**Working Paper CLIO-INFRA: Life Expectancy at Birth (Total)**

1. Title

- Life Expectancy per country (Average, total Population), 1500-2000.

2. Author(s)

- Version 2: Richard Zijdeman

- Version 1: Filipa Ribeiro da Silva.

3. Production date

- Version 2: 2014-07-26 (ISO date format, YYYY-MM-DD)

- Version 1: 03-09-2012

4. Version

- 2

5. Variable group(s)

- Demography.

6. Variable(s)

- Period Life Expectancy at birth for the total population.

7. Unit of analysis

- number of years.

8. Keywords (5)

- Demography, life expectancy, total population, world countries, mortality.

9. Abstract (200 words)

- This dataset provides Period Life Expectancy at birth per country and year. The overall aim of the dataset is to cover the entire world for the period 1500-2000. The current version (version 2) was build as part of the OECD “How was life” project. The dataset has nearly global coverage for the post 1950 period, while pre 1950 the coverage decreases the more historic the time period. Depending on sources, the data are annual estimates, 5 yearly or decadel estimates.

10. Time period

- 1500-2000.

11. Geographical coverage

- Entire World.

12. Methodologies used for data collection and processing

- Bibliographical research, research of published and Online Databases, and cross-analysis of various datasets

13. Data quality

i. Central statistical agencies

ii. Historical reconstructions

iii. Estimates

iv. Conjectures

14. Period of collection

- September 2013 – April 2014.

15. Data collectors

- Richard Zijdeman.

16. Sources

The sources used are:

* UN World Population Project: <http://esa.un.org/wpp/>
* Human Mortality Database: <http://www.mortality.org>
* GAPMINDER: <http://www.gapminder.org>
* OECD: <http://stats.oecd.org>
* Montevideo-Oxford Latin America Economic History Database: <http://www.lac.ox.ac.uk/moxlad-database>
* ONS: <http://www.ons.gov.uk/ons/datasets-and-tables/index.html>
* Australian Bureau of Statistics: <http://www.abs.gov.au/ausstats/abs@.nsf/web+pages/statistics?opendocument#from-banner=LN>
* Kannisto, V., Nieminen, M. & Turpeinen, O. (1999). Finnish Life Tables since 1751, *Demographic Research*, *1*(1), DOI: 10.4054/DemRes.1999.1.1

For specifics concerning (selections of) the sources, see the R-file below, with which the dataset was created.

17. Text

This data collection was carried out within the framework of the CLIO-INFRA project financed by the Netherlands Organisation for Scientific Research (NWO).

It is important to note that some data from version 1 have are not covered in version 2. Version 2 just exists of the afore mentioned (online) sources used for the OECD ‘How was life’ programme.

The data were created using an R-script (le\_total\_un16.R). Below is the code of the entire R-script:

## File: le\_total\_un15.R

## Date: 27092013

## Author: Richard Zijdeman

## Purpose: To extract all total life expectance (le) data from the UN/WPP

## Last change:

## version 16: just kept in rows from clio-infra format when merging OECD data,

## not both (search for "# changed in version 16 to:")

## version 15: 21 may and 21-26 july, 2014

## it appears that for the release for CLIO-INFRA, some decisions

## in version 14, need to be undone. E.g. data should not be limited to

## the 1820-2010 range

## as a result none of the reported numbers in this file make sense

## version 14: release for CLIO-INFRA

## version 13: recorded march 11, but the version is the one with which the

## paper was made for the OECD when Filipa was over for the EHPS conference at

## the IISH

## Version 12: 20 jan 2014

## - added UK data for part of 19th century

## - added AUS data for part of 19th century

## - added Finish data for 18th and 19th century

## - replaced label file with a combination of Auke's country label file

## and my own

## Version 11 was last written on Dec 2, 2013 at 08:55 hours

## Nov 26: Code edited to preserve Taiwan for Asia

## Oct 18: cleaned code

## Oct 3: changed path-typo: orignal into original (folder was named orignal)

# clean up

rm(list = ls()) # clear objects

graphics.off() # close graphics windows

# Loading Libraries #####

library("xlsx")

library("ggplot2")

library("reshape2")

library("plyr")

library("data.table")

# Setting Working Directory #####

setwd("/Users/richard/Dropbox/II/projects/lifexp/")

#############################################################################

# #

# Reading data from UN WPP : http://esa.un.org/wpp/ #

# #

#############################################################################

un <- read.xlsx("./data/original/un/WPP2012\_MORT\_F07\_1\_LIFE\_EXPECTANCY\_0\_BOTH\_SEXES\_20130927.xls",

sheetName="ESTIMATES", startRow=17, endRow=253)

# names(un)

names(un) <- c("index", "variant", "area.region.country", "notes",

"country.code", "letot.1950","letot.1955","letot.1960",

"letot.1965", "letot.1970","letot.1975","letot.1980",

"letot.1985","letot.1990", "letot.1995","letot.2000",

"letot.2005")

# head(un)

un.dt <- data.table(un)

# names(un.dt)

for (i in 1:4) {

un.dt[, paste0('letot.195', i) := letot.1950]

un.dt[, paste0('letot.196', i) := letot.1960]

un.dt[, paste0('letot.197', i) := letot.1970]

un.dt[, paste0('letot.198', i) := letot.1980]

un.dt[, paste0('letot.199', i) := letot.1990]

un.dt[, paste0('letot.200', i) := letot.2000]

}

for (i in 6:9) {

un.dt[, paste0('letot.195', i) := letot.1955]

un.dt[, paste0('letot.196', i) := letot.1965]

un.dt[, paste0('letot.197', i) := letot.1975]

un.dt[, paste0('letot.198', i) := letot.1985]

un.dt[, paste0('letot.199', i) := letot.1995]

un.dt[, paste0('letot.200', i) := letot.2005]

}

sort(names(un.dt))

# head(un.dt)

#

# names(un.dt)

# summary(un.dt$letot.1950)

# summary(un.dt$letot.1951)

# summary(un.dt$letot.1952)

# summary(un.dt$letot.1953)

# summary(un.dt$letot.1954)

# un.dt[1:100, ]

un <- as.data.frame(un.dt)

# summary(un$variant) # constant string "Estimates"

un$variant <- NULL

# head(un)

un.long <- reshape(un, direction = "long", varying = grep("letot.", names(un)))

#head(un.long)

un.long$source <- "UN\_WPP2013"

# head(un.long)

# table(un.long$time) # 236 obs per year

#############################################################################

# #

# Reading data from HMD : http://www.mortality.org #

# #

#############################################################################

# Reading in all files automatically and properly ####

filenames <- as.list(list.files("./data/original/hmd/e0\_per\_20130926/E0per"))

hmd <- do.call("rbind",

lapply(filenames, function(x) {

dat <- read.table(paste0(

"./data/original/hmd/e0\_per\_20130926/E0per/", x),

header = TRUE,

sep = "",

dec = ".",

skip = 2

)

dat$Country <- gsub(".E0per.txt", "", x)

# numeric are annoyingly converted to character

dat$Female <- as.numeric(as.character(dat$Female))

dat$Male <- as.numeric(as.character(dat$Male))

dat$Total <- as.numeric(as.character(dat$Total))

dat

}))

# str(hmd)

names(hmd) <- tolower(names(hmd))

hmd$source <- "Human Mortality Database"

#head(hmd)

#str(hmd)

#dim(hmd) # 4332 rows 6 columns

hmd.sub1 <- subset(hmd, select = c("year", "total", "country", "source"))

# head(hmd.sub1)

# unique(hmd.sub1$country)

# [1] "AUS" "AUT" "BEL" "BGR" "BLR" "CAN" "CHE"

# [8] "CHL" "CZE" "DEUTE" "DEUTNP" "DEUTW" "DNK" "ESP"

# [15] "EST" "FIN" "FRACNP" "FRATNP" "GBR\_NIR" "GBR\_NP" "GBR\_SCO"

# [22] "GBRCENW" "GBRTENW" "HUN" "IRL" "ISL" "ISR" "ITA"

# [29] "JPN" "LTU" "LUX" "LVA" "NLD" "NOR" "NZL\_MA"

# [36] "NZL\_NM" "NZL\_NP" "POL" "PRT" "RUS" "SVK" "SVN"

# [43] "SWE" "TWN" "UKR" "USA"

## reading in file with labels for hmd and un merge

labels <- read.xlsx("./data/derived/labels/hmd2uncountrycode\_1.xlsx",

sheetName="labels")

## merging labels with hmd.sub1

hmd.sub2 <- merge(hmd.sub1, labels,

by.x = "country",

by.y = "hmd.country.label")

## did all countries receive a un.country.label / un.country.code?

names(hmd.sub2)

unique(hmd.sub2[is.na(hmd.sub2$un.country.label),"country"])

# "FRACNP" "GBR\_NIR" "GBR\_SCO" "GBRCENW" "GBRTENW" "NZL\_MA" "NZL\_NM" "TWN"

# it's o.k. that these values are missing

# dropping them from the sample

# names(hmd.sub2)

# hmd.sub2.2bdropped <- subset(hmd.sub2, is.na(un.country.code))

# dim(hmd.sub2.2bdropped) # 994 6

# 994 cases, checks out allright

hmd.sub2 <- subset(hmd.sub2, !is.na(un.country.code))

# dim(hmd.sub2)

# 3338 + 994 yep, checks out fine

head(hmd.sub2)

head(un.long)

un.long.sub1 <- subset(un.long, select = c("area.region.country", "notes",

"country.code", "time", "letot",

"source"))

# names(hmd.sub2)

# names(un.long.sub1)

## making varnames equal in order to use rbind

names(hmd.sub2) <- c("country", "time", "letot", "source",

"area.region.country", "country.code")

# head(hmd.sub2)

# removing and adding variables in order to use rbind

hmd.sub3 <- hmd.sub2[,-1]

# head(hmd.sub3)

hmd.sub3$notes <- NA

hmd.sub3$notes <- NA

# names(hmd.sub3)

# names(un.long.sub1)

# head(hmd.sub3)

# head(un.long.sub1)

# binding hmd.sub3 and un.long.sub1 (adding them as rows)

un.hmd <- rbind(hmd.sub3, un.long.sub1)

# names(un.hmd)

# un.hmd[200:700,]

hmd.countries <- c(36, 40, 56, 100, 112, 124, 152, 203, 208, 233, 246, 250, 276,

348, 352, 372, 376, 380, 392, 428, 440, 442, 528, 554, 578,

616, 620, 643, 703, 705, 724, 752, 756, 804, 826, 840)

un.hmd.both <- subset(un.hmd, country.code %in% hmd.countries)

names(un.hmd.both)

## graphical check of overall comparability of data #####

# p <- ggplot(un.hmd.both, aes(time,letot, colour = source)) +

# geom\_line()

# # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# p

# p1 <- p +

# facet\_wrap(~ area.region.country, ncol = 8) +

# theme\_bw() +

# geom\_vline(xintercept=c(1914,1918,1939,1945),

# colour="green",

# linetype = "longdash")

# p1

# # Both sources (Human Mortality Database and UN\_WPP2013) look insanely similar

# # UN is somewhat smoother, but remember that those data are based on 5-year

# # points.

#

# q <- ggplot(subset(un.hmd.both,

# area.region.country %in% "United States of America"),

# aes(time,letot, colour = source)) +

# geom\_line()

# q

#

# q1 <- ggplot(subset(un.hmd.both,

# area.region.country %in% "Netherlands"),

# aes(time,letot, colour = source)) +

# geom\_line()

# q1

#

# head(un.hmd.both)

## HLD - NOT REALLY ####

### Ok, so this goes down really well. Let's add the human lifetable database

### I am using a version Auke 'scraped' from the website using R:

### see 'lifeexp.r' file in data/original/hld/

# hld stands for human lifetable database

## commented out, lines below, since not using (read below why)

# hld <- read.csv("./data/original/hld/lifetable\_de\_abridged\_oct2012.csv")

# names(hld)

# table(hld$sex) #yikes, only men and women, how to combine to total?

# leaving for now, perhaps to be picked upon in a later stage

# hld data is however also available through gapminder, so perhaps

# there it is already provided as 'total' life expectancy (e.g. man+women)

## Next add new data sources from GAPMINDER

## WARNING: GAPMINDER says not to use their own estimates!!! So only using

## selected sources: Riley (preferred),

# the following doesn't work: memmory issues ("Java heap space")

# gap <- read.xlsx("./data/original/gapminder/gapdata004\ v6.xlsx",

# sheetName="Data & meta data", startColumn = 1,

# endColumn = 5)

## GAPMINDER ####

gap <- read.csv("./data/original/gapminder/gapdata004\ v6.csv")

# downsizing the file

# names(gap)

# str(gap)

# dropping all data after 2010

gap.sub1 <- gap #subset(gap, Year <= 2010) # changed in v15

# dropping 'questionable' sources, e.g. gapminder models

unique(gap.sub1$Our.source..see.next.tab.for.explanations)

gap.sub1$source <- as.character(

gap.sub1$Our.source..see.next.tab.for.explanations)

gap.sub2 <- subset(gap.sub1,

source == "Riley file" |

source == "HLD" |

source == "Estonian Interuniversity Population Research Centre" |

source == "Estimate from Boyle & O Grada (1986)." |

source == "Administrator of Tokelau (2006)" |

source == "Wrigley et al. 1997")

# removing data from HLD based on life table type 1

names(gap.sub2)

gap.sub3 <- subset(gap.sub2, gap.sub2$Other.notes != "Life table type 1 used, since type 3 was not available (see Gapminder documentation 004 or methodology note at www.lifetable.de). Life expectancy aggregated from male/female data assuming a sex ratio of 0.512/0.488)")

names(un.hmd)

names(gap.sub3)

gap.sub4 <- subset(gap.sub3,

select = c("Year", "Area", "Life.expectancy.at.birth",

"source", "Other.notes"))

# names(un.hmd)

# names(gap.sub4)

gap.sub5 <- gap.sub4

names(gap.sub5) <- c("time", "area.region.country", "letot", "source", "notes")

gap.sub5$country.code <- NA # this is necessary for rbind

gap.sub5$area.region.country.old <- gap.sub5$area.region.country

# this step is necessary to merge UN labels and rename the labels

# area.region.country

names(gap.sub5)

gap.sub6 <- gap.sub5[,-2] #drop column 2 (area.region.country)

names(gap.sub6)

gap.un.labels <-

read.xlsx("./data/derived/labels/gapminder2selected\_un\_country\_labels01.xlsx",

sheetName = "label")

#names(gap.sub6)

#names(gap.un.labels)

#gap.sub7 <- merge(gap.sub6, gap.un.labels,

# by.x = "area.region.country.old",

# by.y = "gapminder.country.label",

# all.x = TRUE)

# This didn't work: lots of whitespace in area.region.country.old

trim <- function (x) gsub("^\\s+|\\s+$", "", x) # trims leading/trailng white-

# space

#names(gap.sub6)

gap.sub6$area.region.country.old <- trim(gap.sub6$area.region.country.old)

#names(gap.un.labels)

gap.un.labels$gapminder.country.label <-

trim(gap.un.labels$gapminder.country.label)

gap.sub7 <- merge(gap.sub6, gap.un.labels,

by.x = "area.region.country.old",

by.y = "gapminder.country.label",

all.x = TRUE)

names(gap.sub7)

table(gap.sub7$un.country.code) # perfect match on y

unique(gap.sub7$un.country.code)

# why the NA?

temp.na <- subset(gap.sub7, is.na(un.country.code))

temp.na

# Code edit: Nov 27: Preserve Taiwan for Asia

unique(temp.na$area.region.country.old)

taiwan\_string <- unique(temp.na$area.region.country.old)[1]

taiwan\_string

gap.sub7a <- gap.sub7

names(gap.sub7a)

unique(gap.sub7a$un.country.code)

unique(gap.sub7a$un.country.label)

gap.sub7a$un.country.code[gap.sub7a$area.region.country.old == taiwan\_string] <-

9991

gap.sub7a$country.code[gap.sub7a$area.region.country.old == taiwan\_string] <-

9991

levels(gap.sub7a$un.country.label) <- c(levels(gap.sub7a$un.country.label),

taiwan\_string)

# this adds taiwan as a factor level

gap.sub7a$un.country.label[gap.sub7a$area.region.country.old == taiwan\_string] <-

taiwan\_string

gap.sub7a[gap.sub7a$area.region.country.old == taiwan\_string, ]

table(gap.sub7a$un.country.label)

# That's Taiwan and Tokelau <- not UN recognized, dropping these obs

gap.sub8 <- gap.sub7a #removed in v15:subset(gap.sub7a, !is.na(un.country.code))

# only 1 obs dropped:

# dim(gap.sub7a)[1] - dim(gap.sub8)[1]

names(un.hmd)

gap.sub9 <- subset(gap.sub8, select = -c(area.region.country.old, country.code))

#- deselects

names(gap.sub9)

names(gap.sub9) <- c("time", "letot", "source", "notes", "area.region.country",

"country.code")

head(gap.sub9)

## quick check on data:

# gp <- ggplot(gap.sub9, aes(time,letot, colour = source)) +

# geom\_line()

# # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# gp

# gp1 <- gp +

# facet\_wrap(~ area.region.country, ncol = 8) +

# theme\_bw() +

# geom\_vline(xintercept=c(1914,1918,1939,1945),

# colour="green",

# linetype = "longdash")

# gp1

### checks out ok

## ok, now adding gapminder data to un.hmd file

#names(un.hmd)

#names(gap.sub9)

un.hmd.gap <- rbind(un.hmd, gap.sub9)

# CREATING INTERSECTION OF THE DATA: TO SEE WHAT IS COMMON AND

# HOW COMMON IT IS

# below commented out in v 15

# ## list.un.hmd.gap <-un.hmd.both

# # todo: make a list of countries that are in un/hmd/gap

# clist.un <- as.list(unique(un.long.sub1$country.code))

# clist.hmd <- as.list(unique(hmd.sub3$country.code))

# clist.gap <- as.list(unique(gap.sub9$country.code))

# clist.un.hmd.gap <- Reduce(intersect, list(clist.un, clist.hmd, clist.gap))

# # intersect takes common values

# # Reduce is needed, since intersect only takes 2 values

#

# un.hmd.gap.intersect <- subset(un.hmd.gap, country.code %in% clist.un.hmd.gap)

#

#

# ## quick check on data:

# # uhg.i <- ggplot(un.hmd.gap.intersect, aes(time,letot, colour = source)) +

# # geom\_line()

# # # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# # #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# # uhg.i

# # uhg.i1 <- uhg.i +

# # facet\_wrap(~ area.region.country, ncol = 6) +

# # theme\_bw() +

# # geom\_vline(xintercept=c(1914,1918,1939,1945),

# # colour="green",

# # linetype = "longdash")

# # uhg.i1

#

#

# un.hmd.gap.intersect.2 <- subset(un.hmd.gap.intersect, time >= 1800)

# # I do this to remove early England data, which crops the data.

# # uhg.i2 <- ggplot(un.hmd.gap.intersect.2, aes(time,letot, colour = source)) +

# # geom\_line()

# # # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# # #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# # uhg.i2

# # uhg.i3 <- uhg.i2 +

# # facet\_wrap(~ area.region.country, ncol = 6) +

# # theme\_bw() +

# # geom\_vline(xintercept=c(1914,1918,1939,1945),

# # colour="green",

# # linetype = "longdash") +

# # theme(axis.text.x = element\_text(angle = 45, hjust = 1))

# # uhg.i3

#

# # ppi <- 300 # make a constant defining number of pixels later on

# # png("./analyses/graphs/un\_hmd\_gap\_intersect01.png",

# # , width = 12\*ppi, height = 8\*ppi, res = ppi)

# # uhg.i3

# # dev.off()

# # ##

#

# ## graphing all countries (probably useless)

# # un.hmd.gap.all <- ggplot(un.hmd.gap, aes(time,letot, colour = source)) +

# # geom\_line()

# # # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# # #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# # un.hmd.gap.all

# # un.hmd.gap.all1 <- un.hmd.gap.all +

# # facet\_wrap(~ area.region.country, ncol = 15) +

# # theme\_bw() +

# # geom\_vline(xintercept=c(1914,1918,1939,1945),

# # colour="green",

# # linetype = "longdash")

# # un.hmd.gap.all1

### OECD DATA

oecd.raw <-

read.csv("./data/original/oecd/Health Status-1/HEALTH\_STAT\_Data\_27976937-3d52-4d15-90a1-bc1abd1c4147.csv")

names(oecd.raw)

table(oecd.raw$Flags)

# Flags: two flags: b and e, adding meaning

oecd.raw$notes <- as.character(oecd.raw$Flags)

trim <- function (x) gsub("^\\s+|\\s+$", "", x) # trims leading/trailng white-

# space

oecd.raw$notes <- trim(oecd.raw$notes)

oecd.raw$notes <- gsub("b", "b: break in series", oecd.raw$notes)

oecd.raw$notes <- gsub("e", "e: estimate", oecd.raw$notes)

# table(oecd.raw$notes)

# selecting only "Total Population at birth" (e.g. not men/women)

oecd.raw$Variable <- trim(oecd.raw$Variable)

oecd.raw.sub1 <- subset(oecd.raw, Variable == "Total population at birth")

# names(oecd.raw.sub1)

# dropping unneeded vars: Variable, Unit and Flags

oecd.raw.sub2 <- oecd.raw.sub1[,-c(1, 2, 6)]

names(oecd.raw.sub2)

names(oecd.raw.sub2) <- c("country.oecd", "time", "letot", "notes" )

names(oecd.raw.sub2)

# adding source

oecd.raw.sub2$source <- "OECD"

# changing country.oecd from factor to character in order to merge later on

# str(oecd.raw.sub2)

oecd.raw.sub2$country.oecd <- as.character(oecd.raw.sub2$country.oecd)

# str(oecd.raw.sub2)

# now to add un country labels and numbers

unique(oecd.raw.sub2$country.oecd)

un.labels <- read.xlsx("./data/derived/labels/un\_labels\_2013\_4.xls",

sheetName = "labels")

# names(un.labels)

# str(un.labels)

un.labels$area.region.country <- as.character(un.labels$area.region.country)

un.labels$region.sort.label <- as.character(un.labels$region.sort.label)

# str(un.labels)

# str(oecd.raw.sub2)

oecd.raw.sub3 <- merge(oecd.raw.sub2, un.labels,

by.x = "country.oecd",

by.y = "area.region.country",

all.x = TRUE)

# names(oecd.raw.sub3)

## NB: area.region.country var is dropped (overlaps with country.oecd)

# checking how succesfull this was (e.g. how many observations 'matched')

table(is.na(oecd.raw.sub3$country.code))

# so 147 entries were not coded. which are these?

temp.not.coded <- subset(oecd.raw.sub3, is.na(oecd.raw.sub3$country.code))

# unique(temp.not.coded$country.oecd)

# [1] "Korea" "Slovak Republic" "United States"

# ok, so these 3 countries need to changed manually

oecd.raw.sub3$country.code[oecd.raw.sub3$country.oecd == "United States"] <- 840

oecd.raw.sub3$region.sort.label[oecd.raw.sub3$country.oecd ==

"United States"] <- "na.2"

oecd.raw.sub3$country.oecd[oecd.raw.sub3$country.oecd ==

"United States"] <- "United States of America"

oecd.raw.sub3$country.code[oecd.raw.sub3$country.oecd == "Korea"] <- 410

oecd.raw.sub3$region.sort.label[oecd.raw.sub3$country.oecd ==

"Korea"] <- "eas.7"

oecd.raw.sub3$country.oecd[oecd.raw.sub3$country.oecd ==

"Korea"] <- "Republic of Korea"

# the fact that this is south korea and not north korea is derived from

# http://www.oecd-ilibrary.org/economics/country-statistical-profile-korea\_20752288-table-kor

# see at the bottom the note: Countries covered: Korea, Republic of

oecd.raw.sub3$country.code[oecd.raw.sub3$country.oecd ==

"Slovak Republic"] <- 703

oecd.raw.sub3$region.sort.label[oecd.raw.sub3$country.oecd ==

"Slovak Republic"] <- "eeu.9"

oecd.raw.sub3$country.oecd[oecd.raw.sub3$country.oecd ==

"Slovak Republic"] <- "Slovakia"

unique(oecd.raw.sub3$country.oecd)

# ok, so this checks out. names have changed.

temp.not.coded2 <- subset(oecd.raw.sub3, is.na(oecd.raw.sub3$country.code))

# zero obs, good.

temp.not.coded2 <- subset(oecd.raw.sub3, is.na(oecd.raw.sub3$region.sort.label))

# zero obs, good.

# last: change name of country.oecd

names(oecd.raw.sub3)[1]

names(oecd.raw.sub3)[1] <- "area.region.country"

names(oecd.raw.sub3)

unique(oecd.raw.sub3$area.region.country)

# Next merge this with the overall data file

# need to drop region.sort.label first

names(oecd.raw.sub3)

oecd.raw.sub4 <- oecd.raw.sub3[,-c(7,8)]

names(oecd.raw.sub4)

names(un.hmd.gap)

un.hmd.gap.oecd <- rbind(un.hmd.gap, oecd.raw.sub4)

# commented out in version 15

# # check: how do OECD figures differ from other estimates

# oecd.names.df <- unique(subset(un.hmd.gap.oecd,

# source == "OECD",

# select = c(source, area.region.country)))

# oecd.names <- as.character(oecd.names.df$area.region.country)

# oecd.names

#

# un.hmd.gap.oecd.sub1 <- subset(un.hmd.gap.oecd,

# area.region.country %in% oecd.names)

# un.hmd.gap.oecd.sub2 <- subset(un.hmd.gap.oecd.sub1, time >= 1800)

#

# # uhgo <- ggplot(un.hmd.gap.oecd.sub2, aes(time,letot, colour = source)) +

# # geom\_line(alpha=.7)

# # # geom\_point(colour = "blue", size = .5) + # bad for quality of pic

# # #scale\_y\_continuous(limits = c(10,90), breaks = seq(10,90,10) )

# # #uhgo

# # uhgo1 <- uhgo +

# # facet\_wrap(~ area.region.country, ncol = 9) +

# # theme\_bw() +

# # labs(x = "Time (years)", y = "Life Expectancy, Total population",

# # title = "Life Expectancy in OECD and other sources") +

# # geom\_vline(xintercept=c(1914,1918,1939,1945),

# # colour = "green",

# # linetype = "longdash") +

# # theme(legend.position = "bottom",

# # axis.text.x = element\_text(angle = 45, hjust = 1))

# # uhgo1

#

# # ppi <- 300 # make a constant defining number of pixels later on

# # png("./analyses/graphs/un\_hmd\_gap\_oecd\_comparison01.png",

# # , width = 12\*ppi, height = 8\*ppi, res = ppi)

# # uhgo1

# # dev.off()

write.csv(un.hmd.gap.oecd,

file = "./data/derived/clio\_infra/un\_hmd\_gap\_oecd\_20140521.csv")

## NB: something went wrong with names of Human Mortality Database

# reading in moxlad data

# Montevideo-Oxford Latin American Economic History Data Base

moxlad <- read.xlsx("./data/original/moxlad/moxlad20131002.xlsx",

sheetName = "moxlad", startRow = 3, endRow = 145)

# head(moxlad)

names(moxlad)[1] <- "time" # give name to first column (time/year)

# names(moxlad)

moxlad.sub1 <- subset(moxlad, select = -c(grep("NA..", names(moxlad))))

# - in select indicates 'drop'

# - grep: grep all column names with "NA.." (in the name)

# remove redundant first row

# head(moxlad.sub1)

moxlad.sub2 <- moxlad.sub1[-1,]

# head(moxlad.sub2)

# tail(moxlad.sub2)

# selecting years with information: (moxlad data are only for every 5 years)

years <- c(1900, 1910, seq(1920, 1990, 5), seq(1991, 1999, 1), 2000)

moxlad.sub3 <- subset(moxlad.sub2, time %in% years)

str(moxlad.sub3)

## melt data in right format

names(moxlad.sub3)

moxlad.sub3.molten <- melt(moxlad.sub3, id.vars = "time",

variable.name = "country", value.name = "letot")

# removing NA observations

moxlad.sub4 <- subset(moxlad.sub3.molten, !is.na(letot))

# replacing comma's in letot values dots

moxlad.sub4$letot <- gsub(",",".", moxlad.sub4$letot)

moxlad.sub4$letot <- as.numeric(as.character(moxlad.sub4$letot))

moxlad.sub4$time <- as.numeric(as.character(moxlad.sub4$time))

# str(moxlad.sub4)

# combining moxlad with un.hmd.gap.oecd

# names(un.hmd.gap.oecd)

moxlad.sub5 <- moxlad.sub4

moxlad.sub5$source <- "MOxLAD"

# head(moxlad.sub5)

@ unique(moxlad.sub5$country)

# [1] Argentina Bolivia Brazil Chile

# [5] Colombia Costa.Rica Cuba Dominican.Republic

# [9] Ecuador El.Salvador Guatemala Haiti

# [13] Honduras Mexico Nicaragua Panama

# [17] Paraguay Peru Uruguay Venezuela

# 20 Levels: Argentina Bolivia Brazil Chile Colombia Costa.Rica ... Venezuela

## changing names in order to match UN labels

moxlad.sub5$country <- gsub("Bolivia", "Bolivia (Plurinational State of)",

moxlad.sub5$country)

moxlad.sub5$country <- gsub("Costa.Rica", "Costa Rica",

moxlad.sub5$country)

moxlad.sub5$country <- gsub("Dominican.Republic", "Dominican Republic",

moxlad.sub5$country)

moxlad.sub5$country <- gsub("El.Salvador", "El Salvador",

moxlad.sub5$country)

moxlad.sub5$country <- gsub("Venezuela", "Venezuela (Bolivarian Republic of)",

moxlad.sub5$country)

moxlad.sub6 <- merge(moxlad.sub5, un.labels,

by.x = "country",

by.y = "area.region.country",

all.x = TRUE)

# this worked perfectly

# NB: moxlad may contain values from other databases for recent periods

## add moxlad to main.database

names(moxlad.sub6)

names(un.hmd.gap.oecd)

moxlad.sub7 <- moxlad.sub6

names(moxlad.sub7)

names(moxlad.sub7)[1] <- "area.region.country"

names(moxlad.sub7)

moxlad.sub7 <- moxlad.sub7[,-c(6,7)]

moxlad.sub7$notes <- NA

names(moxlad.sub7)

names(un.hmd.gap.oecd)

un.hmd.gap.oecd.moxlad <- rbind(un.hmd.gap.oecd, moxlad.sub7)

tail(un.hmd.gap.oecd.moxlad)

## Adding data, insert Jan 21, 2014 ####

## Adding early data for UK, Australia and Finland

ons2008 <- read.xlsx("./data/original/ons/figures1and2\_tcm77-291936.xlsx",

sheetName = "Figures 1 and 2",

colIndex = c(2:3,8),

startRow = 10,

endRow = 179)

# ons2008

names(ons2008) <- c("time", "male.le", "female.le")

# ons2008

ons2008$letot <- (ons2008$male.le + ons2008$female.le)/2

ons2008$source <- "ONS 2008"

ons2008$area.region.country <- "United Kingdom"

ons2008$country.code <- 826

ons2008$notes <- "Unweighted average of men/women for England and Wales"

# NB: unweighted average: proper average should be weighted by pop size

ons2008 <- ons2008[, -c(2,3)] # dropping male and female data

head(ons2008)

un.hmd.gap.oecd.moxlad.ons <- rbind(un.hmd.gap.oecd.moxlad, ons2008)

tail(un.hmd.gap.oecd.moxlad.ons)

## Adding Australian data from ONS2008

# these are decadal averages for 1880-1890, 1891-1900, 1901-1910

# male averages: 47.2 51.1 55.2

# female averages: 50.9 54.8 58.8

# unweighted average total population

abs2008 <- as.data.frame(

c(mean(c(47.2, 50.9)), mean(c(51.1, 54.8)), mean(c(55.2, 58.8)) ))

names(abs2008) <- "letot"

abs2008$time <- c(1885, 1895, 1905) # NB: decadal averages!!

abs2008$source <- "Australian Bureau of Statistics 2008"

abs2008$area.region.country <- "Australia"

abs2008$country.code <- 36

abs2008$notes <- "Decadal averages of unweighted male/female life expectancy"

abs2008

un.hmd.gap.oecd.moxlad.ons.abs <- rbind(un.hmd.gap.oecd.moxlad.ons, abs2008)

tail(un.hmd.gap.oecd.moxlad.ons.abs)

## hier verder met Finland !!!! XXXXXX ###

min(un.hmd.gap.oecd.moxlad.ons.abs[

un.hmd.gap.oecd.moxlad.ons.abs$area.region.country=="Finland",

"time"])# 1875

un.hmd.gap.oecd.moxlad.ons.abs[

un.hmd.gap.oecd.moxlad.ons.abs$area.region.country=="Finland",

c("time", "letot")]

un.hmd.gap.oecd.moxlad.ons.abs[

un.hmd.gap.oecd.moxlad.ons.abs$area.region.country=="Finland",

"country.code"]

# source: Kannisto, Turpeinen & Nieminen (1999) Demographic Research, 1(1)

# DOI: 10.4054/DemRes.1999.1.1

# decadal averages:

# 1751-1760 -> 1755

time <- c(1755, 1765, 1775, 1785, 1795,

1805, 1815, 1825, 1835, 1845, 1855, 1865)

le.male <- c(36.1, 33.9, 38.3, 32.4, 35.6,

30.4, 34.5, 36.6, 33.0, 38.4, 33.4, 30.7)

le.female <- c(38.4, 35.8, 40.4, 35.2, 38.5,

33.2, 37.1, 40.2, 36.3, 42.5, 36.8, 33.5)

kannisto99 <- as.data.frame(cbind(time, le.male, le.female))

kannisto99$letot <- (le.male+le.female)/2

kannisto99$source <- "Kannisto, Nieminen and Turpeinen (1999)"

kannisto99$notes <-

"Decadal averages of unweighted male/female life expectancies"

kannisto99$area.region.country <- "Finland"

kannisto99$country.code <- "246"

kannisto99

names(kannisto99)

kannisto99 <- kannisto99[, -c(2,3)]

names(kannisto99)

un.hmd.gap.oecd.moxlad.ons.abs.kann <-

rbind(un.hmd.gap.oecd.moxlad.ons.abs, kannisto99)

tail(un.hmd.gap.oecd.moxlad.ons.abs.kann)

### /end insert Jan 21, 2014

## testing whether UN labels can be added to this file ####

main.df <- un.hmd.gap.oecd.moxlad.ons.abs.kann

names(main.df)

table(main.df$country.code, useNA = "ifany") # v15 1 NA's.

# see le\_total\_un03.R for version where there were missings

# in this version solved: east + west germany got country codes

# other countries are not recognized or have specific samples

# that are not suitable: these are left out

#names(main.df)

# Fri 9 May, 2014: added "copy" to filename below .

# write.csv(main.df,

# file = "./data/derived/clio\_infra/un\_hmd\_gap\_oecd\_moxlad\_20140121\_copy.csv")

# Fri 21 May, 2014:

write.csv(main.df,

file = "./data/derived/clio\_infra/un\_hmd\_gap\_oecd\_moxlad\_20140521.csv")

# NB: this file still contains multiple measurement for country / time

# fixing that below and writing out under new file name

#head(main.df)

#dim(main.df)

#table(is.na(main.df$country.code))

#table(is.na(main.df$area.region.country)) # no missings

##########################################################

## Now reduce file: from multiple observations per year ##

## to one observation per year ##

##########################################################

un.oecd.labels <- read.xlsx("./data/derived/labels/un\_labels\_2013\_5.xls",

sheetName = "labels")

# names(main.df)

# names(un.oecd.labels)

main.df.1 <- merge(main.df, un.oecd.labels,

by.x = "country.code",

by.y = "country.code",

all.x = TRUE)

table(is.na(main.df.1$country.code), useNA = "ifany")

# which code is missing?

main.df.1[is.na(main.df.1$country.code), ]

# Tokelau: dropping that one

main.df.1 <- subset(main.df.1, source != "Administrator of Tokelau (2006)")

# dim(main.df.1)

# names(main.df.1)

main.df.1[main.df.1$country.code==9991,]

# names(main.df.1)

# removing area.region.country.y (.y comes from merge with un.oecd.labels)

names(main.df.1)

main.df.2 <- main.df.1[, -7]

names(main.df.2)

names(main.df.2)[5] <- "area.region.country"

names(main.df.2)

###### remove duplicate entries per country

# Q1: which country-year combinations have multiple entries?

main.dt.2 <- data.table(main.df.2)

main.dt.2[, mult.obs := .N, by = c("country.code", "time")]

table(main.dt.2$mult.obs, useNA = "ifany")

# 1 2 3 4

# 13915 2314 3885 144

# which sources do the multiple obs belong to?

table(subset(main.dt.2, mult.obs>1, select=source), useNA = "ifany")

# Human Mortality Database MOxLAD

# 1984 257

# OECD ONS 2008

# 1541 89

# Riley file UN\_WPP2013

# 60 2412

# which time periods do the multiple obs belong to?

table(subset(main.dt.2, mult.obs>1, select=time), useNA = "ifany")

# which countries do the multiple obs belong to?

table(subset(main.dt.2, mult.obs>1, select=area.region.country),

useNA = "ifany")

# before continuing with the multiple observations, I will now first check

# from which the multiple obs are coming from

table(main.dt.2$mult.obs, useNA = "ifany")

main.dt.2.sobs <- subset(main.dt.2, mult.obs==1)

dim(main.dt.2.sobs)

unique(main.dt.2.sobs[,source])

# [1] "Riley file"

# [2] "Human Mortality Database"

# [3] "UN\_WPP2013"

# [4] "MOxLAD"

# [5] "OECD"

# [6] "Estonian Interuniversity Population Research Centre"

# [7] "Kannisto, Nieminen and Turpeinen (1999)"

# [8] "HLD"

# [9] "Australian Bureau of Statistics 2008"

# [10] "ONS 2008"

# [11] "Wrigley et al. 1997"

main.dt.2.mobs <- subset(main.dt.2, mult.obs>1)

# main.dt.2.mobs

# main.dt.2.mobs[1:30,]

# trying to come to a unique set of data, through elimination

# table(main.dt.2.mobs$source)

# Human Mortality Database MOxLAD OECD

#1984 257 1541

#ONS 2008 Riley file UN\_WPP2013

#89 60 2412

# is.data.table(main.dt.2.mobs) TRUE

# unique number of cases by country and year

is.numeric(main.dt.2.mobs$country.code) # FALSE

main.dt.2.mobs[, ct := as.numeric(country.code)\*100000+time]

table(is.na(main.dt.2.mobs$ct))

length(unique(main.dt.2.mobs$ct))

# 2488 unique observations

main.dt.2.mobs[, source.type := NULL]

names(main.dt.2.mobs)

main.dt.2.mobs[, source.type := -1L]

main.dt.2.mobs[source == "Human Mortality Database", source.type := 1L]

main.dt.2.mobs[source == "OECD" , source.type := 2L]

main.dt.2.mobs[source == "UN\_WPP2013" , source.type := 3L]

main.dt.2.mobs[source == "MOxLAD" , source.type := 4L]

main.dt.2.mobs[source == "Riley file" , source.type := 5L]

main.dt.2.mobs[source == "ONS 2008" , source.type := 6L]

summary(main.dt.2.mobs$source.type)

## now order the data.table by source.type

setkeyv(main.dt.2.mobs, c("ct", "source.type"))

main.dt.2.mobs[1:100]

main.dt.2.mobs[, rank := rank(source.type, ties.method = "first"), by = "ct"]

# rank: ranks variables

# ties.method: what to do, when there are ties

dim(main.dt.2.mobs[rank==1, ])

# 2488 observations

# overview of ranks for multiple observations

# HMD

table(main.dt.2.mobs[source.type == 1, rank])

# OECD

table(main.dt.2.mobs[source.type == 2, rank])

# UN

table(main.dt.2.mobs[source.type == 3, rank])

# MOxLAD

table(main.dt.2.mobs[source.type == 4, rank])

# Riley (only provides duplicates)

table(main.dt.2.mobs[source.type == 5, rank])

# ONS 2008

table(main.dt.2.mobs[source.type == 6, rank])

###########

main.dt.2.mobs1 <- main.dt.2.mobs[rank==1, ]

summary(main.dt.2.mobs1$rank)

names(main.dt.2.mobs1)

names(main.dt.2.sobs)

# dropping vars in order to rbind

main.dt.2.mobs1[, mult.obs := NULL]

main.dt.2.mobs1[, ct := NULL]

main.dt.2.mobs1[, rank := NULL]

main.dt.2.mobs1[, source.type := NULL]

main.dt.2.sobs[, mult.obs := NULL]

names(main.dt.2.mobs1)

names(main.dt.2.sobs)

main.df.2.mobs1 <- as.data.frame(main.dt.2.mobs1)

main.df.2.sobs <- as.data.frame(main.dt.2.sobs)

dim(main.df.2.mobs1)

dim(main.df.2.sobs)

main.df.3 <- rbind(main.df.2.sobs, main.df.2.mobs1)

head(main.df.3)

dim(main.df.3)

dim(main.df.2.mobs1)[1] + dim(main.df.2.sobs)[1]

# 2488+13915 = 16403

main.df.3$ct <- as.numeric(main.df.3$country.code)\*100000+main.df.3$time

unique(main.df.3$ct)

head(main.df.3)

tail(main.df.3)

unique(main.df.3$time)

unique(main.df.3$country.code)

table(duplicated(main.df.3$ct))

head(main.df.3)

main.df.3copy <- main.df.3

main.df.3 <- main.df.3[order(main.df.3$area.region.country, main.df.3$time), ]

main.df.3[11000:11300, "area.region.country"]

main.df.3[3977:4025, "area.region.country"]

main.df.3$area.region.country <- as.character(main.df.3$area.region.country)

main.df.3 <- main.df.3[order(main.df.3$area.region.country, main.df.3$time), ]

dim(main.df.3)

write.csv(main.df.3,

file = "./data/derived/clio\_infra/clio\_letot\_20140721.csv",

row.names = FALSE)

chap.dt <- data.table(main.df.3)

setkeyv(chap.dt, c("oecd.chapter", "country.code", "time"))

chap.dt

dim(chap.dt)

# drop empty labels

chap.dt.1 <- chap.dt[!is.na(oecd.chapter), ]

dim(chap.dt.1) # (2000 cases less)

chap.dt.1

# how much unique countries?

unique(chap.dt.1$area.region.country) # 204 countries

# what's the max year we have for these 204 countries?

chap.dt.1[, max.year := max(time), by = area.region.country]

table(chap.dt.1$area.region.country, chap.dt.1$max.year)

# except for Taiwan (max.year = 1960), data of all countries go to

# and including 2009, or beyond.

# ##### 21 JULY 2014 skipping decadal averages for CLIO-INFRA #####

#

# ### function to create decadal averages

# ## OECD INSTRUCTIONS

# # Data are presented in decadal form (1820s=average of 1820-1829, to 2000s,

# # average of 2000-2009), always like this in tables when presenting the data

# # themselves;

# # when presenting growth rates, the usual Maddison benchmark years can be

# # used (1820, 1870, 1913, 1950, 1973, 2008).

#

# floor.fun <- function(x){

# floor(x/10)\*10

# }

# chap.dt.1[, dec := floor.fun(time)]

# table(chap.dt.1$dec)

#

# floor.fun(1820)

# floor.fun(1823)

# floor.fun(1829)

# floor.fun(2011)

# floor.fun(2009)

#

#

# names(chap.dt.1)

# chap.dt.1[, le.dec := mean(letot), by=c("country.code", "dec")]

# chap.dt.1[1:100, list(country.code, dec, le.dec, time, letot)]

# ## checking whether the decadal averages worked for the newly

# ## added data

# # test <- chap.dt.1[area.region.country=="United Kingdom",

# # list(country.code, dec, le.dec, time, letot, source)]

# #test <- chap.dt.1[area.region.country=="Australia",

# # list(country.code, dec, le.dec, time, letot, source)]

# # works indeed

#

# # Are there countries for which we have data just before 1920

# # but not for the 1820-1829 decade?

# chap.dt.1[time>1814 & time <1820,

# list(time, letot, le.dec, area.region.country)]

# # Finland, France, Sweden, UK have data between 1815 and 1820

# chap.dt.1[dec==1820, list(area.region.country, le.dec)]

# # but these four countries have values for dec==1820, so

# # no countries with data on 1815-1819, without values for decade

# ##### 21 JULY 2014 skipping selection of years for CLIO-INFRA #####

# # selecting just data between 1820 and 2009

# dim(chap.dt.1)

# chap.dt.2 <- chap.dt.1 # chap.dt.1[time>=1820 & time <2010, ] # changed in v15

# dim(chap.dt.2)

# names(chap.dt.2)

# # 14303-14087 # loosing 216 cases

## Next merge Auke's labels

## For this purpose I merged my labels with Auke's in

## oecd\_labels\_join01.R

chap.dt.2 <- chap.dt.1 # copying since I did not restrain to years above

# first, dropping label variables in chap.dt.2

names(chap.dt.2)

# removing redundant vars

chap.dt.2[, area.region.country := NULL]

chap.dt.2[, region.sort.label := NULL]

chap.dt.2[, facet.label := NULL]

chap.dt.2[, oecd.chapter := NULL]

chap.dt.2[, max.year := NULL]

# Reading in merged label file

all.labels <- read.xlsx("./data/derived/labels/un\_labels\_2013\_6.xlsx",

sheetName = "labels")

names(chap.dt.2)

names(all.labels)

str(chap.dt.2$country.code)

str(all.labels$country.code)

# different setting in files

chap.dt.2[, country.code := as.numeric(country.code)]

chap.dt.2 <- merge(chap.dt.2, all.labels,

by = "country.code",

all.x = TRUE, all.y = TRUE)

names(chap.dt.2)

head(chap.dt.2)

haskey(chap.dt.2)

setkeyv(chap.dt.2, c("area.region.country", "time"))

write.csv(chap.dt.2,

file = "./data/derived/clio\_infra/clio\_letot\_OECD\_CLIO\_20140721.csv",

row.names = FALSE)

## THE ABOVE IS THE FILE I WILL GIVE TO AUKE, after taking care of issues

### Dealing with issues Auke reported ####

ledat <- read.csv(file = "./data/derived/clio\_infra/clio\_letot\_OECD\_CLIO\_20140721.csv")

dim(ledat) # 14389 15

names(ledat)

## Issue 1:

## Observations for Germany as a whole, West and East at the same point in time

## as well as observations for West and East after 1990

p1 <- ggplot(aes(x = time, y = letot), data = subset(ledat, country.code == 276))

p1 + geom\_line() +

geom\_line(colour = "red", data = subset(ledat, country.code == 278)) +

geom\_line(colour = "blue", data = subset(ledat, country.code == 280))

unique(ledat[country.code == 276, source]) #

ledat[country.code == 276, ]

ledat[country.code == 278, ]

ledat[country.code == 280, ]

## solution: For the OECD data just the "Germany as a whole" data were preserved

## for this release for CLIO it however makes more sense to retain all data

## since researchers might be interested in all three data series. The data

## after 1990 are based on the HMD, which assembles this data by studying the

## seperate parts of Germany.

## Issue 2: which 'Sudan' to use?

unique(ledat[country.code == 728, area.region.country])

ledat <- ledat[!(country.code == 728), ] #remove 'South-Sudan'

ledat[country.code == 729, area.region.country] #'North-Sudan' == Sudan

ledat[country.code == 736, area.region.country] # NA

# so Auke uses 736 for Sudan, which is non-existent in my data

table(ledat$country.code) #60 729's and 1 736

ledat[country.code == 736, ] # no value for life expectancy

ledat[country.code == 729, country.code := 736] # change to Auke's code

table(ledat$country.code) #0 729's and 61 736 #(including one empty row)

# # drop later on everything that is.na(letot)

# For future reference use: https://unstats.un.org/unsd/methods/m49/m49alpha.htm

# which reads:

# As of July of 2011:

# South Sudan (9/07/11- 728

# Sudan (9/07/11- 729 Refers to the new Sudan, now excluding the southern part.

# Sudan (12/11/56-8/07/11) 736 Old code "retired" to be used for

# historical/statistical analysis.

## Issue 3: Two instances of Taiwan

# drop 158 'Other non-specified areas' (== Taiwan)

# but Taiwan is also 9991

ledat[area.region.country == "Taiwan", list(country.code, letot, time, region, source)]

ledat[country.code == 158, list(area.region.country, letot, time, region, source)]

# ok, so in Riley file's there is data on Taiwan, but has regional code 9991

# UN-WPP data has Taiwan ISO code 158

# Let's plot the data

p2 <- ggplot(aes(x = time, y = letot), data = subset(ledat, country.code == 9991))

p2 + geom\_line(colour = "blue") +

geom\_line(colour = "red", data = subset(ledat, country.code == 158))

# the graph shows overlap between 1950-1969. For these overlapping years

# I choose to retain the Riley data, since these data are changing annualy.

# drop overlapping values

ledat <- ledat[!(country.code == 158 & time <= 1969), ]

#p2 <- ggplot(aes(x = time, y = letot), data = subset(ledat, country.code == 9991))

#p2 + geom\_line(colour = "blue") +

# geom\_line(colour = "red", data = subset(ledat.copy, country.code == 158))

# change country.code for riley file to 158

ledat[country.code == 9991, country.code := 158]

p2 <- ggplot(aes(x = time, y = letot), data = subset(ledat, country.code == 158))

p2 + geom\_line(colour = "blue")

### CLEAN data (removing NA's)

dim(ledat) # 14389 15

table(ledat$letot, useNA = "ifany") # 91 missings, let's examine

# ledat.mis <- subset(ledat, is.na(letot)) # appears to be just missing letot

# table(ledat$time, useNA = "ifany") # 91 missings, let's examine

# ledat.nomis <- subset(ledat, !(is.na(letot)))

# table(ledat.nomis$time, useNA = "ifany") # 91 missings, let's examine

# ok, so missings on letot and time, but if you remove missings on letot, there

# are no more missings on time

ledat <- subset(ledat, !(is.na(letot)))

dim(ledat) # 14298 15

# the next few lines actually show no missings

# ledat.copy <- ledat

# ledat.copy <- ledat[!(is.na(letot)), ]

# dim(ledat.copy)

# ledat <- ledat[!is.na(area.region.country)]

# dim(ledat.copy)

# ledat <- ledat[!is.na(country.code)]

# dim(ledat.copy)

## save this file as outfile

write.csv(ledat, "./data/derived/clio\_infra/clio\_letot\_OECD\_20140727.csv")

## slim down and reshape file to match with clio-infra format excel file ####

head(ledat)

is.data.table(ledat)

ledat <- data.table(ledat)

ledat.slim <- ledat[, list(country.code, time, letot, area.region.country)]

ledat.slim.wide <- reshape(ledat.slim, timevar = "time",

idvar = c("country.code", "area.region.country"),

direction = "wide")

sort(names(ledat.slim.wide))

# change codes for some regions to those of clio-infra (others are already ok)

ledat.slim.wide[area.region.country == "Sub-Saharan Africa", country.code := NA]

ledat.slim.wide[area.region.country == "AFRICA", country.code := 2]

ledat.slim.wide[area.region.country == "Micronesia", country.code := 57]

ledat.slim.wide[area.region.country == "Taiwan", country.code := 9002]

ledat.slim.wide[area.region.country == "Middle Africa", country.code := 17]

ledat.slim.wide[area.region.country == "LATIN AMERICA AND THE CARIBBEAN", country.code := 419]

ledat.slim.wide[area.region.country == "Polynesia", country.code := 61]

ledat.slim.wide[area.region.country == "Australia/New Zealand", country.code := 53]

ledat.slim.wide[area.region.country == "Eastern Asia", country.code := 30]

ledat.slim.wide[area.region.country == "ASIA", country.code := 142]

ledat.slim.wide[area.region.country == "Less developed regions", country.code := NA]

ledat.slim.wide[area.region.country == "Northern Africa", country.code := 15]

ledat.slim.wide[area.region.country == "Central Asia", country.code := 143]

ledat.slim.wide[area.region.country == "Melanesia", country.code := 54]

ledat.slim.wide[area.region.country == "Other non-specified areas", country.code := NA]

ledat.slim.wide[area.region.country == "Southern Asia", country.code := 34]

ledat.slim.wide[area.region.country == "Eastern Africa", country.code := 14]

ledat.slim.wide[area.region.country == "EUROPE", country.code := 150]

ledat.slim.wide[area.region.country == "Western Africa", country.code := 11]

ledat.slim.wide[area.region.country == "Caribbean", country.code := 29]

ledat.slim.wide[area.region.country == "Central America", country.code := 13]

ledat.slim.wide[area.region.country == "Northern Europe", country.code := 154]

ledat.slim.wide[area.region.country == "Eastern Europe", country.code := 151]

ledat.slim.wide[area.region.country == "NORTHERN AMERICA", country.code := 21]

ledat.slim.wide[area.region.country == "South America", country.code := 5]

ledat.slim.wide[area.region.country == "Less developed regions, excluding China", country.code := NA]

ledat.slim.wide[area.region.country == "Less developed regions, excluding least developed countries", country.code := NA]

ledat.slim.wide[area.region.country == "Western Asia", country.code := 145]

ledat.slim.wide[area.region.country == "WORLD", country.code := NA]

ledat.slim.wide[area.region.country == "Least developed countries", country.code := NA]

ledat.slim.wide[area.region.country == "More developed regions", country.code := NA]

ledat.slim.wide[area.region.country == "Southern Africa", country.code := 18]

ledat.slim.wide[area.region.country == "OCEANIA", country.code := 9]

ledat.slim.wide[area.region.country == "Southern Europe", country.code := 39]

ledat.slim.wide[area.region.country == "Western Europe", country.code := 155]

ledat.slim.wide[area.region.country == "South-Eastern Asia", country.code := 35]

ledat.slim.wide[area.region.country == "South-Central Asia", country.code := NA]

## now read in the clio-infra formatted file

ciff <- read.xlsx("./data/original/clio\_infra/template/template clio infra.xlsx",

sheetName = "Sheet1", startRow = 3, colIndex = c(1,2))

ciff$id <- 1:nrow(ciff)

# summary(ciff$id)

# head(ciff$id)

# tail(ciff$id)

is.data.table(ledat.slim.wide)

ledat2 <- as.data.frame(ledat.slim.wide)

ledat.merge <- merge(ciff, ledat2,

by.x = "Code",

by.y = "country.code",

# all = TRUE # changed in version 16 to:

all.x = TRUE,

sort = FALSE)

ledat.merge <- ledat.merge[order(ledat.merge$id), ]

# dropping id var

ledat.merge$id <- NULL

# expanding number of years

years.exp <- paste0("letot.", seq(1500, 2013, 1))

years.to.add <- subset(years.exp, !(years.exp %in% names(ledat.merge)))

df <- as.data.frame(matrix(NA,

ncol = length(years.to.add),

nrow = nrow(ledat.merge)))

names(df) <- years.to.add

head(df)

dim(df)

ledat.merge2 <- cbind(ledat.merge, df)

names(ledat.merge2)

ledat.merge2.sort <- ledat.merge2[,c(names(ledat.merge2)[1:3],

sort(names(ledat.merge2)[4:ncol(ledat.merge2)]))]

# dropping third column

ledat.merge2.sort2 <- ledat.merge2.sort[, c(1,2,4:ncol(ledat.merge2))]

## ok, nearly there, now create first three rows and then rbind it

# this doesn't work, resolve later, fixing manually for now!

# ncol(ledat.merge2.sort2)

# row1 <- c("period life expectancy (total population)",

# rep("", ncol(ledat.merge2.sort2)-1))

# row2 <- c("years",

# rep("", ncol(ledat.merge2.sort2)-1))

# row3 <- gsub("..", ",", names(ledat.merge2.sort2), fixed = TRUE)

# row3 <- gsub("letot.", "", row3)

# t1 <- rbind(row1, row2)

# t2 <- rbind(t1, row3)

# t2 <- as.data.frame(t2)

# ledat.merge2.sort3 <- ledat.merge2.sort2

# names(t2) <- names(ledat.merge2.sort3)

write.xlsx(ledat.merge2.sort2, "./data/derived/clio\_infra/clio\_letot\_OECD\_CLIO\_20140726.xlsx",

col.names = FALSE, row.names = FALSE, showNA = FALSE, sheetName =

"le\_total\_un16.R")

##### EOF ####