```
                        # ECON-429 - Lab 1 -  Setting up, Importing,
Manipulating and Mapping Data in R

# 1. Create a folder on your laptop named "econ429labs"
# 2. Download the folder from Moodle with example datasets for todays
lab ("lab1"), and add it to the folder you created above
# 3. Install R and Rstudio
## Rstudio is an environment that we will use to do all the work
involving R
# 4. R script (code file), console, environment, files/plots/packages/
help/viewer
## Like "do files" in stata. You should do all you work within the R
script
## Creating objects
x <- 3
x <- "Texas"
z <- c("apple", "orange")
z
w <- list(x, y)
# Find an element of z (R is a vector based system)
z2 <- z[2]
w2 <- w[2]

# 5. Set the working directory (default folder) in this R document to
be lab1
setwd("/Users/mgebresilasse/Dropbox (Amherst College)/Courses/
202324_Spring/Econ429/econ429labs/lab1")

# 6. Installing and loading key packages in R
## In this course, we will be using six key packages in R:
"tidyverse","sf","tmap","raster","exactextractr", "fixest"
## Suppose you want to install "tidyverse". The standard way to do
this is as following:

install.packages("tidyverse")

## The above approach installs the package from CRAN (The
Comprehensive R Archive Network) which is a repository for R packages.
## Unlike Stata, each time you open a new R window you have to
(typically) add the specific packages you want to use to the library.
To do so:

library("tidyverse")

### You would generally load the packages you need at the beginning of
your code file.
### However, note that the tools in base R are always loaded
```

```
#### YOUR TURN: install and load "sf" into R

### Shortcut! There is a package called "pacman" the both installs (if
needed) and loads R packages in one step
### First install it
install.packages("pacman")

### After installing it once, then all you need to do for any new R
project is
pacman::p_load("tidyverse","sf","tmap")

### Note that in the above line we called the command "p_load" (load
package) without having to load pacman into R

##### YOUR TURN: install and/or load all the six packages

# 7. Getting help in R (?)
#### the ? command provides you the documentation page where you can
get specific information
#### about the package assuming you have downloaded it
?exactextractr

### Most package developers also provide vignettes about their
packages and common uses, which is availabel on
### CRAN, the package developer's github or webpage
### e.g.  Google "exactextractr cran"
### The package author also has a github page: https://github.com/
isciences/exactextractr
### Stack Exchange is a great resources -- typically someone has
already asked a question you are looking for
### Nick HK - https://nickchk.com/econometrics.html has great
resources ("R for Stata Users)
##### YOUR TURN: see how you can get help about "tmap"


# 8. Getting help outside of the formal channels (from people other
than the developer's)
### There is a lot of help online on how to particular tasks in R or
packages. Google is your best friend!
#### e.g. How to map points data in R?
####      How to map points data in R using tmap?


# 9. Reading and saving different data formats in R - RDS (base) & csv
(need tidyverse)
## Read and Save in RDS (base R)
eth_zonepop_2022_fromRDS <-
readRDS("ethiopia_population_by_zone_2022.rds")
#saveRDS(eth_zonepop_2022_fromRDS,
"ethiopia_population_by_zone_2022_new.rds")
```

```
## Read and save/write CSV (readr from tidyverse)
eth_zonepop_2022 <- read_csv("ethiopia_population_by_zone_2022.csv")
#write_csv(eth_zonepop_2022, "eth_pop_zone_2022_new.csv")

## Read and write dta (stata) files
eth_fulldemo <- read_dta("ethiopia_fulldemography_by_zone_2022.dta")
#write_dta(eth_fulldemo, "ethiopia_full_demo_new.dta")

# 10. Basic operations (piping %>%, select, rename, filter, mutate)
## first let's fix a conflict in command names from different packages
select <- dplyr::select

### piping (%>%) :   chains different commands (take this and then do
that)
### select : subsets the data columns (variables)
### filter : subsets the rows (observation) based on variables
### mutate : generates new columns

## Let's take the zone population data and then select only the
columns we need and rename the variables

ethzonepop <- eth_zonepop_2022 %>%

select(admin1Name_en,admin2Name_en,admin2Pcode,pop_total,pop_male,pop_
female) %>%

rename(region_name=admin1Name_en,zone_name=admin2Name_en,zone_code=adm
in2Pcode)

## Let's generate a dataset that has the sex ratio and keep only the
data for the Oromia region only
sexratio_oromia <- ethzonepop %>%  mutate(sexrat=pop_male/pop_female)
%>% filter(region_name=="Oromia")

#### YOUR TURN: generate a dataset containing columns with only the
zonename and female share of the puplation for only for zone in
####            in the "Somali" region

# 11. Reading/Importing Geospatial datasets
### Geospatial datasets can be stored in various format. The common
format is a "shapefile". However, once you have imported the shapefile
### into R, you can save it as an RDS file

#### Reading a map in a shapefile format
zonebound <-  st_read("eth_admbnda_adm2_csa_bofedb_2021.shp")
#### Reading a map in an RDS format (its like any other dataset)
roads <- readRDS("ethiopia_major_roads.rds")
#### Importing a raster dataset
### Let's download the data on malaria mortality for Ethiopia from the
```

```
Malaria Atlas project and add it to "lab1" folder
### https://malariaatlas.org/

malmort2020 <- raster("./malaria/
202206_Global_Pf_Mortality_Rate_ETH_2020.tiff")

#### Writing to a shapefile
#st_write(roads, "roads_new.shp")

### YOUR TURN: (1) import the hospital data in the lab1 folder and
save it as "hospitals"

# 12. Maping geospatial datasets (using various commands in"tmap")

## tmap has two modes ("view" and "plot"). If you use "plot", it
generates plot of the map that you can easily save to your folder
## if you want to explore the data interactively, the "view" mode
would be better
tmap_mode("plot")

## Mapping polygon/borders (tm_borders)
tm_shape(zonebound) + tm_borders()

# Let's swith back to the "view" mode
tmap_mode("view")
tm_shape(zonebound) + tm_borders()

## Mapping lines (tm_lines)
tm_shape(roads) + tm_lines()

## Mapping points (tm_dots) (in blue)
tm_shape(hospitals) + tm_dots("blue")

## Mapping multiple maps into one (just add them!). For instance, to
show the zone boundaries and roads together
tm_shape(zonebound) + tm_borders() + tm_shape(roads) + tm_lines("red")

## Mapping rasters
tm_shape(malmort2020) + tm_raster()
tm_shape(malmort2020) + tm_raster(style="quantile",n=5)

#### YOUR TURN: (1) map the zone borders in black, roads in purple and
hospitals in blue all together
####            (2) map the the location of hospitals and the malaria
mortality rate

# 13. Customization of plots
## The help documentation and vignettes are going to be key for you to
learn how to customize the maps as needed
```

```
# 14. You do all the operations with the geospatial datsets as we did
above
### Suppose we only want to plot the zone boundaries and hospitals and
roads in "Amhara" region
tm_shape(hospitals %>% filter(regname21=="Amhara")) + tm_dots("blue")

# 15. Merging Data (left_join, right_join, inner_join, full_join)
## Suppose we want to generate a map of the distribution of population
of children under 4 at the zone level

zonepopunder4 <-
read_xlsx("ethiopia_populationunder4_by_zone_2022.xlsx")

# Let's merge in the data on popunder 4 - each zone is uniquely
identified here by "admin2Pcode" which we had renamed zone_code
zonebound_withpop <- zonebound %>%
    left_join(zonepopunder4,by=c("ADM2_PCODE"="admin2Pcode"))

#### you can subset to only the variables you want to merge in and
combine select and rename as well
zonebound_withpop <- zonebound %>%
  left_join(zonepopunder4 %>%
select(ADM2_PCODE=admin2Pcode,popunder4),by=c("ADM2_PCODE"))

#### now plot the distribution of popunder4 (tm_fill)
tm_shape(zonebound_withpop) + tm_fill("popunder4")

# 16. Saving your maps as images
newmap <- tm_shape(zonebound_withpop) + tm_fill("popunder4")
#tmap_save(newmap,"map_popunder4_zone.png")


# 17. Let's put all of this together.
### YOUR TURN : Show a map of the sex ratio at the zone level (in
deciles) and location of hospitals in the "SNNP" region, and
###             then save it the "lab1" folder
```