

The results below are generated from an R script.

```
#####  
# Epidemiologic Methods II  
# PHW250F, PHW250G, PH250B  
  
# Solutions: Iteration  
#####  
# Load okR autograder  
library(here)  
  
## here() starts at /Users/Nolan  
source("hw_iteration.ok.R")  
  
## Warning: package 'dplyr' was built under R version 3.5.2  
##  
## Attaching package: 'dplyr'  
## The following objects are masked from 'package:stats':  
##  
## filter, lag  
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union  
## Warning: package 'reticulate' was built under R version 3.5.2  
AutograderInit()  
  
# Let's load the dplyr package  
library(dplyr)  
  
# This assignment uses data from the WASH Benefits  
# Bangladesh trial. The trial assessed whether  
# water, sanitation, handwashing, and nutrition  
# interventions delivered separately or together  
# could reduce child diarrhea and/or improve child  
# growth. See Luby et al. 2018 for full details  
# (doi: http://dx.doi.org/10.1016/)  
  
# In this problem set we will calculate the incidence  
# density of diarrhea in different treatment arms after the  
# interventions were delivered.  
  
# The data and codebooks are publicly available here:  
# https://osf.io/pqzj5/  
  
# Make sure you downloaded the data from  
# the course site. Save the csv files in the same  
# location on your computer as the script hw_iteration.R  
# Then use the following commands to read in the data.  
  
# Load the diarrhea dataset:  
d = read.csv("washb-bangladesh-diar-public.csv")  
  
# Load the dataset with treatment variables:  
tr = read.csv("washb-bangladesh-tr-public.csv")
```

```

# Next let's merge the two datasets together.
# This will allow us to calculate the prevalence
# of diarrhea in different treatment arms
# (e.g., water, sanitation, handwashing, etc.)

d_tr = left_join(d, tr, by=c("block","clusterid"))

# Now let's filter to only keep the rows for diarrhea
# measurements after the interventions were delivered.
# The svy variable includes values 0, 1, 2. We are going
# to drop the 0 values, which indicate the time period
# before interventions were delivered.
d_tr = d_tr %>% filter(svy!=0)

# Now we are going to drop children with missing values
# in the diarrhea variable from the dataset. This
# assumes that they were missing at random - i.e.,
# that there are no characteristics associated with whether
# a child was missing diarrhea measurement.
d_tr = d_tr %>% filter(!is.na(diar7d))

# Take a look at the merged dataset:
head(d_tr)

```

##	dataid	childid	tchild	clusterid	block	svy	month	sex	agedays	ageyrs
## 1	28001	C1	Sibling	280	1	2	10	female	1537	4.208076
## 2	28001	T1	Target child	280	1	2	10	male	659	1.804244
## 3	28001	T1	Target child	280	1	1	9	male	268	0.733744
## 4	28001	C1	Sibling	280	1	1	9	female	1146	3.137577
## 5	28002	C1	Sibling	280	1	1	9	female	1525	4.175222
## 6	28002	C1	Sibling	280	1	2	10	female	1916	5.245722

##	enrolage	newbirth	sibnewbirth	gt36mos	d3plus2d	d3plus7d	dloose2d	dloose7d	dblood2d
## 1	1.9329227	0	0	0	0	0	0	0	0
## 2	-0.4709103	1	0	0	0	0	0	0	0
## 3	-0.4709103	1	0	0	0	0	0	0	0
## 4	1.9329227	0	0	0	0	0	0	0	0
## 5	2.9705681	0	0	0	0	0	0	0	0
## 6	2.9705681	0	0	0	0	0	0	0	0

##	dblood7d	diar2d	diar7d	bruise2d	bruise7d	tooth2d	tooth7d	svyweek	svyyear	tr
## 1	0	0	0	0	0	0	0	42	2015	Sanitation
## 2	0	0	0	0	0	0	0	42	2015	Sanitation
## 3	0	0	0	0	0	0	0	38	2014	Sanitation
## 4	0	0	0	0	0	0	0	38	2014	Sanitation
## 5	0	0	0	0	0	0	0	38	2014	Sanitation
## 6	0	0	0	0	0	0	0	42	2015	Sanitation

```

# Problem 1: Calculate the diarrhea prevalence in
# each treatment arm. Save the results in an object
# called prevalence_tr. It should have 7 rows (one
# for each treatment) and two columns. The first
# column should be for the treatment name and the
# second should be for the prevalence.

```

```
# (Hint: you did this in the homework assignment
# called hw_prev, and you can use the same code)
```

```
prevalence_tr = d_tr %>%
  group_by(tr) %>%
  summarise(prevalence=mean(diar7d))
prevalence_tr
```

```
## # A tibble: 7 x 2
##   tr          prevalence
##   <fct>          <dbl>
## 1 Control        0.0597
## 2 Handwashing    0.0394
## 3 Nutrition      0.0380
## 4 Nutrition + WSH 0.0377
## 5 Sanitation     0.0353
## 6 Water          0.0531
## 7 WSH            0.0423
```

```
# Check your answer
```

```
CheckProblem1()
```

```
## [1] "Correct!!"
```

```
## Problem 1: 1/1
```

```
# Problem 2: Write a function that converts
# prevalence to incidence density. Assume
# that the disease is rare. The function should
# be called calculate_inc_rare. It should take
# two arguments for prevalence (prev) and duration (d)
# of disease. It should return the estimated
# incidence density in person-days.
```

```
calculate_inc_rare=function(prev,d){
  id = prev / d
  return(id)
}
```

```
# Check your answer
```

```
# Note: the autograder will only check that
# you defined a function with the correct name
# and that the arguments are named correctly.
```

```
CheckProblem2()
```

```
## [1] "Correct!"
```

```
## Problem 2: 1/1
```

```
# Problem 3: Using dplyr (and not a for loop),
# add a column to the data frame
# prevalence_tr that has the incidence density
# in each arm assuming that the duration is 5 days.
```

```
# Use the function calculate_inc_rare to
# estimate the incidence density and name the column
# incidence.
```

```
prevalence_tr = prevalence_tr %>%
  mutate(incidence = calculate_inc_rare(prev = prevalence, d = 5))

# Check your answer
CheckProblem3()

## [1] "Correct!"
## Problem 3: 1/1
```

```
# Problem 4: Using dplyr (and not a for loop),
# calculate the incidence density difference
# (incidence density in each intervention arm minus
# the incidence density in the control arm).
# Save results in a column called inc_diff
# in the data frame prevalence_tr.
```

```
# To get you started, here is code to save
# the incidence density in the control arm as
# a scalar. You can subtract this scalar from the
# incidence density in each treatment arm.
control_inc = prevalence_tr %>%
  filter(tr=="Control") %>%
  select(incidence)
control_inc = as.vector(control_inc$incidence)

# calculate the incidence difference in each arm
prevalence_tr = prevalence_tr %>%
  mutate(inc_diff = incidence - control_inc)

# Check your answer
CheckProblem4()

## [1] "Correct!"
## Problem 4: 1/1
```

```
# Problem 5: Using a for loop (and not dplyr),
# calculate the incidence density difference
# (incidence density in each intervention arm minus
# the incidence density in the control arm).
# Save results in a vector called inc_diff_loop.
# Hint: the length of the vector should be 6.
# Do not include the control arm in the vector.
```

```
# The purpose of this problem is to show you
# that there is more than one way to do the same
# thing in R. This is an example of a situation
# in which most R users would probably prefer to
```

```
# use dplyr, but for your learning purposes, it is
# helpful to practice using a loop to see the pros
# and cons of each approach.
```

```
inc_diff_loop = vector(length=6)
for(i in 1:length(inc_diff_loop)){
  inc_diff_loop[i] = prevalence_tr$incidence[i+1] - control_inc
}

inc_diff_loop

## [1] -0.004058814 -0.004332022 -0.004390663 -0.004874400 -0.001318247 -0.003467338

# Check your answer
CheckProblem5()

## [1] "Correct!"
## Problem 5: 1/1
```

```
# Problem 6: Optional challenge question!

# In the data frame d_tr, columns d3plus2d through tooth7d
# contain indicators for whether each child
# experienced those symptoms in the past 7 days
# or the past 2 days.

# Write a for loop to take the mean of each of those
# columns, and save the results in a vector called
# symptoms. When taking the mean, you will need to
# use the option na.rm=TRUE in order to ignore
# missing values when calculating the mean.

# There is more than one way to approach this problem.
```

```
symptoms_df = d_tr %>%
  select(d3plus2d:tooth7d)

symptoms = vector(length = ncol(symptoms_df))
for(i in 1:length(symptoms)){
  symptoms[i] = mean(symptoms_df[,i], na.rm=TRUE)
}

# Check your answer
CheckProblem6()

## [1] "Correct!"
## Problem 6: 1/1
```

```
# Problem 7: Which symptoms were the most common?
# Enter your answer as a string scalar named p7
# including the column name in the quotes.
# (e.g., p7 = "dloose2d")
```

```
p7 = "d3plus7d"
```

```
# Check your answer
```

```
CheckProblem7()
```

```
## [1] "Correct!"
```

```
## Problem 7: 1/1
```

```
# Check your total score
```

```
MyTotalScore()
```

```
##
```

```
## Problem 1: 1/1
```

```
## Problem 2: 1/1
```

```
## Problem 3: 1/1
```

```
## Problem 4: 1/1
```

```
## Problem 5: 1/1
```

```
## Problem 6: 1/1
```

```
## Problem 7: 1/1
```

```
## Total Score: 7/7
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()
```

```
## R version 3.5.0 (2018-04-23)
```

```
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
```

```
## Running under: macOS Sierra 10.12.6
```

```
##
```

```
## Matrix products: default
```

```
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
```

```
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
```

```
##
```

```
## locale:
```

```
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

```
##
```

```
## attached base packages:
```

```
## [1] stats      graphics  grDevices  utils      datasets  methods    base
```

```
##
```

```
## other attached packages:
```

```
## [1] reticulate_1.12  dplyr_0.8.0.1    assertthat_0.2.0 checkr_0.3.0     rlist_0.4.6.1
```

```
## [6] jsonlite_1.6     here_0.1          knitr_1.22
```

```
##
```

```
## loaded via a namespace (and not attached):
```

```
## [1] Rcpp_1.0.0        magrittr_1.5      tidyselect_0.2.5  lattice_0.20-35
```

```
## [5] R6_2.4.0          rlang_0.3.1.9000  fansi_0.4.0       stringr_1.4.0
```

```
## [9] highr_0.7         tools_3.5.0       grid_3.5.0        data.table_1.11.4
```

```
## [13] xfun_0.5          utf8_1.1.4        cli_1.0.1         rprojroot_1.3-2
```

```
## [17] tibble_2.0.1      crayon_1.3.4      Matrix_1.2-17     purrr_0.3.1
```

```
## [21] glue_1.3.0        evaluate_0.13     stringi_1.3.1     compiler_3.5.0
```

```
## [25] pillar_1.3.1      backports_1.1.2   pkgconfig_2.0.2
```

```
Sys.time()
```

```
## [1] "2019-07-30 00:29:26 PDT"
```