job submitted

```
library(foreign)
library(readstata13)
# This function takes a dataset and extract the share data of estimated post-tax labour income
get_ratio <- function(data){
r <- data$pxitax / data$pitotal
post_tax_labour <- data$pilabour * r
post_tax_labour <- na.omit(post_tax_labour)
#sprintf("pitotal missing ratio: %f", sum(is.na(data$pitotal)) / length(data$pitotal))
#sprintf("pxitax missing ratio: %f", sum(is.na(data$pxitax)) / length(data$pxitax))
#sprintf("pilabour missing ratio: %f", sum(is.na(data$pilabour)) / length(data$pilabour))
post_tax_labour[post_tax_labour <= 0] <- NA # exclude the people with negative labour income
post_tax_labour <- na.omit(post_tax_labour)
cutoff_1 <- quantile(post_tax_labour, probs = seq(0.9, 1), na.rm = TRUE) # top 10% cutoff
cutoff_2 <- quantile(post_tax_labour, probs = seq(0.99, 1), na.rm = TRUE) # top 1% cutof
s1 <- 0
s2 <- 0
s3 <- 0
for(i in 1:length(post_tax_labour)){
    if (post_tax_labour[i] > cutoff_2){
     s3 <- s3 + post_tax_labour[i]
}
if(post_tax_labour[i] > cutoff_1 & post_tax_labour[i] <= cutoff_2){
     s2 <- s2 + post_tax_labour[i]
}
if (post_tax_labour[i] <= cutoff_1){
     s1 <- s1 + post_tax_labour[i]
}
}
s <- s1 + s2 + s3
sprintf("final ratios: %f, %f, %f", s1/s, s2/s, s3/s)
return(c(s1/s, s2/s, s3/s))
}
# This function plots the evolution of share data (of estimated post-tax labour income) across years
plot_ts <- function(country, start_year, end_year){
    n <- end_year - start_year + 1
    l1 <- rep(NA, n)
    l2 <- rep(NA, n)
    l3 <- rep(NA, n)
    for(i in 1:n){
        year <- start_year + i - 1
        year_str <- substr(paste0(year), 3, 4)
        name <- paste0(country, year_str, "p")
        data <- read.LIS(name)
        l <- get_ratio(data)
        l1[i] <- l[1]
        l2[i] <- l[2]
        l3[i] <- l[3]
    }
    matplot(seq(start_year, end_year), cbind(l1, l2, l3), type = "l", lty = 1, col = c("blue", "green", "red"),
xlab = "year", ylab = "share")
    legend("topright", legend = c("bottom 90%", "top 10%-1%", "top 1%"), col = c("red", "blue", "green"), lty =
1)
}
# sample dataset
```

```
#data <- read.LIS("fr06p")
#print(str(data)) # show variable information
country <- "fr"
start_year <- 1996
end_year <- 2018
plot_ts(country, start_year, end_year)
```

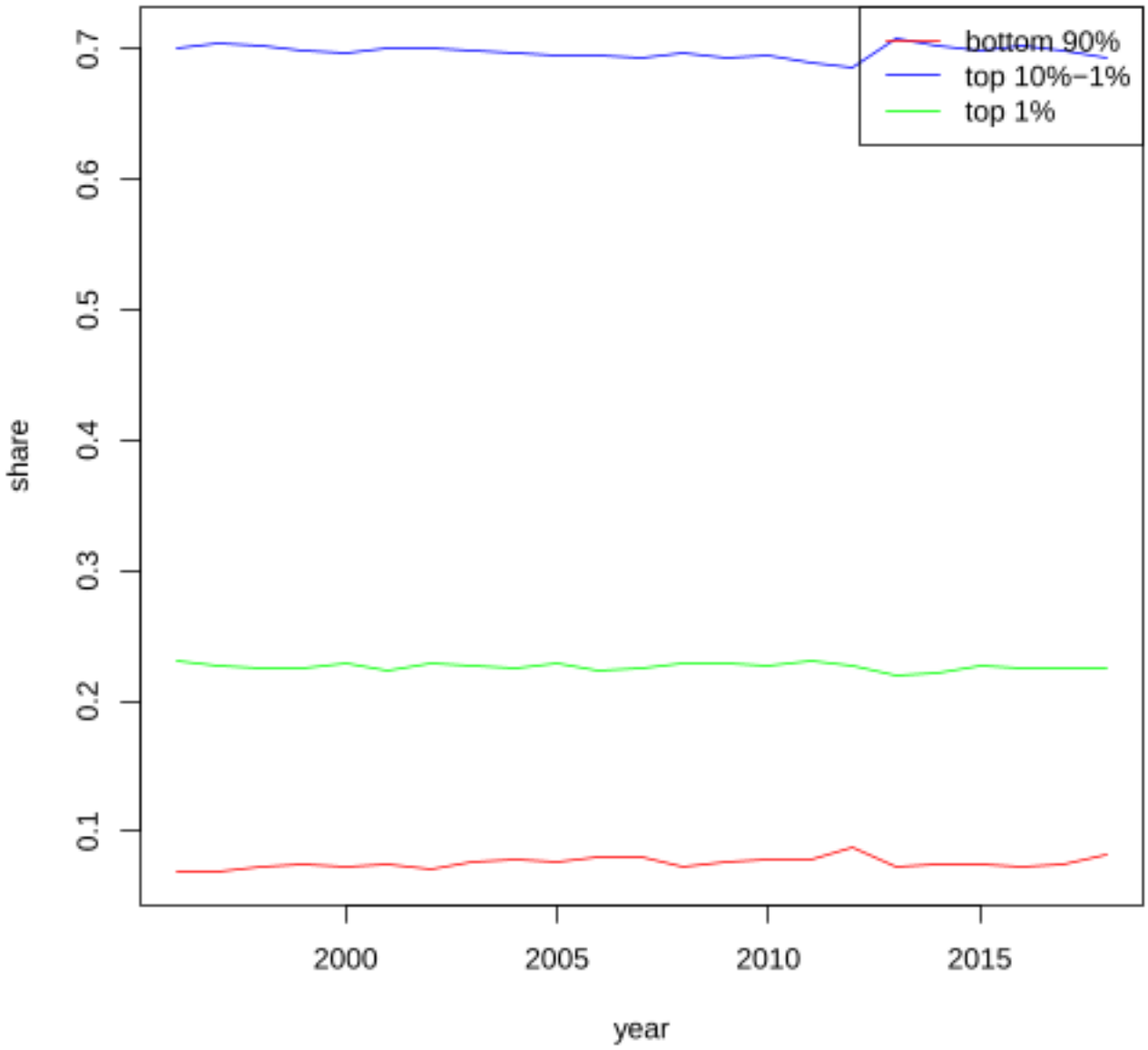
listing

```
################################ NOTICE TO USERS ################################

  Use of the data in the LUXEMBOURG INCOME STUDY DATABASE is governed by
  regulations which do not allow copying or further distribution of the
  survey microdata.

  Anyone violating these regulations will lose all privileges to the
  databases and may be subject to prosecution under the law. In addition,
  any attempt to circumvent the LIS processing system or unauthorized entry
  into the LIS computing system will result in prosecution.

  All papers written using the LUXEMBOURG INCOME STUDY DATABASE must be
  submitted for entry into the Working Papers Series.
  Please consult our web site for more information at WWW.LISDATACENTER.ORG

################################ NOTICE TO USERS ################################

> options(echo=FALSE, encoding="UTF-8")
Loading required package: readstata13
> options(error = expression(q('no')))
> library(foreign)
> library(readstata13)
> # This function takes a dataset and extract the share data of estimated post-tax labour income
> get_ratio <- function(data){
+ r <- data$pxitax / data$pitotal
+ post_tax_labour <- data$pilabour * r
+ post_tax_labour <- na.omit(post_tax_labour)
+ #sprintf("pitotal missing ratio: %f", sum(is.na(data$pitotal)) / length(data$pitotal))
+ #sprintf("pxitax missing ratio: %f", sum(is.na(data$pxitax)) / length(data$pxitax))
+ #sprintf("pilabour missing ratio: %f", sum(is.na(data$pilabour)) / length(data$pilabour))
+ post_tax_labour[post_tax_labour <= 0] <- NA # exclude the people with negative labour income
+ post_tax_labour <- na.omit(post_tax_labour)
+ cutoff_1 <- quantile(post_tax_labour, probs = seq(0.9, 1), na.rm = TRUE) # top 10% cutoff
+ cutoff_2 <- quantile(post_tax_labour, probs = seq(0.99, 1), na.rm = TRUE) # top 1% cutof
+ s1 <- 0
+ s2 <- 0
+ s3 <- 0
+ for(i in 1:length(post_tax_labour)){
+     if (post_tax_labour[i] > cutoff_2){
+      s3 <- s3 + post_tax_labour[i]
+ }
+ if(post_tax_labour[i] > cutoff_1 & post_tax_labour[i] <= cutoff_2){
+      s2 <- s2 + post_tax_labour[i]
+ }
+ if (post_tax_labour[i] <= cutoff_1){
+      s1 <- s1 + post_tax_labour[i]
```

```
+ }
+ }
+ s <- s1 + s2 + s3
+ sprintf("final ratios: %f, %f, %f", s1/s, s2/s, s3/s)
+ return(c(s1/s, s2/s, s3/s))
+ }
> # This function plots the evolution of share data (of estimated post-tax labour income) across years
> plot_ts <- function(country, start_year, end_year){
+     n <- end_year - start_year + 1
+     l1 <- rep(NA, n)
+     l2 <- rep(NA, n)
+     l3 <- rep(NA, n)
+     for(i in 1:n){
+         year <- start_year + i - 1
+         year_str <- substr(paste0(year), 3, 4)
+         name <- paste0(country, year_str, "p")
+         data <- read.LIS(name)
+         l <- get_ratio(data)
+         l1[i] <- l[1]
+         l2[i] <- l[2]
+         l3[i] <- l[3]
+     }
+     matplot(seq(start_year, end_year), cbind(l1, l2, l3), type = "l", lty = 1, col = c("blue", "green",
"red"), xlab = "year", ylab = "share")
+     legend("topright", legend = c("bottom 90%", "top 10%-1%", "top 1%"), col = c("red", "blue", "green"), lty
= 1)
+ }
> # sample dataset
> #data <- read.LIS("fr06p")
> #print(str(data)) # show variable information
> country <- "fr"
> start_year <- 1996
> end_year <- 2018
> plot_ts(country, start_year, end_year)
[1] "Loading dataset fr96p..."
[1] "Loading dataset fr97p..."
[1] "Loading dataset fr98p..."
[1] "Loading dataset fr99p..."
[1] "Loading dataset fr00p..."
[1] "Loading dataset fr01p..."
[1] "Loading dataset fr02p..."
[1] "Loading dataset fr03p..."
[1] "Loading dataset fr04p..."
[1] "Loading dataset fr05p..."
[1] "Loading dataset fr06p..."
[1] "Loading dataset fr07p..."
[1] "Loading dataset fr08p..."
[1] "Loading dataset fr09p..."
[1] "Loading dataset fr10p..."
[1] "Loading dataset fr11p..."
[1] "Loading dataset fr12p..."
[1] "Loading dataset fr13p..."
[1] "Loading dataset fr14p..."
[1] "Loading dataset fr15p..."
[1] "Loading dataset fr16p..."
[1] "Loading dataset fr17p..."
[1] "Loading dataset fr18p..."
```

```
>
> proc.time()
   user  system elapsed
 29.581   0.975  32.371
```

graphs


bottom 90%
top 10%−1%
top 1%
share
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2000
2005
2010
2015
year