# HW3

## Summary

In HW2, we created an R package with functions with documentation. In this HW, we are going to add some additional functionality to that R package, including error messages and tests (extra credit), to make the package robust. You should start from what you have from HW2.

### What to hand in

Please submit your zipped package and associated files described below. The submission needs to contain:

1. an R package that includes the functions needed for the HW exercises.
2. an R notebook called hw3_exercises.Rmd and associated knitted html that shows the function calls and output as displayed in the hw3 starter Rmd/html with example solutions.
3. a link to your GitHub repo, which contains your package, Rmd, and knitted html file.

Grading: 25 points + 5 points extra credit

- 10 points for exercise 1
- 10 points for exercise 2
- 5 points for exercise 3
- 5 points extra credit for exercise 4
- Maximum point for HW3 is 25.

```
# to load your package
# my package is called hw3
# load_all("../hw3")
```

## HW3 exercises

In HW3, we will keep working on the package we developed in HWs 1 and 2. We will (i) extend the plotting function to provide informative error messages if the function is called with incorrect inputs, (ii) add a new function, and (iii) write a test for the new function in the package.

As a reminder, the data set csv file contraceptive_use.csv (which should be put under `data` folder within your R package) contains observed mCPR values. Relevant variable info is as follows:

- contraceptive_use_modern = observed mCPR
- division_numeric_code = country iso code
- is_in_union == "Y" refers to observations among married women.
- (start_date + end_date)/2 is the reference time for the observation.

## Exercise 1: write a new function `get_width_ci()` to calculate the width of the uncertainty intervals

Write a function `get_width_ci` that calculates the widths of the uncertainty intervals for a country, based on 95 or 80% bounds. The function output should be a data frame, where the first column is the year, and the second column is the corresponding interval width.

Similar to HW2, in this HW3, you should not define the `get_width_ci()` in this Rmd. Instead, you should define it in `R/get_width_ci.R` file and load it by `load_all()` function call before.

You should be able to replicate the following results:

```
get_width_ci(est, iso_code = 4, coverage = 95)
```

```
## # A tibble: 61 x 2
##     year width
##    <dbl> <dbl>
##  1  1970   2
##  2  1971   2.1
##  3  1972   2.1
##  4  1973   2.3
##  5  1974   2.4
##  6  1975   2.6
##  7  1976   2.9
##  8  1977   3.1
##  9  1978   3.4
## 10  1979   3.7
## # i 51 more rows
```

```
get_width_ci(est, iso_code = 4, coverage = 80)
```

```
## # A tibble: 61 x 2
##     year width
##    <dbl> <dbl>
##  1  1970   1.3
##  2  1971   1.3
##  3  1972   1.4
##  4  1973   1.5
##  5  1974   1.5
##  6  1975   1.6
##  7  1976   1.8
##  8  1977   2
##  9  1978   2.1
## 10  1979   2.3
## # i 51 more rows
```

## Exercise 2: add error messages in `plot_cp`

Extend the function `plot_cp` such that it produces an informative error message when

(a) input "dat" does NOT contain variable iso, info on reference year, and/or contraceptive_use_modern
(b) iso_code is not found in "dat" or "est"

(c) contraceptive_use_modern is not numeric
(d) CI is not one of 80, 95, or NA

Write code (function calls) in your notebook to print the error message (set chunk options as in starter code Rmd to make sure that you can still knit the rmd). See examples below.

**Input dat does NOT contain variable iso, year and/or cp**

```
plot_cp(dat %>% select(-iso), est, iso_code = 4)
```

```
## Error in plot_cp(dat %>% select(-iso), est, iso_code = 4):
##   Input data file dat and estimates file est must contain variable iso.
```

```
plot_cp(dat %>% select(-year), est, iso_code = 4)
```

```
## Error in plot_cp(dat %>% select(-year), est, iso_code = 4):
##   Input data file dat must contain variable year and cp.
```

```
plot_cp(dat %>% select(-cp), est, iso_code = 4)
```

```
## Error in plot_cp(dat %>% select(-cp), est, iso_code = 4):
##   Input data file dat must contain variable year and cp.
```

**iso_code is not found in "dat" or "est"**

```
plot_cp(dat %>% select(-iso), est, iso_code = 4)
```

```
## Error in plot_cp(dat %>% select(-iso), est, iso_code = 4):
##   Input data file dat and estimates file est must contain variable iso.
```

```
plot_cp(dat, est %>% select(-iso), iso_code = 4)
```

```
## Error in plot_cp(dat, est %>% select(-iso), iso_code = 4):
##   Input data file dat and estimates file est must contain variable iso.
```

**cp is not numeric**

```
dat_bug <- dat %>%
  mutate(cp = FALSE)
plot_cp(dat_bug, est, iso_code = 4)
```

```
## Error in plot_cp(dat_bug, est, iso_code = 4):
##   Input cp in data file dat must be numeric.
```

**CI is not one of 80, 95, or NA**

```
plot_cp(dat, est, iso_code = 4, CI = 99)
```

```
## Error in plot_cp(dat, est, iso_code = 4, CI = 99):
##   CI must be 80, 95, or NA.
```

### Exercise 3: GitHub

Now it is time to share your work again! Create a repo on your GitHub. Commit and push your package files, Rmd and knitted html to your repo. Paste the link to your GitHub repo below.

Note that in order to earn full credits, your GitHub repo commit history should show at least two commits.

Paste your link here...

### Exercise 4 (extra credit): write a test for function `get_width_ci`

Add the testing set up to your package and write unit tests to verify that the function `get_width_ci` (i) correctly returns the width of a 95% interval, defined as U95-L95, for some dummy data set "est_dummy" (for which you know the answer) (ii) returns NA if coverage = 95 and U95 and/or L95 are missing.

```
#usethis::use_testthat() # do this once

# now add your tests in tests/testthat/test_somename.R

# then check if your tests pass
devtools::test()
```

```
## i Testing hw2

## v | F W  S  OK | Context
## / |          0 | widthci_checkoutput                                          v |          1 | wid
## / |          0 | widthci_nahandling                                           v |          2 | wid
##
## == Results =====================================================================
## [ FAIL 0 | WARN 0 | SKIP 0 | PASS 3 ]
```