SP_DemTech2_Problem Set 2

Sara Peters

2023-09-27

###Lifetable Estimates Using Sullivan's Method

#In this problem set, you will be calculating gender differences in the expected duration of years lived in poverty for the United States in 2004. You will calculate variance estimates for these expectancies to determine if the estimates of differences between men and women are statistically meaningful, and at what ages.

1. Sullivan's method requires high quality lifetable data to construct total person years in each age group. Set up an account on the Human Mortality Database: https://www.mortality.org/ (https://www.mortality.org/) and get the nLx values for U.S. males and U.S. females in 2004 from the 5x1 abridged tables, e.g. for women: https://www.mortality.org/hmd/USA/STATS/fltper_5x1.txt (https://www.mortality.org/hmd/USA/STATS/fltper_5x1.txt)

#set up

```
#Set working directory
setwd("C:/Users/saraa/OneDrive - UW-Madison/SOC 756- Demography Techniques II/Problem Sets/Probl
em Set 2")

#load libraries
# install.packages("dplyr")
# install.packages("tidyverse")
# install.packages("ggplot2")
# install.packages("HMDHFDplus")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:kableExtra':
##
##     group_rows
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ forcats   1.0.0     ✔ readr     2.1.4
## ✔ ggplot2   3.4.3     ✔ stringr   1.5.0
## ✔ lubridate 1.9.2     ✔ tibble    3.2.1
## ✔ purrr     1.0.1     ✔ tidyr     1.3.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::group_rows() masks kableExtra::group_rows()
## ✖ dplyr::lag()        masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(ggplot2)
library (HMDHFDplus)

#HMD login info:
#username: speters27@wisc.edu
#password: @DemTech2

# Set the 'scipen' option to a large value to prevent scientific notation
options(scipen = 999)

# Set the 'digits' option to control the number of decimal places
options(digits = 6)  # Change the number to the desired decimal places
```

#Get Data from Human Mortality Database

Install the R package HMDHFDplus and get the lifetable values directly: https://cran.r-project.org/web/packages/HMDHFDplus/HMDHFDplus.pdf (https://cran.r-project.org/web/packages/HMDHFDplus/HMDHFDplus.pdf) The commands ask you to supply your user name and password, you'll still need to sign up at HMD first.

```
getHMDcountries()
```

```
## # A tibble: 49 × 3
##    Country   link                         CNTRY
##    <chr>     <chr>                        <chr>
##  1 Australia /Country/Country?cntr=AUS AUS
##  2 Austria   /Country/Country?cntr=AUT AUT
##  3 Belarus   /Country/Country?cntr=BLR BLR
##  4 Belgium   /Country/Country?cntr=BEL BEL
##  5 Bulgaria  /Country/Country?cntr=BGR BGR
##  6 Canada    /Country/Country?cntr=CAN CAN
##  7 Chile     /Country/Country?cntr=CHL CHL
##  8 Croatia   /Country/Country?cntr=HRV HRV
##  9 Czechia   /Country/Country?cntr=CZE CZE
## 10 Denmark   /Country/Country?cntr=DNK DNK
## # ℹ 39 more rows
```

```
#get lifetable data for US females in 2004
USfemales<-readHMDweb(CNTRY = "USA", item = "fltper_5x1", username = "speters27@wisc.edu" , pass
word = "@DemTech2" )

#get lifetable data for US males in 2004
USmales<-readHMDweb(CNTRY = "USA", item = "mltper_5x1", username = "speters27@wisc.edu" , passwo
rd = "@DemTech2" )

#Filter lifetables for only values where year = 2004
#create vector to identify male vs female life table data
USfemales04 <- USfemales %>%
  filter(Year == 2004) %>%
  mutate(sex = "female")

USmales04 <- USmales %>%
  filter(Year == 2004) %>%
  mutate(sex = "male")


#subset life tables to only use Lx, age
USF_LT_Join <- USfemales04 %>%
  select(Age,lx, Lx, ex)

USM_LT_Join <- USmales04 %>%
  select(Age,lx, Lx, ex)
```

#Upload GSS data 2. I have used General Social Survey data to estimate the poverty prevalence by age separately for men and women. I have defined the poverty prevalence as the proportion living at or below the poverty line at the time of the survey. You will find these values, as well as the size of the sample used to compute them in a table on the class webpage.

```
GSS_poverty<- read.csv("ps2data_2023.csv")

#dataset is in long format, convert to two data sets (males and females) to make it easier to wr
angle.

#rename columns
GSS_poverty <- GSS_poverty %>%
  rename(pix = proportion_poverty_1.n_x) %>%
  rename(Sx = number_sampled_N)

Age <- c(0, 1, rep(seq(5, 110, by = 5)))

#separate females from males and add age column for joining with LT data
Poverty_females<- GSS_poverty %>%
  filter (sex == "female") %>%
    mutate (Age = Age)

Poverty_males<- GSS_poverty %>%
  filter (sex == "male") %>%
  mutate (Age = Age)

#join life table data with GSS poverty data

Poverty_LTF <- left_join (USF_LT_Join, Poverty_females, by =c("Age"))
Poverty_LTM <- left_join (USM_LT_Join, Poverty_males, by =c("Age"))

#Fix Sx column to be numeric
Poverty_LTF = Poverty_LTF %>%
  mutate(Sx = str_replace_all(Sx, ",", "")) %>%
  mutate(Sx = as.numeric(Sx))

str(Poverty_LTF$Sx)
```

```
##  num [1:24] 470 1041 1836 1789 1610 ...
```

```
Poverty_LTM = Poverty_LTM %>%
  mutate(Sx = str_replace_all(Sx, ",", "")) %>%
  mutate(Sx = as.numeric(Sx))

str(Poverty_LTM$Sx)
```

```
##  num [1:24] 450 976 1734 1683 1579 ...
```

#Create Poverty Free Life Table for US Females and Males in 2004

```r
#Create Life Table for Years lived without poverty for US females in 2004

#Person years lived with poverty at age x
Poverty_LTF = Poverty_LTF %>%
  mutate(PY_P = (pix)*Lx)


# computing Total years lived with poverty from age x
# Initialize an empty vector to store nqx values

Poverty_LTF$T_pov <- numeric(length(Poverty_LTF$Lx))

#Calculate Tx_poverty
#Sum 1-pix
for (i in 1:nrow(Poverty_LTF)) {

  Poverty_LTF$T_pov[i] <- sum(Poverty_LTF$PY_P[i:24])

}

#Compute poverty free life expectancy (PLE_F)
#Txi/lx

Poverty_LTF = Poverty_LTF %>%
  mutate(PLE_F = T_pov/lx)

#Compute Proportion of life spent in poverty (%PFLE)

Poverty_LTF = Poverty_LTF %>%
  mutate(PLE_prop = PLE_F/ex)


#Create Life Table for Years lived without poverty for US males in 2004

#Person years lived without poverty at age x
Poverty_LTM = Poverty_LTM %>%
  mutate(PY_P = (pix*Lx))


# computing Total years lived without disability from age x
# Initialize an empty vector to store Txi values

Poverty_LTM$T_pov <- numeric(length(Poverty_LTM$Lx))

#Calculate Tx_poverty
#Sum 1-pix
for (i in 1:nrow(Poverty_LTM)) {

  Poverty_LTM$T_pov[i] <- sum(Poverty_LTM$PY_PF[i:24])

}
```

```
#Compute time spent in poverty poverty life expectancy (PFLE_M)
#Txi/lx

Poverty_LTM = Poverty_LTM %>%
  mutate(PLE_M = T_pov/lx)

#Compute Proportion of life spent in poverty

Poverty_LTM = Poverty_LTM %>%
  mutate(PLE_prop = PLE_M/ex)
```

#estimate standard errors Because the GSS uses simple random sampling, we should be able to estimate standard errors using the simplified formula made available in Molla, Wagener, and Madans (2001), as opposed to correcting for the sampling scheme with a more complicated approach.**.

You may use the approximation given in equation 13 of Molla, Wagener, and Madans (2001). A tip: pay careful attention to the notation on the components of each of these equations. For example, equation 11 uses the nLx values, not the nLx prime values.

**We must abstract away from reality in a few ways here: (1) the GSS does not include age values above 85. I have filled in the rest of the lifetable assuming that the poverty rates for the 85+ group could be applied to each of the sub-groups in that larger age intervals. I have also greatly increased the number of total respondents above age 85 by giving each subgroup above age 85 the same number of respondents. (2) The GSS does not interview children. I have pulled child poverty data from the National Center for Educational Statistics, assumed that the age-specific rates do not vary between boys and girls and have fabricated the sample sizes of children used to create these rates. Finally: note that I use the GSS here for illustrative purposes. How to measure poverty is nearly its own field and you are at the institution with people leading this field. Use the IRP as a resource if poverty measurement is important to your work: https://www.irp.wisc.edu/resources/how-is-poverty-measured/ (https://www.irp.wisc.edu/resources/how-is-poverty-measured/)

#Calculate standard errors

```r
#For US Females 2004
#S^2(pix) = (pix*(1-pix)/Sx)

Poverty_LTF <- Poverty_LTF %>%
  mutate(S2 = (pix * (1-pix) / Sx)) %>% #variance of prevalence rates in age interval x
  mutate(VAR = (PY_P^2 * S2 )) #variance of life in poverty in age interval x [Li^2* S2]

# Initialize an empty vector to store var_sum values

Poverty_LTF$var_sum <- numeric(length(Poverty_LTF$Lx))

##Calculate sum of variance of life in poverty
#1/lx^2(Sum(var)

for (i in 1:nrow(Poverty_LTF)) {

  Poverty_LTF$var_sum[i] <- sum(Poverty_LTF$VAR[i:24])

}

#calculate overall variance  of (var_ex)
Poverty_LTF = Poverty_LTF %>%
  mutate(var_ex_f = (1/lx^2* var_sum))

#calculate standard error of life in poverty at each age interval

Poverty_LTF = Poverty_LTF %>%
  mutate(se_p_f = sqrt(var_ex_f))


#For US Males 2004
#S^2(pix) = (pix*(1-pix)/Sx)

Poverty_LTM <- Poverty_LTM %>%
  mutate(S2 = (pix * (1-pix) / Sx)) %>% #variance of prevalence rates in age interval
  mutate(VAR = (PY_P^2 * S2 )) #variance of life in poverty in age interval

# Initialize an empty vector to store var_sum values

Poverty_LTM$var_sum <- numeric(length(Poverty_LTM$Lx))

#Calculate sum of variance of life in poverty

#1/lx^2(Sum(var)

for (i in 1:nrow(Poverty_LTM)) {

  Poverty_LTM$var_sum[i] <- sum(Poverty_LTM$VAR[i:24])

}

Poverty_LTM = Poverty_LTM %>%
```

```
  mutate(var_ex_m = (1/lx^2* var_sum))

#calculate standard error of life in poverty at each age interval

Poverty_LTM = Poverty_LTM %>%
  mutate(se_p_m = sqrt(var_ex_m))

print(Poverty_LTF)
```

```
##    Age     lx     Lx    ex       x    pix   Sx    sex        PY_P       T_pov
## 1    0 100000  99463 80.06     0-1 0.2100  470 female  20887.2300 1628760.8143
## 2    1  99379 397246 79.56     1-5 0.2100 1041 female  83421.6600 1607873.5843
## 3    5  99269 496172 75.65     6-9 0.1830 1836 female  90799.4760 1524451.9243
## 4   10  99204 495854 70.69   10-14 0.1620 1789 female  80328.3480 1433652.4483
## 5   15  99128 495209 65.75   15-19 0.1620 1610 female  80223.8580 1353324.1003
## 6   20  98934 494089 60.87   20-24 0.2714 1562 female 134095.7546 1273100.2423
## 7   25  98700 492850 56.01   25-29 0.1785 1972 female  87973.7250 1139004.4877
## 8   30  98429 491318 51.16   30-34 0.1644 1874 female  80772.6792 1051030.7627
## 9   35  98074 489134 46.33   35-39 0.1567 1659 female  76647.2978  970258.0835
## 10  40  97541 485721 41.57   40-44 0.1363 1357 female  66203.7723  893610.7857
## 11  45  96690 480458 36.91   45-49 0.1253 1197 female  60201.3874  827407.0134
## 12  50  95426 472948 32.37   50-54 0.1301 1099 female  61530.5348  767205.6260
## 13  55  93664 462237 27.93   55-59 0.1522 1104 female  70352.4714  705675.0912
## 14  60  91052 445891 23.65   60-64 0.1976 1063 female  88108.0616  635322.6198
## 15  65  87054 421593 19.61   65-69 0.2573 1057 female 108475.8789  547214.5582
## 16  70  81269 386545 15.82   70-74 0.3055  851 female 118089.4975  438738.6793
## 17  75  72884 336703 12.34   75-79 0.3292  647 female 110842.6276  320649.1818
## 18  80  61196 266856  9.19   80-84 0.3667  360 female  97856.0952  209806.5542
## 19  85  44822 177093  6.60   85-89 0.3785  214 female  67029.7005  111950.4590
## 20  90  25909  87082  4.58   90-94 0.3785  214 female  32960.5370   44920.7585
## 21  95   9926  26810  3.18   95-99 0.3785  214 female  10147.5850   11960.2215
## 22 100   2109   4432  2.27 100-104 0.3785  214 female   1677.5120    1812.6365
## 23 105    208    344  1.71 105-109 0.3785  214 female    130.2040     135.1245
## 24 110      9     13  1.43    110+ 0.3785  214 female      4.9205       4.9205
##        PLE_F PLE_prop           S2              VAR          var_sum    var_ex_f
## 1  16.287608 0.203443 0.0003529787  153996.2786306 31827340.7527419 0.003182734
## 2  16.179209 0.203359 0.0001593660 1109055.5811260 31673344.4741113 0.003207042
## 3  15.356777 0.202998 0.0000814330  671378.0739943 30564288.8929853 0.003101609
## 4  14.451559 0.204436 0.0000758837  489650.6819203 29892910.8189910 0.003037455
## 5  13.652289 0.207639 0.0000843205  542675.5364740 29403260.1370707 0.002992284
## 6  12.868177 0.211404 0.0001265954 2276397.1738735 28860584.6005967 0.002948587
## 7  11.540066 0.206036 0.0000743599  575499.3537647 26584187.4267232 0.002728909
## 8  10.678060 0.208719 0.0000733045  478255.1275730 26008688.0729584 0.002684555
## 9   9.893122 0.213536 0.0000796535  467948.8750766 25530432.9453854 0.002654302
## 10  9.161386 0.220385 0.0000867519  380228.2672189 25062484.0703088 0.002634206
## 11  8.557317 0.231843 0.0000915622  331840.2388812 24682255.8030899 0.002640108
## 12  8.039797 0.248372 0.0001029791  389879.4229767 24350415.5642088 0.002674071
## 13  7.534112 0.269750 0.0001168797  578492.4667687 23960536.1412321 0.002731185
## 14  6.977580 0.295035 0.0001491573 1157912.8918368 23382043.6744633 0.002820353
## 15  6.285921 0.320547 0.0001807916 2127377.5799849 22224130.7826265 0.002932563
## 16  5.398598 0.341251 0.0002493182 3476773.9397381 20096753.2026417 0.003042819
## 17  4.399445 0.356519 0.0003413097 4193360.7393582 16619979.2629035 0.003128716
## 18  3.428436 0.373062 0.0006450864 6177228.4222702 12426618.5235453 0.003318233
## 19  2.497668 0.378434 0.0010992418 4938872.3468048  6249390.1012752 0.003110679
## 20  1.733790 0.378557 0.0010992418 1194213.0174241  1310517.7544703 0.001952278
## 21  1.204939 0.378912 0.0010992418  113192.7572816   116304.7370462 0.001180453
## 22  0.859477 0.378624 0.0010992418    3093.3176142     3111.9797646 0.000699654
## 23  0.649637 0.379905 0.0010992418      18.6355363       18.6621504 0.000431355
## 24  0.546722 0.382323 0.0010992418       0.0266141        0.0266141 0.000328569
##       se_p_f
## 1  0.0564157
```

```
## 2  0.0566308
## 3  0.0556921
## 4  0.0551131
## 5  0.0547018
## 6  0.0543009
## 7  0.0522390
## 8  0.0518127
## 9  0.0515199
## 10 0.0513245
## 11 0.0513820
## 12 0.0517114
## 13 0.0522607
## 14 0.0531070
## 15 0.0541531
## 16 0.0551617
## 17 0.0559349
## 18 0.0576041
## 19 0.0557735
## 20 0.0441846
## 21 0.0343577
## 22 0.0264510
## 23 0.0207691
## 24 0.0181265
```

```
print(Poverty_LTM)
```

```
##    Age     lx     Lx    ex       x    pix   Sx  sex      PY_P T_pov PLE_M
## 1    0 100000  99339 74.96     0-1 0.2100  450 male 20861.190     0     0
## 2    1  99237 396637 74.54     1-5 0.2100  976 male 83293.770     0     0
## 3    5  99107 495323 70.63     6-9 0.1830 1734 male 90644.109     0     0
## 4   10  99025 494901 65.69   10-14 0.1620 1683 male 80173.962     0     0
## 5   15  98919 493708 60.76   15-19 0.1620 1579 male 79980.696     0     0
## 6   20  98479 490686 56.02   20-24 0.1889 1339 male 92690.585     0     0
## 7   25  97791 487228 51.39   25-29 0.1067 1574 male 51987.228     0     0
## 8   30  97106 483821 46.74   30-34 0.0801 1424 male 38754.062     0     0
## 9   35  96399 479768 42.06   35-39 0.0647 1344 male 31040.990     0     0
## 10  40  95461 474048 37.45   40-44 0.0754 1180 male 35743.219     0     0
## 11  45  94074 465477 32.96   45-49 0.0822  997 male 38262.209     0     0
## 12  50  91995 452883 28.65   50-54 0.0936  919 male 42389.849     0     0
## 13  55  89019 435619 24.52   55-59 0.1069  889 male 46567.671     0     0
## 14  60  85015 411328 20.55   60-64 0.1121  794 male 46109.869     0     0
## 15  65  79213 377226 16.86   65-69 0.1534  756 male 57866.468     0     0
## 16  70  71289 330982 13.44   70-74 0.1990  618 male 65865.418     0     0
## 17  75  60636 269880 10.34   75-79 0.2111  360 male 56971.668     0     0
## 18  80  46856 193019  7.63   80-84 0.2642  193 male 50995.620     0     0
## 19  85  30104 109857  5.46   85-89 0.3130  115 male 34385.241     0     0
## 20  90  14295  43261  3.81   90-94 0.3130  115 male 13540.693     0     0
## 21  95   4146   9982  2.71   95-99 0.3130  115 male  3124.366     0     0
## 22 100    633   1198  2.00 100-104 0.3130  115 male   374.974     0     0
## 23 105     44     68  1.57 105-109 0.3130  115 male    21.284     0     0
## 24 110      1      2  1.35    110+ 0.3130  115 male     0.626     0     0
##    PLE_prop           S2               VAR           var_sum   var_ex_m
## 1         0 0.0003686667  160439.769509002 13623271.888117047 0.001362327
## 2         0 0.0001699795 1179292.691437357 13462832.118608046 0.001367065
## 3         0 0.0000862232  708440.240548928 12283539.427170688 0.001250590
## 4         0 0.0000806631  518491.461673877 11575099.186621761 0.001180416
## 5         0 0.0000859759  549980.461796628 11056607.724947885 0.001129958
## 6         0 0.0001144263  983098.497455674 10506627.263151256 0.001083368
## 7         0 0.0000605560  163662.937171038  9523528.765695583 0.000995864
## 8         0 0.0000517444   77713.703728744  9359865.828524545 0.000992607
## 9         0 0.0000450252   43383.747412302  9282152.124795800 0.000998858
## 10        0 0.0000590804   75479.768010390  9238768.377383498 0.001013823
## 11        0 0.0000756702  110780.877508701  9163288.609373108 0.001035409
## 12        0 0.0000923167  165883.797607255  9052507.731864408 0.001069647
## 13        0 0.0001073930  232886.906181302  8886623.934257153 0.001121428
## 14        0 0.0001253572  266524.378395176  8653737.028075852 0.001197326
## 15        0 0.0001717837  575222.392984216  8387212.649680674 0.001336672
## 16        0 0.0002579272 1118953.456155467  7811990.256696459 0.001537151
## 17        0 0.0004626022 1501500.766309289  6693036.800540992 0.001820380
## 18        0 0.0010072454 2619395.257578841  5191536.034231703 0.002364642
## 19        0 0.0018698348 2210789.429511260  2572140.776652862 0.002838222
## 20        0 0.0018698348  342834.893471548   361351.347141602 0.001768322
## 21        0 0.0018698348   18252.696830178    18516.453670054 0.001077207
## 22        0 0.0018698348     262.909055790      263.756839875 0.000658258
## 23        0 0.0018698348       0.847051342        0.847784085 0.000437905
## 24        0 0.0018698348       0.000732743        0.000732743 0.000732743
##       se_p_m
## 1  0.0369097
```

```
## 2  0.0369738
## 3  0.0353637
## 4  0.0343572
## 5  0.0336149
## 6  0.0329146
## 7  0.0315573
## 8  0.0315057
## 9  0.0316047
## 10 0.0318406
## 11 0.0321778
## 12 0.0327055
## 13 0.0334877
## 14 0.0346024
## 15 0.0365605
## 16 0.0392065
## 17 0.0426659
## 18 0.0486276
## 19 0.0532750
## 20 0.0420514
## 21 0.0328208
## 22 0.0256565
## 23 0.0209262
## 24 0.0270692
```

#Perform statistical test for disparity in life lived in poverty at specific ages for females and males

```
#Select years in poverty and standard error for calculating (females)
Poverty_LTF_z = Poverty_LTF %>%
  select(Age, PLE_F, se_p_f)

#Select years in poverty and standard error for calculating (males)
Poverty_LTM_z = Poverty_LTM %>%
  select(Age, PLE_M, se_p_m)

Poverty_z <- left_join(Poverty_LTF_z, Poverty_LTM_z, by= c("Age"))

Poverty_z = Poverty_z %>%
  mutate(diff = PLE_F - PLE_M) %>%
  mutate(SE_diff = se_p_f - se_p_m) %>%
  mutate(z_stat = sqrt(diff/SE_diff)) %>%
  mutate(p_value = 2 * pnorm(abs(z_stat), lower.tail = FALSE))
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `z_stat = sqrt(diff/SE_diff)`.
## Caused by warning in `sqrt()`:
## ! NaNs produced
```

#Interpret results ***(a)Calculate the expected number of years lived in poverty above age x separately for men and women ages 0-100 (using the given intervals).***

```
print(Poverty_z)
```

```
##    Age     PLE_F    se_p_f PLE_M    se_p_m      diff      SE_diff  z_stat
## 1    0 16.287608 0.0564157     0 0.0369097 16.287608  0.019506009 28.8964
## 2    1 16.179209 0.0566308     0 0.0369738 16.179209  0.019656911 28.6894
## 3    5 15.356777 0.0556921     0 0.0353637 15.356777  0.020328410 27.4852
## 4   10 14.451559 0.0551131     0 0.0343572 14.451559  0.020755930 26.3868
## 5   15 13.652289 0.0547018     0 0.0336149 13.652289  0.021086918 25.4446
## 6   20 12.868177 0.0543009     0 0.0329146 12.868177  0.021386337 24.5296
## 7   25 11.540066 0.0522390     0 0.0315573 11.540066  0.020681646 23.6217
## 8   30 10.678060 0.0518127     0 0.0315057 10.678060  0.020307019 22.9310
## 9   35  9.893122 0.0515199     0 0.0316047  9.893122  0.019915212 22.2882
## 10  40  9.161386 0.0513245     0 0.0318406  9.161386  0.019483928 21.6841
## 11  45  8.557317 0.0513820     0 0.0321778  8.557317  0.019204204 21.1091
## 12  50  8.039797 0.0517114     0 0.0327055  8.039797  0.019005963 20.5673
## 13  55  7.534112 0.0522607     0 0.0334877  7.534112  0.018773013 20.0331
## 14  60  6.977580 0.0531070     0 0.0346024  6.977580  0.018504592 19.4184
## 15  65  6.285921 0.0541531     0 0.0365605  6.285921  0.017592622 18.9025
## 16  70  5.398598 0.0551617     0 0.0392065  5.398598  0.015955227 18.3945
## 17  75  4.399445 0.0559349     0 0.0426659  4.399445  0.013269007 18.2087
## 18  80  3.428436 0.0576041     0 0.0486276  3.428436  0.008976519 19.5431
## 19  85  2.497668 0.0557735     0 0.0532750  2.497668  0.002498496 31.6175
## 20  90  1.733790 0.0441846     0 0.0420514  1.733790  0.002133166 28.5093
## 21  95  1.204939 0.0343577     0 0.0328208  1.204939  0.001536894 28.0002
## 22 100  0.859477 0.0264510     0 0.0256565  0.859477  0.000794438 32.8918
## 23 105  0.649637 0.0207691     0 0.0209262  0.649637 -0.000157088     NaN
## 24 110  0.546722 0.0181265     0 0.0270692  0.546722 -0.008942758     NaN
##
p_value
## 1  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000013
2318070256431644885625198426737370027694851160049438476562
## 2  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000005177
9483478505365580575603434709819339332170784473419189453125
## 3  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000002641202347252907530
8282836553885886132775340229272842407226562500000000000000
## 4  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000019436971816143536641655065189127
0717896986752748489379882812500000000000000000000000000000
## 5  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000809443609647046490897237536188413287163712
0842933654785156250000000000000000000000000000000000000000
## 6  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000007142545982700462649152978178435091649589594453573226
9287109375000000000000000000000000000000000000000000000000
## 7  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000023055150885120297842185987935437196938437409698963165283203125
0000000000000000000000000000000000000000000000000000000000
## 8  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000022800635907288341703777478919334953388897702097892761230468750000000000
0000000000000000000000000000000000000000000000000000000000
## 9  0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
```

```
000000000000000000004813712472731255552667317676451830266159959137439727783203125000000000000000
0000000000000000000000000000000000000000000000000000000000
## 10 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000289577046318605664937395616220783267635852098464965820312500000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 11 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000655462079119885031351991355741404277068795636296272277832031250000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 12 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000005386747937256408859385548026921242126263678073883056640625000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 13 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
283216248269361619352424419560065871337428689002990722656250000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 14 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000053981
306566567826762947923668889416148886084556579589843750000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 15 0.0000000000000000000000000000000000000000000000000000000000000000000000000000001087811168
036899347605350740764151851180940866470336914062500000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 16 0.0000000000000000000000000000000000000000000000000000000000000000000000000014528004049346
330633918356056710763368755578994750976562500000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 17 0.0000000000000000000000000000000000000000000000000000000000000000000000000440004945186138
424813998559770311658212449401617050170898437500000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 18 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000004720
124894053064409406295087734406479285098612308502197265625000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000
## 19 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000212007096141951559796490
## 20 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000899478
4790625471974055693547001055776490829885005950927734375000
## 21 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000001617406475681
7621848431626441211506062245462089776992797851562500000000
## 22 0.0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000288353
## 23
NaN
## 24
NaN
```

(b) Assess whether, at birth, the total expected number of years lived in poverty differs for men and women and whether this difference is statistically significant.

At birth the total expected number of years lived in poverty for females in the US in 2004 was 16, while for US males it was about 11. These numbers are different and a z-test indicates that they are statistically significant at alpha = 0.05

(c) Assess these differences at each age interval. According to my results in 2004 in the US females at almost every age (0-100) were expected to live more years while experiencing poverty than males and this difference is statistically significant. The exceptions are between exact ages 105 and 110- I am not sure how to interpret those results or if they are right.