x = v013, y = poorest, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = Albania_SDG1)

#Generate means and SE
    mean_poorest <- svymean(~factor(poorest), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_poorest), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_poorest), sample, na.rm=T)
    mean_disability <- svymean(~factor(disability_poorest), sample, na.rm=T)
    mean_age1 <- svymean(~factor(age1_poorest), sample, na.rm=T)
    mean_age2 <- svymean(~factor(age2_poorest), sample, na.rm=T)
    mean_age3 <- svymean(~factor(age3_poorest), sample, na.rm=T)
    mean_age4 <- svymean(~factor(age4_poorest), sample, na.rm=T)
    mean_age5 <- svymean(~factor(age5_poorest), sample, na.rm=T)
    mean_age6 <- svymean(~factor(age6_poorest), sample, na.rm=T)
    mean_age7 <- svymean(~factor(age7_poorest), sample, na.rm=T)
    mean_age8 <- svymean(~factor(age8_poorest), sample, na.rm=T)
    mean_age9 <- svymean(~factor(age9_poorest), sample, na.rm=T)

#Generate cv
    cv(mean_poorest)*100
    cv(mean_urban)*100
    cv(mean_rural)*100
    cv(mean_disability)*100
    cv(mean_age1)*100
    cv(mean_age2)*100
    cv(mean_age3)*100
    cv(mean_age4)*100
    cv(mean_age5)*100
    cv(mean_age6)*100
    cv(mean_age7)*100
    cv(mean_age8)*100
    cv(mean_age9)*100
```

Goal 2. End hunger, achieve food security and improved nutrition and promote sustainable agriculture

2.2.3 Prevalence of anaemia in women aged 15 to 49 years, by pregnancy status (percentage)

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    Nepal_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/NPIR7HFL.dta")
    Nepal_data <- subset(Nepal_data, v012>=15 & v012 <= 49)

###Step 2: Generate needed variables
#Anemia
    attach(Nepal_data)
    Nepal_data$anemia[v457==4] <- 0
    Nepal_data$anemia[v457==1 | v457==2 | v457==3] <- 1
    Nepal_data$anemia[v457==NA] <- NA
    detach(Nepal_data)

#Wealth
    Nepal_data$poorest <- ifelse(Nepal_data$v190==1,1,0)
    Nepal_data$richest <- ifelse(Nepal_data$v190==5,1,0)

#Weights
    Nepal_data$v005 <- Nepal_data$v005/1000000

#Interactive term

    attach(Nepal_data)
    Nepal_data$pregnant_anemia[v213==1 & anemia==0] <- 0
    Nepal_data$pregnant_anemia[anemia==1 & v213==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$notpregnant_anemia[v213==0 & anemia==0] <- 0
    Nepal_data$notpregnant_anemia[anemia==1 & v213==0] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$urban_anemia[v025==1 & anemia==0] <- 0
    Nepal_data$urban_anemia[anemia==1 & v025==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$rural_anemia[v025==2 & anemia==0] <- 0
    Nepal_data$rural_anemia[anemia==1 & v025==2] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$poor_anemia[poorest==1 & anemia==0] <- 0
    Nepal_data$poor_anemia[anemia==1 & poorest==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$rich_anemia[richest==1 & anemia==0] <- 0
    Nepal_data$rich_anemia[anemia==1 & richest==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$urban_anemia_poor[v025==1 & anemia==0 & poorest==1] <- 0
    Nepal_data$urban_anemia_poor[anemia==1 & v025==1 & poorest==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$urban_anemia_rich[v025==1 & anemia==0 & richest==1] <- 0
    Nepal_data$urban_anemia_rich[anemia==1 & v025==1 & richest==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$rural_anemia_poor[v025==2 & anemia==0 & poorest==1] <- 0
    Nepal_data$rural_anemia_poor[anemia==1 & v025==2 & poorest==1] <- 1
    detach(Nepal_data)

    attach(Nepal_data)
    Nepal_data$rural_anemia_rich[v025==2 & anemia==0 & richest==1] <- 0
    Nepal_data$rural_anemia_rich[anemia==1 & v025==2 & richest==1] <- 1
    detach(Nepal_data)

# Label Variables
    library(sjlabelled)
    Nepal_data$poorest <- to_factor(Nepal_data$poorest)
    levels(Nepal_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Nepal_data$richest <- to_factor(Nepal_data$richest)
    levels(Nepal_data$richest) <- list("Richest" = "1", "Not richest"="0")
    Nepal_data$v025 <- to_factor(Nepal_data$v025)
    levels(Nepal_data$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("anemia", "pregnant_anemia", "notpregnant_anemia", "urban_anemia", "rural_anemia",
    "urban_anemia_rich", "urban_anemia_poor", "rural_anemia_rich", "rural_anemia_poor")

     for (var in varlist4) {
            Nepal_data[[var]] <- to_factor(Nepal_data[[var]])
        levels(Nepal_data[[var]]) <- list("Yes" = "1", "No"="0")
            }
```

***Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = Nepal_data, variable = anemia, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Nepal_data, x = v213, y = anemia, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Nepal_data, x = v025, y = anemia, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Nepal_data, x = poorest, y = anemia, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Nepal_data, x = richest, y = anemia, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = Nepal_data)

#Generate means and SE
    mean_anemia <- svymean(~factor(anemia), sample, na.rm=T)
    mean_pregnant <- svymean(~factor(pregnant_anemia), sample, na.rm=T)
    mean_notpregnant <- svymean(~factor(notpregnant_anemia), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_anemia), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_anemia), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_anemia), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_anemia), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_anemia_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_anemia_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_anemia_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_anemia_poor), sample, na.rm=T)

#Generate cv
    cv(mean_anemia)*100
    cv(mean_pregnant)*100
    cv(mean_notpregnant)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 3. Ensure healthy lives and promote well-being for all at all ages

3.1.2 Proportion of births attended by skilled health personnel

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")
    install.packages("base")
    install.packages("knitr")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    SL_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/SLBR7AFL.dta")

###Step 2: Generate needed variables
library(base)
#Birth attended
    attach(SL_data)
    SL_data$birth_attended3 <- ifelse(m3a==1 | m3b==1 | m3c==1, 1, 0)

#Wealth
    SL_data$poorest <- ifelse(SL_data$v190==1,1,0)
    SL_data$richest <- ifelse(SL_data$v190==5,1,0)

#Weights
    SL_data $v005 <- SL_data $v005/1000000

#Interactive terms
    attach(SL_data)
    SL_data$urban_birth[v025==1 & birth_attended==0] <- 0
    SL_data$urban_birth[birth_attended==1 & v025==1] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$rural_birth[v025==2 & birth_attended==0] <- 0
    SL_data$rural_birth[birth_attended==1 & v025==2] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$poor_birth[poorest==1 & birth_attended==0] <- 0
    SL_data$poor_birth[birth_attended==1 & poorest==1] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$rich_birth[richest==0 & birth_attended==0] <- 0
    SL_data$rich_birth[birth_attended==1 & richest==0] <- 1
    detach(SL_data)
    attach(SL_data)
    SL_data$urban_birth_poor[v025==1 & birth_attended==0 & poorest==1] <- 0
    SL_data$urban_birth_poor[birth_attended==1 & v025==1 & poorest==1] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$urban_birth_rich[v025==1 & birth_attended==0 & richest==1] <- 0
    SL_data$urban_birth_rich[birth_attended==1 & v025==1 & richest==1] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$rural_birth_poor[v025==2 & birth_attended==0 & poorest==1] <- 0
    SL_data$rural_birth_poor[birth_attended==1 & v025==2 & poorest==1] <- 1
    detach(SL_data)

    attach(SL_data)
    SL_data$rural_birth_rich[v025==2 & birth_attended==0 & richest==1] <- 0
    SL_data$rural_birth_rich[birth_attended==1 & v025==2 & richest==1] <- 1
    detach(SL_data)

#Label variables
    library(sjlabelled)

    SL_data$poorest <- to_factor(SL_data$poorest)
    levels(SL_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    SL_data$richest <- to_factor(SL_data$richest)
    levels(SL_data$richest) <- list("Richest" = "1", "Not richest"="0")
    SL_data$v025 <- to_factor(SL_data$v025)
    levels(SL_data$v025) <- list("Urban" = "1", "Rural"="2")

    varlist4 <- c(“birth_attended”, "urban_birth", "rural_birth", "urban_birth_rich", "urban_birth_poor",
        "rural_birth_rich", "rural_birth_poor")

    for (var in varlist4) {
        SL_data[[var]] <- to_factor(SL_data[[var]])
        levels(SL_data[[var]]) <- list("Yes" = "1", "No"="0")
    }

###Step 3: Compute weighted proportions
    library(pollster)
    library(kableExtra)
    library(knitr)
    topline(df = SL_data, variable = birth_attended , weight = v005, pct = FALSE, remove = c("(Missing)"))
    %>%
            knitr::kable(digits = 1, "simple")
    crosstab(df = SL_data, x = v025, y = birth_attended, weight = v005, format = "long") %>%
            knitr::kable(digits = 1, "simple")
    crosstab(df = SL_data, x = poorest, y = birth_attended, weight = v005, format = "long") %>%
            knitr::kable(digits = 1, "simple")
    crosstab(df = SL_data, x = richest, y = birth_attended, weight = v005, format = "long") %>%
        knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = SL_data)

#Generate means and SE
    mean_birth <- svymean(~factor(birth_attended), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_birth), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_birth), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_birth), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_birth), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_birth_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_birth_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_birth_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_birth_poor), sample, na.rm=T)

#Generate cv
    cv(mean_birth)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 3. Ensure healthy lives and promote well-being for all at all ages

3.7.1 Proportion of women of reproductive age (aged 15–49 years) who have their need for family planning satisfied with modern methods

###Step 0: Install needed packages

```
install.packages("questionr")
install.packages("pollster")
install.packages("kableExtra")
install.packages("survey")
```

###Step 1: Import relevant DHS data and limit it to relevant denominator

```
library(haven)
Ug_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/UGIR7BFL.dta")
Ug_filtered <- subset(Ug_data, v012>=15 & v012 <= 49)
```

###Step 2: Generate needed variables

```
#Using modern contraceptives
    attach(Ug_filtered)
    Ug_filtered$use_modern[v313==3] <- 1
    Ug_filtered$use_modern[v313==1 | v313 ==2] <- 0
    Ug_filtered$use_modern[v313==0] <- NA
    detach(Ug_filtered)

#Wealth
    Ug_filtered$poorest <- ifelse(Ug_filtered$v190==1,1,0)
    Ug_filtered$richest <- ifelse(Ug_filtered$v190==5,1,0)

#Weights
    Ug_filtered$v005 <- Ug_filtered$v005/1000000

#Interactive term
    attach(Ug_filtered)
    Ug_filtered$urban_modern[v025==1 & use_modern==0] <- 0
    Ug_filtered$urban_modern[use_modern==1 & v025==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$rural_modern[v025==2 & use_modern==0] <- 0
    Ug_filtered$rural_modern[use_modern==1 & v025==2] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$poor_modern[poorest==1 & use_modern==0] <- 0
    Ug_filtered$poor_modern[use_modern==1 & poorest==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$rich_modern[richest==1 & use_modern==0] <- 0
    Ug_filtered$rich_modern[use_modern==1 & richest==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$urban_modern_poor[v025==1 & use_modern==0 & poorest==1] <- 0
    Ug_filtered$urban_modern_poor[use_modern==1 & v025==1 & poorest==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$urban_modern_rich[v025==1 & use_modern==0 & richest==1] <- 0
    Ug_filtered$urban_modern_rich[use_modern==1 & v025==1 & richest==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$rural_modern_poor[v025==2 & use_modern==0 & poorest==1] <- 0
    Ug_filtered$rural_modern_poor[use_modern==1 & v025==2 & poorest==1] <- 1
    detach(Ug_filtered)

    attach(Ug_filtered)
    Ug_filtered$rural_modern_rich[v025==2 & use_modern==0 & richest==1] <- 0
    Ug_filtered$rural_modern_rich[use_modern==1 & v025==2 & richest==1] <- 1
    detach(Ug_filtered)

# Label Variables
    library(sjlabelled)
    Ug_filtered$poorest <- to_factor(Ug_filtered$poorest)
    levels(Ug_filtered$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Ug_filtered$richest <- to_factor(Ug_filtered$richest)
    levels(Ug_filtered$richest) <- list("Richest" = "1", "Not richest"="0")
    Ug_filtered$v025 <- to_factor(Ug_filtered$v025)
    levels(Ug_filtered$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("use_modern", "urban_modern", "rural_modern", "urban_modern_rich",
    "urban_modern_poor", "rural_modern_rich", "rural_modern_poor")

     for (var in varlist4) {
                Ug_filtered[[var]] <- to_factor(Ug_filtered[[var]])
        levels(Ug_filtered[[var]]) <- list("Yes" = "1", "No"="0")
                }
```

***Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = Ug_filtered, variable = use_modern, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Ug_filtered, x = v025, y = use_modern, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Ug_filtered, x = poorest, y = use_modern, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

```
crosstab(df = Ug_filtered, x = richest, y = use_modern, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = Ug_filtered)

#Generate means and SE
    mean_modern <- svymean(~factor(use_modern), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_modern), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_modern), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_modern), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_modern), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_modern_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_modern_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_modern_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_modern_poor), sample, na.rm=T)

#Generate cv
    cv(mean_modern)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 4. Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all

4.x Proportion of women with primary or less education

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    Mongolia_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/wm.dta")
    Mongolia_data <- subset(Mongolia_data, WM==1) #completed interviews

###Step 2: Generate needed variables
#Primary education
    Mongolia_data$primary <- ifelse(Mongolia_data$WB5==2,1,0)
    Mongolia_data$primary[Mongolia_data$WB6A==1 & Mongolia_data$WB6B<=6] <- 1

#Wealth
    Mongolia_data$poorest <- ifelse(Mongolia_data$windex5==1,1,0)
    Mongolia_data$richest <- ifelse(Mongolia_data$windex5==5,1,0)

#Weights
    Mongolia_data$wmweight <- Mongolia_data$wmweight/1000000

#Interactive term
    attach(Mongolia_data)
    Mongolia_data$urban_primary[HH6==1 & primary==0] <- 0
    Mongolia_data$urban_primary[primary==1 & HH6==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$rural_primary[HH6==2 & primary==0] <- 0
    Mongolia_data$rural_primary[primary==1 & HH6==2] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$poor_primary[poorest==1 & primary==0] <- 0
    Mongolia_data$poor_primary[primary==1 & poorest==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$rich_primary[richest==1 & primary==0] <- 0
    Mongolia_data$rich_primary[primary==1 & richest==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$urban_primary_poor[HH6==1 & primary==0 & poorest==1] <- 0
    Mongolia_data$urban_primary_poor[primary==1 & HH6==1 & poorest==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$urban_primary_rich[HH6==1 & primary==0 & richest==1] <- 0
    Mongolia_data$urban_primary_rich[primary==1 & HH6==1 & richest==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$rural_primary_poor[HH6==2 & primary==0 & poorest==1] <- 0
    Mongolia_data$rural_primary_poor[primary==1 & HH6==2 & poorest==1] <- 1
    detach(Mongolia_data)

    attach(Mongolia_data)
    Mongolia_data$rural_primary_rich[HH6==2 & primary==0 & richest==1] <- 0
    Mongolia_data$rural_primary_rich[primary==1 & HH6==2 & richest==1] <- 1
    detach(Mongolia_data)

# Label Variables
    library(sjlabelled)
    Mongolia_data$poorest <- to_factor(Mongolia_data$poorest)
    levels(Mongolia_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Mongolia_data$richest <- to_factor(Mongolia_data$richest)
    levels(Mongolia_data$richest) <- list("Richest" = "1", "Not richest"="0")
    Mongolia_data$HH6 <- to_factor(Mongolia_data$HH6)
    levels(Mongolia_data$HH6) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("primary", "urban_primary", "rural_primary", "urban_primary_rich",
    "urban_primary_poor", "rural_primary_rich", "rural_primary_poor")

     for (var in varlist4) {
             Mongolia_data[[var]] <- to_factor(Mongolia_data[[var]])
        levels(Mongolia_data[[var]]) <- list("Yes" = "1", "No"="0")
             }
```

***Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = Mongolia_data, variable = primary, weight = wmweight, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Mongolia_data, x = HH6, y = primary, weight = wmweight, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Mongolia_data, x = poorest, y = primary, weight = wmweight, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Mongolia_data, x = richest, y = primary, weight = wmweight, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

```
###Step 4: Compute CV and SE
library(survey)

#Set survey design
      sample <- svydesign(id =~WM1, strata = ~HH7, weights = ~wmweight, data = Mongolia_data)

#Generate means and SE
      mean_primary <- svymean(~factor(primary), sample, na.rm=T)
      mean_urban <- svymean(~factor(urban_primary), sample, na.rm=T)
      mean_rural <- svymean(~factor(rural_primary), sample, na.rm=T)
      mean_poor <- svymean(~factor(poor_primary), sample, na.rm=T)
      mean_rich <- svymean(~factor(rich_primary), sample, na.rm=T)
      mean_rich_urban <- svymean(~factor(urban_primary_rich), sample, na.rm=T)
      mean_rich_rural <- svymean(~factor(rural_primary_rich), sample, na.rm=T)
      mean_poor_urban <- svymean(~factor(urban_primary_poor), sample, na.rm=T)
      mean_poor_rural <- svymean(~factor(rural_primary_poor), sample, na.rm=T)

#Generate cv
      cv(mean_primary)*100
      cv(mean_urban)*100
      cv(mean_poor)*100
      cv(mean_rural)*100
      cv(mean_rich)*100
      cv(mean_poor_rural)*100
      cv(mean_poor_urban)*100
      cv(mean_rich_rural)*100
      cv(mean_rich_urban)*100
```

Goal 5. Achieve gender equality and empower all women and girls

5.2.1 Proportion of ever-partnered women and girls aged 15 years and older subjected to physical, sexual or psychological violence by a current or former intimate partner in the previous 12 months, by form of violence and by age

```
###Step 0: Use required library packages
library(haven)
library(dplyr)
library(sjlabelled)
library(questionr)
library(pollster)
library(kableExtra)
library(knitr)
library(survey)

###Step 1: Import relevant DHS data and limit it to relevant denominator
Tajikistan <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/TJIR71FL.dta")
Tajikistan$d005 <- Tajikistan$d005/1000000
Tajikistan$d005[is.na(Tajikistan$d005)] = 0
#View(Tajikistan)

#Filtering variables of interest
Tajikistan_SDG5 <- subset(Tajikistan, v012>=15) # women aged 15+ years
#View(Tajikistan_SDG5)

###Step 2: Generate needed variables
#Create an index of VAW using d111 (physical), d104 (emotional), and d108 (sexual)
Tajikistan_SDG5$Index <- if_else(Tajikistan_SDG5$d111 == 1 | Tajikistan_SDG5$d104 == 1 |
Tajikistan_SDG5$d108 == 1, "Yes", "No")

#Interactions
attach(Tajikistan_SDG5)
Tajikistan_SDG5$rural_Index[v025==2 & Index=="Yes"] <- 1
Tajikistan_SDG5$rural_Index[v025==2 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

attach(Tajikistan_SDG5)
Tajikistan_SDG5$urban_Index[v025==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$urban_Index[v025==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$poorest <- if_else(Tajikistan_SDG5$v190==1,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$poorest_Index[poorest==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$poorest_Index[poorest==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$richest <- if_else(Tajikistan_SDG5$v190==5,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$richest_Index[richest==1 & Index=="Yes"] <- 1
```

```
Tajikistan_SDG5$richest_Index[richest==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age1 <- if_else(Tajikistan_SDG5$v013==1,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age1_Index[age1==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age1_Index[age1==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age2 <- if_else(Tajikistan_SDG5$v013==2,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age2_Index[age2==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age2_Index[age2==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age3 <- if_else(Tajikistan_SDG5$v013==3,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age3_Index[age3==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age3_Index[age3==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age4 <- if_else(Tajikistan_SDG5$v013==4,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age4_Index[age4==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age4_Index[age4==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age5 <- if_else(Tajikistan_SDG5$v013==5,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age5_Index[age5==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age5_Index[age5==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age6 <- if_else(Tajikistan_SDG5$v013==6,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age6_Index[age6==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age6_Index[age6==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

Tajikistan_SDG5$age7 <- if_else(Tajikistan_SDG5$v013==7,1,0)
attach(Tajikistan_SDG5)
Tajikistan_SDG5$age7_Index[age7==1 & Index=="Yes"] <- 1
Tajikistan_SDG5$age7_Index[age7==1 & Index=="No"] <- 0
detach(Tajikistan_SDG5)

###Step 3: Compute weighted proportions
#Tabulate proportion of women ever experienced violence by their partner
topline(df = Tajikistan_SDG5, variable = Index, weight = d005, pct = FALSE, remove = c("(Missing)")) %>%
  kable(digits = 2, "simple") #proportion that ever experienced violence of any form by partner/ husband
```

```
#Tabulation by the form of violence
topline(df = Tajikistan_SDG5, variable = violence, weight = d005, pct = FALSE, remove = c("(Missing)")) %>%
  kable(digits = 2, "simple")

topline(df = Tajikistan_SDG5, variable = sexualviolence, weight = d005, pct = FALSE, remove = c("(Missing)")) %>%
  kable(digits = 2, "simple")

topline(df = Tajikistan_SDG5, variable = emotionalviolence, weight = d005, pct = FALSE, remove = c("(Missing)")) %>%
  kable(digits = 2, "simple")


#Crosstabulation of violence by wealth, locality, age
pollster::crosstab(df = Tajikistan_SDG5, x = poorest, y = Index, weight = d005, format = "long")%>%
  kable(digits = 2, "simple")

pollster::crosstab(df = Tajikistan_SDG5, x = richest, y = Index, weight = d005, format = "long")%>%
  kable(digits = 2, "simple")

pollster::crosstab(df = Tajikistan_SDG5, x = v025, y = Index, weight = d005, format = "long")%>%
  kable(digits = 2, "simple")

pollster::crosstab(df = Tajikistan_SDG5, x = v013, y = Index, weight = d005, n = FALSE, format = "long")%>%
  kable(digits = 2, "simple")
```

###Step 4: Compute CV and SE

```
#Set survey design
sample <- svydesign(ids = ~v021, weights = ~d005, strata = ~v023, data = Tajikistan_SDG5, na.rm=T) #ids = specifying cluster ids from largest level to smallest level

#Mean and standard errors
mean_VAW <- svymean(~factor(Index), sample, na.rm=T)
mean_urban_vaw <- svymean(~factor(urban_Index), sample, na.rm=T)
mean_rural_vaw <- svymean(~factor(rural_Index), sample, na.rm=T)
mean_poorest_vaw <- svymean(~factor(poorest_Index), sample, na.rm=T)
mean_richest_vaw <- svymean(~factor(richest_Index), sample, na.rm=T)

mean_age1 <- svymean(~factor(age1_Index), sample, na.rm=T)
mean_age2 <- svymean(~factor(age2_Index), sample, na.rm=T)
mean_age3 <- svymean(~factor(age3_Index), sample, na.rm=T)
mean_age4 <- svymean(~factor(age4_Index), sample, na.rm=T)
mean_age5 <- svymean(~factor(age5_Index), sample, na.rm=T)
mean_age6 <- svymean(~factor(age6_Index), sample, na.rm=T)
mean_age7 <- svymean(~factor(age7_Index), sample, na.rm=T)

print("mean and standard errors are")
mean_VAW
mean_urban_vaw
mean_rural_vaw
```

```
mean_poorest_vaw
mean_richest_vaw

mean_age1
mean_age2
mean_age3
mean_age4
mean_age5
mean_age6
mean_age7

#Coefficient of variation
print("coefficient of variation is")
cv(mean_VAW)*100
cv(mean_urban_vaw)*100
cv(mean_rural_vaw)*100
cv(mean_poorest_vaw)*100
cv(mean_richest_vaw)*100

cv(mean_age1)*100
cv(mean_age2)*100
cv(mean_age3)*100
cv(mean_age4)*100
cv(mean_age5)*100
cv(mean_age6)*100
cv(mean_age7)*100
```

Goal 5. Achieve gender equality and empower all women and girls

5.3.1 Proportion of women aged 20–24 years who were married or in a union before age 15 and before age 18

###Step 0: Install needed packages

```
install.packages("questionr")
install.packages("pollster")
install.packages("kableExtra")
install.packages("survey")
```

###Step 1: Import relevant DHS data and limit it to relevant denominator

```
library(haven)
Egypt <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/EGIR61FL.dta")
Egypt$v005 <- Egypt$v005/1000000

Egypt_SDG5 <- subset(Egypt, v012>=20 & v012 <= 24)
```

###Step 2: Generate needed variables

```
#Before age15
    attach(Egypt_SDG5)
    Egypt_SDG5$age15 <- ifelse(v511<15,1,0)
    Egypt_SDG5$age15[is.na(v511)] <- NA
    detach(Egypt_SDG5)

#Before age18
    attach(Egypt_SDG5)
    Egypt_SDG5$age18 <- ifelse(v511<18,1,0)
    Egypt_SDG5$age18[is.na(v511)] <- NA
    detach(Egypt_SDG5)

#Wealth
    Egypt_SDG5$poorest <- ifelse(Egypt_SDG5$v190==1,1,0)
    Egypt_SDG5$richest <- ifelse(Egypt_SDG5$v190==5,1,0)

#Interactive term

    attach(Egypt_SDG5)
    Egypt_SDG5$urban_age15[v025==1 & age15==0] <- 0
    Egypt_SDG5$urban_age15[age15==1 & v025==1] <- 1
    detach(Egypt_SDG5)

    attach(Egypt_SDG5)
    Egypt_SDG5$rural_age15[v025==2 & age15==0] <- 0
    Egypt_SDG5$rural_age15[age15==1 & v025==2] <- 1
    detach(Egypt_SDG5)

    attach(Egypt_SDG5)
    Egypt_SDG5$poor_age15[poorest==1 & age15==0] <- 0
    Egypt_SDG5$poor_age15[age15==1 & poorest==1] <- 1
```

```
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$rich_age15[richest==1 & age15==0] <- 0
Egypt_SDG5$rich_age15[age15==1 & richest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$urban_age15_poor[v025==1 & age15==0 & poorest==1] <- 0
Egypt_SDG5$urban_age15_poor[age15==1 & v025==1 & poorest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$urban_age15_rich[v025==1 & age15==0 & richest==1] <- 0
Egypt_SDG5$urban_age15_rich[age15==1 & v025==1 & richest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$rural_age15_poor[v025==2 & age15==0 & poorest==1] <- 0
Egypt_SDG5$rural_age15_poor[age15==1 & v025==2 & poorest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$rural_age15_rich[v025==2 & age15==0 & richest==1] <- 0
Egypt_SDG5$rural_age15_rich[age15==1 & v025==2 & richest==1] <- 1
detach(Egypt_SDG5)


attach(Egypt_SDG5)
Egypt_SDG5$urban_age18[v025==1 & age18==0] <- 0
Egypt_SDG5$urban_age18[age18==1 & v025==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$rural_age18[v025==2 & age18==0] <- 0
Egypt_SDG5$rural_age18[age18==1 & v025==2] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$poor_age18[poorest==1 & age18==0] <- 0
Egypt_SDG5$poor_age18[age18==1 & poorest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$rich_age18[richest==1 & age18==0] <- 0
Egypt_SDG5$rich_age18[age18==1 & richest==1] <- 1
detach(Egypt_SDG5)

attach(Egypt_SDG5)
Egypt_SDG5$urban_age18_poor[v025==1 & age18==0 & poorest==1] <- 0
```

```
    Egypt_SDG5$urban_age18_poor[age18==1 & v025==1 & poorest==1] <- 1
    detach(Egypt_SDG5)

    attach(Egypt_SDG5)
    Egypt_SDG5$urban_age18_rich[v025==1 & age18==0 & richest==1] <- 0
    Egypt_SDG5$urban_age18_rich[age18==1 & v025==1 & richest==1] <- 1
    detach(Egypt_SDG5)

    attach(Egypt_SDG5)
    Egypt_SDG5$rural_age18_poor[v025==2 & age18==0 & poorest==1] <- 0
    Egypt_SDG5$rural_age18_poor[age18==1 & v025==2 & poorest==1] <- 1
    detach(Egypt_SDG5)

    attach(Egypt_SDG5)
    Egypt_SDG5$rural_age18_rich[v025==2 & age18==0 & richest==1] <- 0
    Egypt_SDG5$rural_age18_rich[age18==1 & v025==2 & richest==1] <- 1
    detach(Egypt_SDG5)

# Label Variables
    library(sjlabelled)
    Egypt_SDG5$poorest <- to_factor(Egypt_SDG5$poorest)
    levels(Egypt_SDG5$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Egypt_SDG5$richest <- to_factor(Egypt_SDG5$richest)
    levels(Egypt_SDG5$richest) <- list("Richest" = "1", "Not richest"="0")
    Egypt_SDG5$v025 <- to_factor(Egypt_SDG5$v025)
    levels(Egypt_SDG5$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("age15", “poor_age15”, “rich_age15”, "urban_age15", "rural_age15", "urban_age15_rich",
    "urban_age15_poor", "rural_age15_rich", "rural_age15_poor", "age18", “poor_age15”, “rich_age15”,
    "urban_age18", "rural_age18", "urban_age18_rich", "urban_age18_poor", "rural_age18_rich",
    "rural_age18_poor")

     for (var in varlist4) {
            Egypt_SDG5[[var]] <- to_factor(Egypt_SDG5[[var]])
        levels(Egypt_SDG5[[var]]) <- list("Yes" = "1", "No"="0")
            }
```

###Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = Egypt_SDG5, variable = age15, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = v025, y = age15, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = poorest, y = age15, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = richest, y = age15, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

```
topline(df = Egypt_SDG5, variable = age18, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = v025, y = age18, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = poorest, y = age18, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Egypt_SDG5, x = richest, y = age18, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = Egypt_SDG5)

#Generate means and SE
    mean_age15 <- svymean(~factor(age15), sample, na.rm=T)
    mean_urban_15 <- svymean(~factor(urban_age15), sample, na.rm=T)
    mean_rural_15 <- svymean(~factor(rural_age15), sample, na.rm=T)
    mean_poor_15 <- svymean(~factor(poor_age15), sample, na.rm=T)
    mean_rich_15 <- svymean(~factor(rich_age15), sample, na.rm=T)
    mean_rich_urban_15 <- svymean(~factor(urban_age15_rich), sample, na.rm=T)
    mean_rich_rural_15 <- svymean(~factor(rural_age15_rich), sample, na.rm=T)
    mean_poor_urban_15 <- svymean(~factor(urban_age15_poor), sample, na.rm=T)
    mean_poor_rural_15 <- svymean(~factor(rural_age15_poor), sample, na.rm=T)
    mean_age18 <- svymean(~factor(age18), sample, na.rm=T)
    mean_urban_18 <- svymean(~factor(urban_age18), sample, na.rm=T)
    mean_rural_18 <- svymean(~factor(rural_age18), sample, na.rm=T)
    mean_poor_18 <- svymean(~factor(poor_age18), sample, na.rm=T)
    mean_rich_18 <- svymean(~factor(rich_age18), sample, na.rm=T)
    mean_rich_urban_18 <- svymean(~factor(urban_age18_rich), sample, na.rm=T)
    mean_rich_rural_18 <- svymean(~factor(rural_age18_rich), sample, na.rm=T)
    mean_poor_urban_18 <- svymean(~factor(urban_age18_poor), sample, na.rm=T)
    mean_poor_rural_18 <- svymean(~factor(rural_age18_poor), sample, na.rm=T)

#Generate cv
    cv(mean_age15)*100
    cv(mean_urban_15)*100
    cv(mean_poor_15)*100
    cv(mean_rural_15)*100
    cv(mean_rich_15)*100
    cv(mean_poor_rural_15)*100
    cv(mean_poor_urban_15)*100
    cv(mean_rich_rural_15)*100
    cv(mean_rich_urban_15)*100
```

cv(mean_age18)*100
cv(mean_urban_18)*100
cv(mean_poor_18)*100
cv(mean_rural_18)*100
cv(mean_rich_18)*100
cv(mean_poor_rural_18)*100
cv(mean_poor_urban_18)*100
cv(mean_rich_rural_18)*100
cv(mean_rich_urban_18)*100

Goal 5. Achieve gender equality and empower all women and girls

5.6.1 Proportion of women aged 15–49 years who make their own informed decisions regarding contraceptive use and reproductive health care[6]

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    Kyrgyz <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/KYIR61FL.dta")
    Kyrgyz $v005 <- Kyrgyz$v005/1000000

    Kyrgyz_SDG5 <- subset(Kyrgyz, v012>=15 & v012 <= 49)

###Step 2: Generate needed variables
#Decision on contraceptive use
    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$cu <- ifelse(v632==1,1,0)
    Kyrgyz_SDG5$cu[is.na(v632)] <- NA
    detach(Kyrgyz_SDG5)

#Decision on reproductive health
    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$rh <- ifelse(v743a==1,1,0)
    Kyrgyz_SDG5$rh[is.na(v743a)] <- NA
    detach(Kyrgyz_SDG5)


#Wealth
    Kyrgyz_SDG5$poorest <- ifelse(Kyrgyz_SDG5$v190==1,1,0)
    Kyrgyz_SDG5$richest <- ifelse(Kyrgyz_SDG5$v190==5,1,0)

#Interactive term

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$urban_cu[v025==1 & cu==0] <- 0
    Kyrgyz_SDG5$urban_cu[cu==1 & v025==1] <- 1
    detach(Kyrgyz_SDG5)

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$rural_cu[v025==2 & cu==0] <- 0
    Kyrgyz_SDG5$rural_cu[cu==1 & v025==2] <- 1
```

[6] Full SDG indicator: Proportion of women aged 15–49 years who make their own informed decisions regarding sexual relations, contraceptive use and reproductive health care. Computation should be made as a combination of all three components. However, since this country dataset only had two out of three components, the variables were computed separately.

```
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$poor_cu[poorest==1 & cu==0] <- 0
Kyrgyz_SDG5$poor_cu[cu==1 & poorest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$rich_cu[richest==1 & cu==0] <- 0
Kyrgyz_SDG5$rich_cu[cu==1 & richest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$urban_cu_poor[v025==1 & cu==0 & poorest==1] <- 0
Kyrgyz_SDG5$urban_cu_poor[cu==1 & v025==1 & poorest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$urban_cu_rich[v025==1 & cu==0 & richest==1] <- 0
Kyrgyz_SDG5$urban_cu_rich[cu==1 & v025==1 & richest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$rural_cu_poor[v025==2 & cu==0 & poorest==1] <- 0
Kyrgyz_SDG5$rural_cu_poor[cu==1 & v025==2 & poorest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$rural_cu_rich[v025==2 & cu==0 & richest==1] <- 0
Kyrgyz_SDG5$rural_cu_rich[cu==1 & v025==2 & richest==1] <- 1
detach(Kyrgyz_SDG5)


attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$urban_rh[v025==1 & rh==0] <- 0
Kyrgyz_SDG5$urban_rh[rh==1 & v025==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$rural_rh[v025==2 & rh==0] <- 0
Kyrgyz_SDG5$rural_rh[rh==1 & v025==2] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$poor_rh[poorest==1 & rh==0] <- 0
Kyrgyz_SDG5$poor_rh[rh==1 & poorest==1] <- 1
detach(Kyrgyz_SDG5)

attach(Kyrgyz_SDG5)
Kyrgyz_SDG5$rich_rh[richest==1 & rh==0] <- 0
```

```
    Kyrgyz_SDG5$rich_rh[rh==1 & richest==1] <- 1
    detach(Kyrgyz_SDG5)

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$urban_rh_poor[v025==1 & rh==0 & poorest==1] <- 0
    Kyrgyz_SDG5$urban_rh_poor[rh==1 & v025==1 & poorest==1] <- 1
    detach(Kyrgyz_SDG5)

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$urban_rh_rich[v025==1 & rh==0 & richest==1] <- 0
    Kyrgyz_SDG5$urban_rh_rich[rh==1 & v025==1 & richest==1] <- 1
    detach(Kyrgyz_SDG5)

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$rural_rh_poor[v025==2 & rh==0 & poorest==1] <- 0
    Kyrgyz_SDG5$rural_rh_poor[rh==1 & v025==2 & poorest==1] <- 1
    detach(Kyrgyz_SDG5)

    attach(Kyrgyz_SDG5)
    Kyrgyz_SDG5$rural_rh_rich[v025==2 & rh==0 & richest==1] <- 0
    Kyrgyz_SDG5$rural_rh_rich[rh==1 & v025==2 & richest==1] <- 1
    detach(Kyrgyz_SDG5)

# Label Variables
    library(sjlabelled)
    Kyrgyz_SDG5$poorest <- to_factor(Kyrgyz_SDG5$poorest)
    levels(Kyrgyz_SDG5$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    Kyrgyz_SDG5$richest <- to_factor(Kyrgyz_SDG5$richest)
    levels(Kyrgyz_SDG5$richest) <- list("Richest" = "1", "Not richest"="0")
    Kyrgyz_SDG5$v025 <- to_factor(Kyrgyz_SDG5$v025)
    levels(Kyrgyz_SDG5$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("cu", "poor_cu", "rich_cu", "urban_cu", "rural_cu", "urban_cu_rich", "urban_cu_poor",
    "rural_cu_rich", "rural_cu_poor", "rh", "poor_cu", "rich_cu", "urban_rh", "rural_rh", "urban_rh_rich",
    "urban_rh_poor", "rural_rh_rich", "rural_rh_poor")

     for (var in varlist4) {
            Kyrgyz_SDG5[[var]] <- to_factor(Kyrgyz_SDG5[[var]])
        levels(Kyrgyz_SDG5[[var]]) <- list("Yes" = "1", "No"="0")
            }

###Step 3: Compute weighted proportions
library(pollster)
library(kableExtra)
topline(df = Kyrgyz_SDG5, variable = cu, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = v025, y = cu, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = poorest, y = cu, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = richest, y = cu, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

topline(df = Kyrgyz_SDG5, variable = rh, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = v025, y = rh, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = poorest, y = rh, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = Kyrgyz_SDG5, x = richest, y = rh, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = Kyrgyz_SDG5)

#Generate means and SE
    mean_cu <- svymean(~factor(cu), sample, na.rm=T)
    mean_urban_cu <- svymean(~factor(urban_cu), sample, na.rm=T)
    mean_rural_cu <- svymean(~factor(rural_cu), sample, na.rm=T)
    mean_poor_cu <- svymean(~factor(poor_cu), sample, na.rm=T)
    mean_rich_cu <- svymean(~factor(rich_cu), sample, na.rm=T)
    mean_rich_urban_cu <- svymean(~factor(urban_cu_rich), sample, na.rm=T)
    mean_rich_rural_cu <- svymean(~factor(rural_cu_rich), sample, na.rm=T)
    mean_poor_urban_cu <- svymean(~factor(urban_cu_poor), sample, na.rm=T)
    mean_poor_rural_cu <- svymean(~factor(rural_cu_poor), sample, na.rm=T)
    mean_rh <- svymean(~factor(rh), sample, na.rm=T)
    mean_urban_rh <- svymean(~factor(urban_rh), sample, na.rm=T)
    mean_rural_rh <- svymean(~factor(rural_rh), sample, na.rm=T)
    mean_poor_rh <- svymean(~factor(poor_rh), sample, na.rm=T)
    mean_rich_rh <- svymean(~factor(rich_rh), sample, na.rm=T)
    mean_rich_urban_rh <- svymean(~factor(urban_rh_rich), sample, na.rm=T)
    mean_rich_rural_rh <- svymean(~factor(rural_rh_rich), sample, na.rm=T)
    mean_poor_urban_rh <- svymean(~factor(urban_rh_poor), sample, na.rm=T)
    mean_poor_rural_rh <- svymean(~factor(rural_rh_poor), sample, na.rm=T)

#Generate cv
    cv(mean_cu)*100
    cv(mean_urban_cu)*100
    cv(mean_poor_cu)*100
    cv(mean_rural_cu)*100
    cv(mean_rich_cu)*100
cv(mean_poor_rural_cu)*100
cv(mean_poor_urban_cu)*100
cv(mean_rich_rural_cu)*100
cv(mean_rich_urban_cu)*100
cv(mean_rh)*100
cv(mean_urban_rh)*100
cv(mean_poor_rh)*100
cv(mean_rural_rh)*100
cv(mean_rich_rh)*100
cv(mean_poor_rural_rh)*100
cv(mean_poor_urban_rh)*100
cv(mean_rich_rural_rh)*100
cv(mean_rich_urban_rh)*100
```

Goal 5. Achieve gender equality and empower all women and girls

5.b.1 Proportion of women who own a mobile telephone[7]

```
###Step 0: Install needed packages
install.packages("questionr")
install.packages("pollster")
install.packages("kableExtra")
install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
library(haven)
JO_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/JOIR73FL.dta ")

###Step 2: Generate needed variables
#Mobileown Access
    attach(JO_data)
    JO_data$mobileown_use[v169a==0] <- 0
    JO_data$mobileown_use[v169a==1] <- 1
    detach(JO_data)

#Wealth
    JO_data$poorest <- ifelse(JO_data$v190==1,1,0)
    JO_data$richest <- ifelse(JO_data$v190==5,1,0)

#Weights
    JO_data$v005 <- JO_data$v005/1000000

#Interactive terms
    attach(JO_data)
    JO_data$urban_use[v025==1 & mobileown_use==0] <- 0
    JO_data$urban_use[mobileown_use==1 & v025==1] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$rural_use[v025==2 & mobileown_use==0] <- 0
    JO_data$rural_use[mobileown_use==1 & v025==2] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$poor_use[poorest==1 & mobileown_use==0] <- 0
    JO_data$poor_use[mobileown_use==1 & poorest==1] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$rich_use[richest==1 & mobileown_use==0] <- 0
    JO_data$rich_use[mobileown_use==1 & richest==1] <- 1
    detach(JO_data)
```

[7] Full SDG indicator: Proportion of individuals who own a mobile telephone, by sex

```
    attach(JO_data)
    JO_data$urban_use_poor[v025==1 & mobileown_use==0 & poorest==1] <- 0
    JO_data$urban_use_poor[mobileown_use==1 & v025==1 & poorest==1] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$urban_use_rich[v025==1 & mobileown_use==0 & richest==1] <- 0
    JO_data$urban_use_rich[mobileown_use==1 & v025==1 & richest==1] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$rural_use_poor[v025==2 & mobileown_use==0 & poorest==1] <- 0
    JO_data$rural_use_poor[mobileown_use==1 & v025==2 & poorest==1] <- 1
    detach(JO_data)

    attach(JO_data)
    JO_data$rural_use_rich[v025==2 & mobileown_use==0 & richest==1] <- 0
    JO_data$rural_use_rich[mobileown_use==1 & v025==2 & richest==1] <- 1
    detach(JO_data)

#Label variables
    library(sjlabelled)
    varlist4 <- c("mobileown_use", "urban_use", "rural_use", "urban_use_rich", "urban_use_poor",
"rural_use_rich", "rural_use_poor")

    for (var in varlist4)  {
    JO_data[[var]] <- to_factor(JO_data[[var]])
    levels(JO_data[[var]]) <- list("Yes" = "1", "No"="0")
    }

    JO_data$poorest <- to_factor(JO_data$poorest)
      levels(JO_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")

    JO_data$v025 <- to_factor(JO_data$v025)
    levels(JO_data$v025) <- list("Urban" = "1", "Rural"="2")

    JO_data$richest <- to_factor(JO_data$richest)
    levels(JO_data$richest) <- list("Richest" = "1", "Not richest"="0")

###Step 3: Compute weighted proportions
library(pollster)
library(kableExtra)
topline(df = JO_data, variable = mobileown_use, weight = v005, valid_pct = FALSE) %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = JO_data, x = v025, y = mobileown_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = JO_data, x = poorest, y = mobileown_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = JO_data, x = richest, y = mobileown_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = JO_data)

#Generate means and SE
    mean_mobileown <- svymean(~factor(mobileown_use), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_use), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_use), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_use), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_use), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_use_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_use_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_use_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_use_poor), sample, na.rm=T)

#Generate cv
    cv(mean_mobileown)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 7. Ensure access to affordable, reliable, sustainable and modern energy for all

7.1.2 Proportion of population with primary reliance on clean fuels and technology

```
###Step 0: Install needed packages
    install.packages("questionr")
    install.packages("pollster")
    install.packages("kableExtra")
    install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    BD_Data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/BDIR7RFL.dta")
    BD_Data$v005 <- BD_Data$v005/1000000

###Step 2: Generate needed variables
#Using cleanfuel
    attach(BD_Data)
    BD_Data$cleanfuel <- ifelse(v161==1 | v161==2 | v161==3 | v161==4,1,0)
    BD_Data$cleanfuel[is.na(v161)] <- NA
    detach(BD_Data)

#Wealth
    BD_Data$poorest <- ifelse(BD_Data$v190==1,1,0)
    BD_Data$richest <- ifelse(BD_Data$v190==5,1,0)

#Interactive term

    attach(BD_Data)
    BD_Data$urban_cleanfuel[v025==1 & cleanfuel==0] <- 0
    BD_Data$urban_cleanfuel[cleanfuel==1 & v025==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$rural_cleanfuel[v025==2 & cleanfuel==0] <- 0
    BD_Data$rural_cleanfuel[cleanfuel==1 & v025==2] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$poor_cleanfuel[poorest==1 & cleanfuel==0] <- 0
    BD_Data$poor_cleanfuel[cleanfuel==1 & poorest==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$rich_cleanfuel[richest==1 & cleanfuel==0] <- 0
    BD_Data$rich_cleanfuel[cleanfuel==1 & richest==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$urban_cleanfuel_poor[v025==1 & cleanfuel==0 & poorest==1] <- 0
    BD_Data$urban_cleanfuel_poor[cleanfuel==1 & v025==1 & poorest==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$urban_cleanfuel_rich[v025==1 & cleanfuel==0 & richest==1] <- 0
    BD_Data$urban_cleanfuel_rich[cleanfuel==1 & v025==1 & richest==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$rural_cleanfuel_poor[v025==2 & cleanfuel==0 & poorest==1] <- 0
    BD_Data$rural_cleanfuel_poor[cleanfuel==1 & v025==2 & poorest==1] <- 1
    detach(BD_Data)

    attach(BD_Data)
    BD_Data$rural_cleanfuel_rich[v025==2 & cleanfuel==0 & richest==1] <- 0
    BD_Data$rural_cleanfuel_rich[cleanfuel==1 & v025==2 & richest==1] <- 1
    detach(BD_Data)

# Label Variables
    library(sjlabelled)
    BD_Data$poorest <- to_factor(BD_Data$poorest)
    levels(BD_Data$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    BD_Data$richest <- to_factor(BD_Data$richest)
    levels(BD_Data$richest) <- list("Richest" = "1", "Not richest"="0")
    BD_Data$v025 <- to_factor(BD_Data$v025)
    levels(BD_Data$v025) <- list("Urban" = "1", "Rural"="2")
    varlist4 <- c("cleanfuel", "poor_cleanfuel", "rich_cleanfuel", "urban_cleanfuel", "rural_cleanfuel",
    "urban_cleanfuel_rich", "urban_cleanfuel_poor", "rural_cleanfuel_rich", "rural_cleanfuel_poor”)

     for (var in varlist4) {
            BD_Data[[var]] <- to_factor(BD_Data[[var]])
        levels(BD_Data[[var]]) <- list("Yes" = "1", "No"="0")
            }
```

###Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = BD_Data, variable = cleanfuel, weight = v005, valid_pct = FALSE) %>%
knitr::kable(digits = 1, "simple")

crosstab(df = BD_Data, x = v025, y = cleanfuel, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = BD_Data, x = poorest, y = cleanfuel, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")

crosstab(df = BD_Data, x = richest, y = cleanfuel, weight = v005, format = "long") %>%
knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = BD_Data)

#Generate means and SE
    mean_cleanfuel <- svymean(~factor(cleanfuel), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_cleanfuel), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_cleanfuel), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_cleanfuel), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_cleanfuel), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_cleanfuel_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_cleanfuel_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_cleanfuel_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_cleanfuel_poor), sample, na.rm=T)

#Generate cv
    cv(mean_cleanfuel)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 8. Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all

8.10.2 Proportion of adults (15 years and older) with an account at a bank or other financial institution or with a mobile-money-service provider

```
###Step 0: Install needed packages
install.packages("questionr")
install.packages("pollster")
install.packages("kableExtra")
install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
    library(haven)
    TZ_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/TZIR7BFL.dta")
    TZ_data<- subset(TZ_data, v012>=15) # women 15+

###Step 2: Generate needed variables
#Bank Account
    attach(TZ_data)
    TZ_data$bank_own[v170==0] <- 0
    TZ_data$bank_own[v170==1] <- 1
    detach(TZ_data)

#Wealth
    TZ_data$poorest <- ifelse(TZ_data$v190==1,1,0)
    TZ_data$richest <- ifelse(TZ_data$v190==5,1,0)

#Weights
    TZ_data $v005 <- TZ_data $v005/1000000

#Interactive terms
    attach(TZ_data)
    TZ_data$urban_bankown[v025==1 & v170==0] <- 0
    TZ_data$urban_bankown[v170==1 & v025==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$rural_bankown[v025==2 & v170==0] <- 0
    TZ_data$rural_bankown[v170==1 & v025==2] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$poor_bankown[poorest==1 & v170==0] <- 0
    TZ_data$poor_bankown[v170==1 & poorest==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$rich_bankown[richest==1 & v170==0] <- 0
    TZ_data$rich_bankown[v170==1 & richest==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$urban_bankown_poor[v025==1 & v170==0 & poorest==1] <- 0
    TZ_data$urban_bankown_poor[v170==1 & v025==1 & poorest==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$urban_bankown_rich[v025==1 & v170==0 & richest==1] <- 0
    TZ_data$urban_bankown_rich[v170==1 & v025==1 & richest==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$rural_bankown_poor[v025==2 & v170==0 & poorest==1] <- 0
    TZ_data$rural_bankown_poor[v170==1 & v025==2 & poorest==1] <- 1
    detach(TZ_data)

    attach(TZ_data)
    TZ_data$rural_bankown_rich[v025==2 & v170==0 & richest==1] <- 0
    TZ_data$rural_bankown_rich[v170==1 & v025==2 & richest==1] <- 1
    detach(TZ_data)

#Label variables
    library(sjlabelled)
    TZ_data$v025 <- to_factor(TZ_data$v025)
    levels(TZ_data$v025) <- list("Urban" = "1", "Rural"="2")
    TZ_data$poorest <- to_factor(TZ_data$poorest)
    levels(TZ_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")
    TZ_data$richest <- to_factor(TZ_data$richest)
    levels(TZ_data$richest) <- list("Richest" = "1", "Not richest"="0")
```

###Step 3: Compute weighted proportions

```
library(pollster)
library(kableExtra)
topline(df = TZ_data, variable = bank_own, weight = v005, valid_pct = FALSE) %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = TZ_data, x = v025, y = bank_own, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = TZ_data, x = poorest, y = bank_own, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = TZ_data, x = richest, y = bank_own, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = TZ_data)

#Generate means and SE
```

```
    mean_bank <- svymean(~factor(bank_own), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_bankown), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_bankown), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_bankown), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_bankown), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_bankown_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_bankown_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_bankown_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_bankown_poor), sample, na.rm=T)

# Generate cv
    cv(mean_bank)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```

Goal 16. Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels

16.2.3 Proportion of young women aged 18–29 years who experienced sexual violence by age 18

```
###Step 0: Use required library packages
library(haven)
library(dplyr)
library(sjlabelled)
library(questionr)
library(pollster)
library(kableExtra)
library(knitr)
library(survey)

###Step 1: Import relevant DHS data and limit it to relevant denominator
Senegal <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/SNIR8BFL.dta")
Senegal$d005 <- Senegal$d005/1000000
Senegal$d005[is.na(Senegal$d005)] = 0

Senegal_SDG16 <- subset(Senegal, v012>=18 & v012<=29)

###Step 2: Generate needed variables
attach(Senegal_SDG16)
Senegal_SDG16$sexualvaw <- ifelse(d126<=18,1,0)
Senegal_SDG16$sexualvaw[is.na(d126)] <- NA
detach(Senegal_SDG16)

attach(Senegal_SDG16)
Senegal_SDG16$urban_sexualvaw[v025==1 & sexualvaw==1] <- 1
Senegal_SDG16$urban_sexualvaw[v025==1 & sexualvaw==0] <- 0
detach(Senegal_SDG16)

attach(Senegal_SDG16)
Senegal_SDG16$rural_sexualvaw[v025==2 & sexualvaw==1] <- 1
Senegal_SDG16$rural_sexualvaw[v025==2 & sexualvaw==0] <- 0
detach(Senegal_SDG16)

Senegal_SDG16$poorest <- ifelse(Senegal_SDG16$v190==1,1,0)
attach(Senegal_SDG16)
Senegal_SDG16$poorest_sexualvaw[poorest==1 & sexualvaw==1] <- 1
Senegal_SDG16$poorest_sexualvaw[poorest==1 & sexualvaw==0] <- 0
detach(Senegal_SDG16)

Senegal_SDG16$richest <- ifelse(Senegal_SDG16$v190==5,1,0)
attach(Senegal_SDG16)
Senegal_SDG16$richest_sexualvaw[richest==1 & sexualvaw==1] <- 1
Senegal_SDG16$richest_sexualvaw[richest==1 & sexualvaw==0] <- 0
detach(Senegal_SDG16)
```

###Step 3: Compute weighted proportions

```
    #tabulate proportion of women had sexual violence
    topline(df = Senegal_SDG16, variable = sexualvaw, weight = d005, pct = FALSE, remove = c("(Missing)"))
        %>%
      kable(digits = 2, "simple") #proportion that ever experienced sexual violence

    pollster::crosstab(df = Senegal_SDG16, x = poorest, y = sexualvaw, weight = d005, n = FALSE, format =
        "long") %>%
      kable(digits = 2, "simple")

    pollster::crosstab(df = Senegal_SDG16, x = richest, y = sexualvaw, weight = d005, n = FALSE, format =
        "long") %>%
      kable(digits = 2, "simple")

    pollster::crosstab(df = Senegal_SDG16, x = v025, y = sexualvaw, weight = d005, n = FALSE, format =
        "long") %>%
      kable(digits = 2, "simple")
```

###Step 4: Compute CV and SE

```
#Calculate Standard errors and coefficient of variation
    sample <- svydesign(ids = ~v021, weights = ~d005, strata = ~v023, data = Senegal_SDG16) #ids =
        specifying cluster ids from largest level to smallest level

#Mean and standard errors
    mean_sexualvaw <- svymean(~factor(sexualvaw), sample, na.rm=T)
    mean_urban_sexualvaw <- svymean(~factor(urban_sexualvaw), sample, na.rm=T)
    mean_rural_sexualvaw <- svymean(~factor(rural_sexualvaw), sample, na.rm=T)
    mean_poorest_sexualvaw <- svymean(~factor(poorest_sexualvaw), sample, na.rm=T)
    mean_richest_sexualvaw <- svymean(~factor(richest_sexualvaw), sample, na.rm=T)

    print("mean and standard errors are")
    mean_sexualvaw
    mean_rural_sexualvaw
    mean_urban_sexualvaw

#Coefficient of variation
    print("coefficient of variation is")
    cv(mean_sexualvaw)
    cv(mean_urban_sexualvaw)
    cv(mean_rural_sexualvaw)
    cv(mean_poorest_sexualvaw)
    cv(mean_richest_sexualvaw)
```

Goal 17. Strengthen the means of implementation and revitalize the Global Partnership for Sustainable Development

17.8.1 Proportion of women using the Internet in the last 12 months[8]

```
###Step 0: Install needed packages
install.packages("questionr")
install.packages("pollster")
install.packages("kableExtra")
install.packages("survey")

###Step 1: Import relevant DHS data and limit it to relevant denominator
library(haven)
CA_data <- read_dta("D:/OneDrive - UN Women/Toolkit/Data/CMIR71FL.dta ")

###Step 2: Generate needed variables
#Internet Access
    attach(CA_data)
    CA_data$internet_use[v171a==0 | v171a==2] <- 0
    CA_data$internet_use[v171a==1] <- 1
    detach(CA_data)

#Wealth
    CA_data$poorest <- ifelse(CA_data$v190==1,1,0)
    CA_data$richest <- ifelse(CA_data$v190==5,1,0)

#Weights
    CA_data$v005 <- CA_data$v005/1000000

#Interactive terms
    attach(CA_data)
    CA_data$urban_use[v025==1 & internet_use==0] <- 0
    CA_data$urban_use[internet_use==1 & v025==1] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$rural_use[v025==2 & internet_use==0] <- 0
    CA_data$rural_use[internet_use==1 & v025==2] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$poor_use[poorest==1 & internet_use==0] <- 0
    CA_data$poor_use[internet_use==1 & poorest==1] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$rich_use[richest==1 & internet_use==0] <- 0
    CA_data$rich_use[internet_use==1 & richest==1] <- 1
```

[8] The full SDG goal is: Proportion of individuals using the Internet in the last three months.

```
    detach(CA_data)

    attach(CA_data)
    CA_data$urban_use_poor[v025==1 & internet_use==0 & poorest==1] <- 0
    CA_data$urban_use_poor[internet_use==1 & v025==1 & poorest==1] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$urban_use_rich[v025==1 & internet_use==0 & richest==1] <- 0
    CA_data$urban_use_rich[internet_use==1 & v025==1 & richest==1] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$rural_use_poor[v025==2 & internet_use==0 & poorest==1] <- 0
    CA_data$rural_use_poor[internet_use==1 & v025==2 & poorest==1] <- 1
    detach(CA_data)

    attach(CA_data)
    CA_data$rural_use_rich[v025==2 & internet_use==0 & richest==1] <- 0
    CA_data$rural_use_rich[internet_use==1 & v025==2 & richest==1] <- 1
    detach(CA_data)

#Label variables
    library(sjlabelled)
    varlist4 <- c("internet_use", "urban_use", "rural_use", "urban_use_rich", "urban_use_poor",
"rural_use_rich", "rural_use_poor")

    for (var in varlist4)  {
    CA_data[[var]] <- to_factor(CA_data[[var]])
    levels(CA_data[[var]]) <- list("Yes" = "1", "No"="0")
    }

    CA_data$poorest <- to_factor(CA_data$poorest)
      levels(CA_data$poorest) <- list("Poorest" = "1", "Not poorest"="0")

    CA_data$v025 <- to_factor(CA_data$v025)
    levels(CA_data$v025) <- list("Urban" = "1", "Rural"="2")

    CA_data$richest <- to_factor(CA_data$richest)
    levels(CA_data$richest) <- list("Richest" = "1", "Not richest"="0")

###Step 3: Compute weighted proportions
library(pollster)
library(kableExtra)
topline(df = CA_data, variable = internet_use, weight = v005, valid_pct = FALSE) %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = CA_data, x = v025, y = internet_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
crosstab(df = CA_data, x = poorest, y = internet_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
```

```
crosstab(df = CA_data, x = richest, y = internet_use, weight = v005, format = "long") %>%
    knitr::kable(digits = 1, "simple")
```

###Step 4: Compute CV and SE

```
library(survey)

#Set survey design
    sample <- svydesign(id =~v021, strata = ~v023, weights = ~v005, data = CA_data)

#Generate means and SE
    mean_internet <- svymean(~factor(internet_use), sample, na.rm=T)
    mean_urban <- svymean(~factor(urban_use), sample, na.rm=T)
    mean_rural <- svymean(~factor(rural_use), sample, na.rm=T)
    mean_poor <- svymean(~factor(poor_use), sample, na.rm=T)
    mean_rich <- svymean(~factor(rich_use), sample, na.rm=T)
    mean_rich_urban <- svymean(~factor(urban_use_rich), sample, na.rm=T)
    mean_rich_rural <- svymean(~factor(rural_use_rich), sample, na.rm=T)
    mean_poor_urban <- svymean(~factor(urban_use_poor), sample, na.rm=T)
    mean_poor_rural <- svymean(~factor(rural_use_poor), sample, na.rm=T)

#Generate cv
    cv(mean_internet)*100
    cv(mean_urban)*100
    cv(mean_poor)*100
    cv(mean_rural)*100
    cv(mean_rich)*100
    cv(mean_poor_rural)*100
    cv(mean_poor_urban)*100
    cv(mean_rich_rural)*100
    cv(mean_rich_urban)*100
```