

EPIC Model Interface Tutorial

Amin Adibi

2016-10-17

Contents

	1
Introduction	1
System Information	1
EPIC Files	1
Quick Setup	2
Model Validation	2
sanity_check()	3
sanity_COPD()	3
validate_population (remove_COPD=0,incidence_k=1)	4
validate_smoking(remove_COPD=1,intercept_k=NULL)	11
validate_comorbidity()	18
validate_lung_function()	22
Troubleshooting	24

Introduction

This guide aims to provide a basic introduction to the interface of the EPIC model. The purpose of this document is to familiarize the user with nuts and bolts of interacting with the EPIC model, including the steps involved in setting up the EPIC model using desired parameters, running simulations and producing the results in graphical and tabular formats. A discussion of the innerworkings of the model and the under-the-hood algorithms is beyond the scope of this document. Please note that this document is automatically generated using a markdown tool; it is a work in progress.

System Information

This document is generated using R version 3.3.2 (2016-10-31)

EPIC Files

The following files are necessary in order to run the model:

name and location	functionality
./code.R	The main model code that is executed in R
./input.R	Includes model input parameters
./C/model.WIP.cpp	The computational engine of the model in C
./validation.R	Functions and tools to check the validity of the model
./calibration.R	Calibration of the model
./report.R	Produces graphical results from the model

Quick Setup

1. Open code.R in RStudio. In this file, you can set the default settings for running the model. These include the following. A more comprehensive discussion of model settings and their values can be found in subsequent sections.

```
[1] "record_mode"           "events_to_record"
[3] "agent_creation_mode"   "update_continuous_outcomes_mode"
[5] "n_base_agents"         "runif_buffer_size"
[7] "rnorm_buffer_size"     "rexp_buffer_size"
[9] "agent_stack_size"      "event_stack_size"
```

2. Set model settings in Core.R to desired values, if required.
3. Source Core.R
4. Initialize a session. This is done using the following command:

```
init_session()
```

Initializing the session

```
[1] 0
```

If you cannot see the output as above, the model is not being compiled correctly. Make sure your R and R Studio are updated and all required packages are installed.

4. Run the simulation, using the following command.

```
run()
```

```
[1] 0
```

If you cannot see the output as above, the simulation is not being carried out correctly. For running the simulation for a smaller number of patients (n), you can type in Run(n), i.e. Run (1000).

5. Get the output.
6. Terminate the session to release allocated resources.

```
terminate_session()
```

Terminating the session

```
[1] 0
```

Model Validation

Functions for model validation can be accessed once core.R and validation.R are sourced. The following functions are available:

1. sanity_check()
2. validate_population(remove_COPD=0,incidence_k=1)
3. validate_smoking()
4. sanity_COPD()
5. validate_COPD()
6. validate_mortality()
7. validate_comorbidity()
8. validate_lung_function()

sanity_check()

The first and the most basic validation function is called `sanity_check()`. When `sanity_check()` is called, the following test cases are checked:

1. The case where all costs are zero. The test will pass if the model returns total cost as zero, and fails otherwise.
2. The case where all utilities are zero. The test will pass if the model returns total QALY as zero, and fails otherwise.
3. The case where all utilities are set as 1 and there is no discounting. The test will pass if the model returns total QALY as equal to cumulative time, and fails otherwise.
4. The case where all mortality rates are zero. The test will pass if the model returns total deaths as zero, and fails otherwise.

An example of the outcome of the function can be see below:

sanity_check()

```
Initializing the session
test 1: zero all costs
Test passed!
test 2: zero all utilities
Test passed!
test 3: one all utilities ad get one QALY without discount
Test passed!
test 4: zero mortality (both bg and exac)
Test passed!
```

sanity_COPD()

An example of the outcome of the function can be see below:

sanity_COPD()

```
Initializing the session
Welcome! I am going to check EPIC's sanity with regard to modeling COPD
COPD incidence and prevalence parameters are as follows
model_input$COPD$logit_p_COPD_betas_by_sex:
      male      female
intercept -6.00000000 -6.20000000
age        0.06729445  0.06626880
age2       0.00000000  0.00000000
pack_years 0.01823216  0.01823216
current_smoking 0.00000000 0.00000000
year       0.00000000  0.00000000
asthma     0.00000000  0.00000000
model_input$COPD$p_prevalent_COPD_stage:
NULL
model_input$COPD$ln_h_COPD_betas_by_sex:
      male female
Intercept -6.650 -6.920
age        0.046  0.050
```

```

age2          0.000  0.000
pack_years    0.016  0.016
smoking_status 0.000  0.000
year          0.000  0.000
asthma        0.000  0.000

```

```

Now I am going to first turn off both prevalence and incidence parameters and run the model to see how r
The model is reporting it has got that many COPDs: 0 out of 16360 agents.
The prevalence of COPD in Start event dump is: 0
The prevalence of COPD in End event dump is: 0
Now I am going to switch off incidence and create COPD patients only through prevalence (set at 0.5)The
The prevalence of COPD in Start event dump is: 0.493154
The prevalence of COPD in End event dump is: 0.493154
Now I am going to switch off prevalence and create COPD patients only through incidence (set at 1 per 1
The prevalence of COPD in Start event dump is: 0
The prevalence of COPD in End event dump is: 0.7241443
Terminating the session

[1] 0

```

`validate_population (remove_COPD=0,incidence_k=1)`

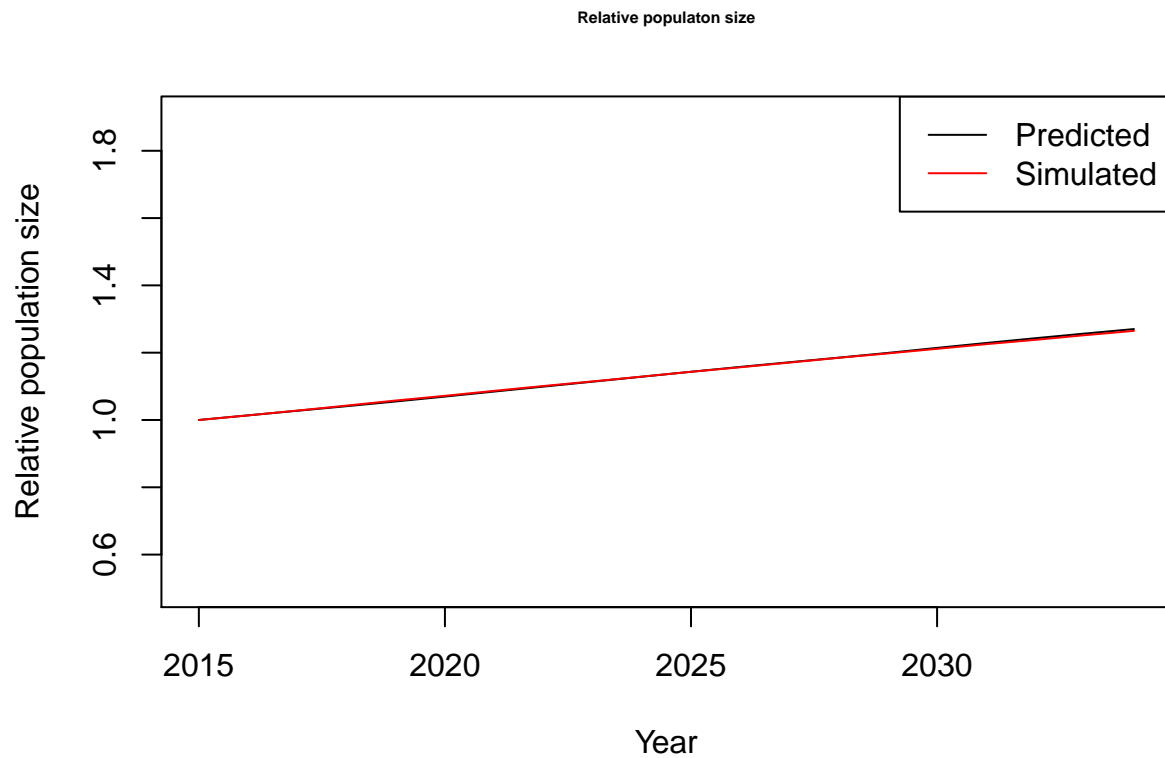
The `validate_population` function is responsible for producing output that can be used to test if the demographic module is calibrated properly. It produces multiple population pyramid plots that compare EPIC's demographic with that predicted by StatCan.

An example of the outcome of the function can be see below:

```
validate_population()
```

`Validate_population(...)` is responsible for producing output that can be used to test if the population
Initializing the session

Because you have called me with `remove_COPD= 0` , I am NOT going to remove COPD-related mortality from m



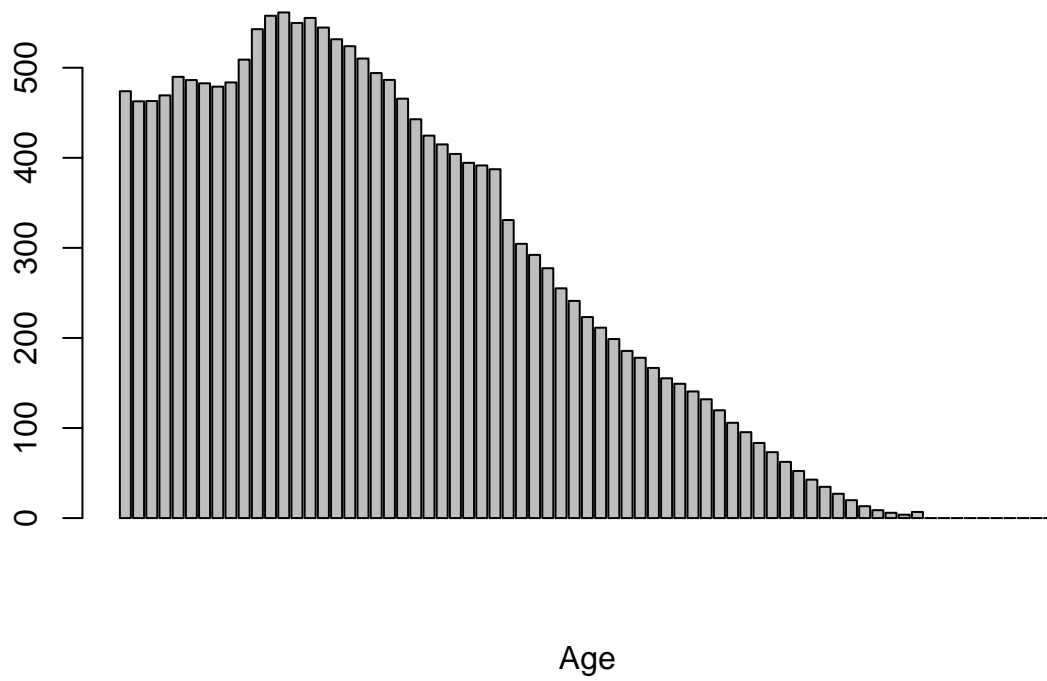
The plot I just drew is the expected (well, StatCan's predictions) relative population growth from 2015 working...

And the black one is the observed (simulated) growth

Also, the ratio of the expected to observed population in years 10 and 20 are 102924.9 and 102454.4N

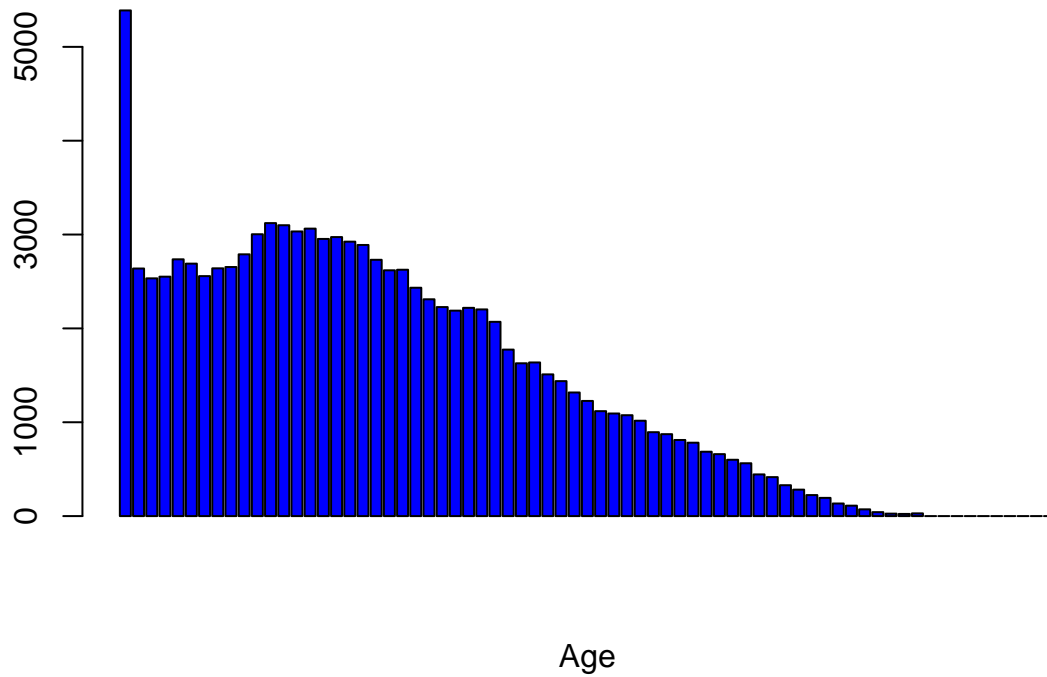
The observed population pyramid in 2015 is just drawn

Predicted Pyramid – 2015



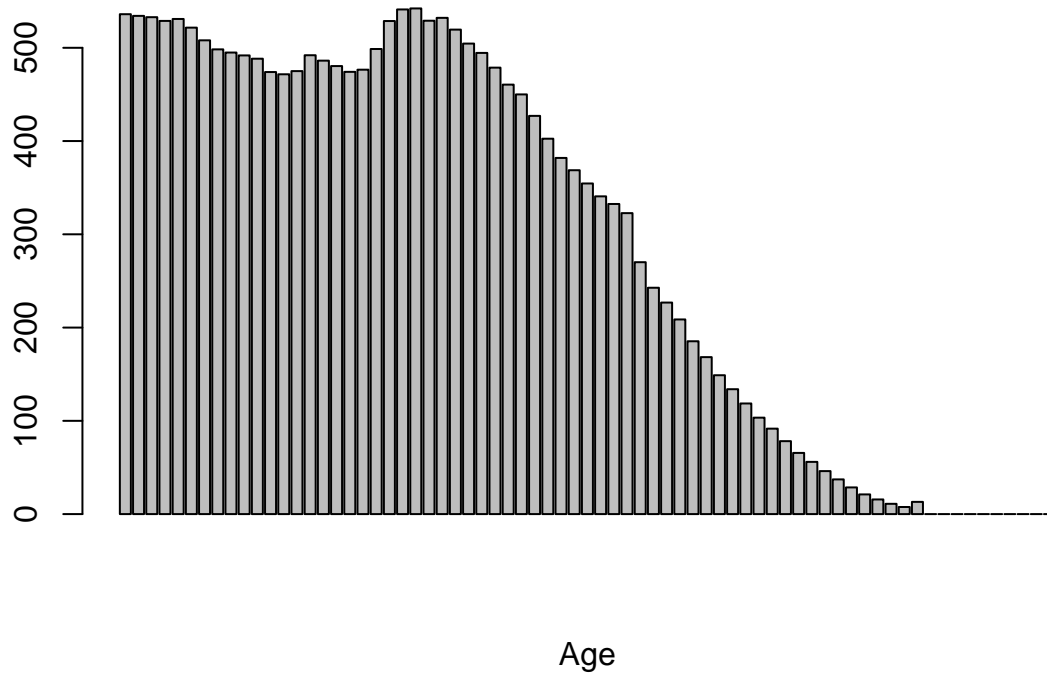
PRedicted average age of those >40 y/o is 59.15069

Simulated Pyramid – 2015



Simulated average age of those >40 y/o is 58.60625
The observed population pyramid in 2025 is just drawn

Predicted Pyramid – 2025



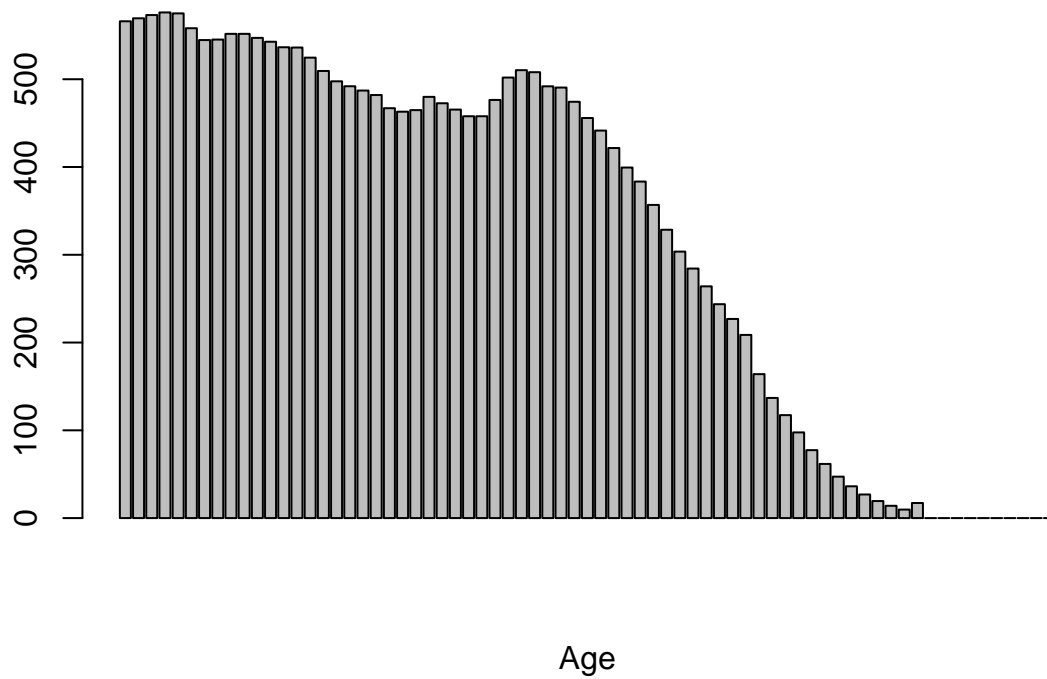
PRedicted average age of those >40 y/o is 60.85587

Simulated Pyramid – 2025

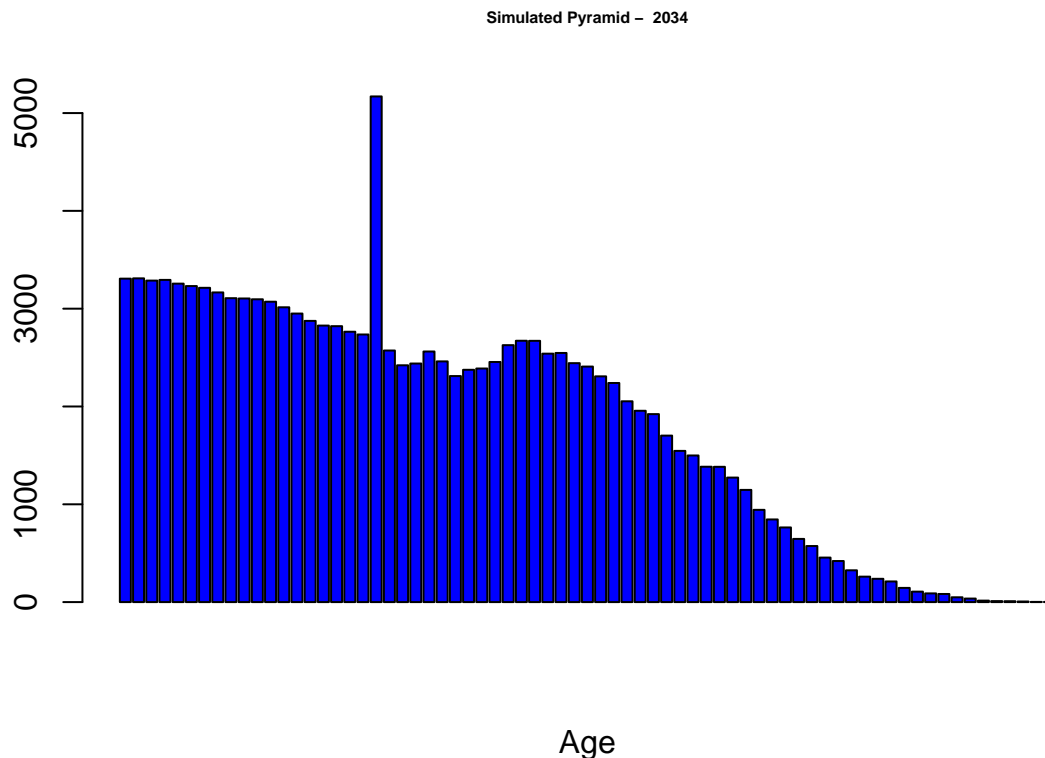


Simulated average age of those >40 y/o is 60.65757
The observed population pyramid in 2034 is just drawn

Predicted Pyramid – 2034



PRedicted average age of those >40 y/o is 62.04563
Simulated average age of those >40 y/o is 61.98896
This task is over... terminating



Terminating the session

```
[1] 0
```

validate_smoking(remove_COPD=1,intercept_k=NULL)

The `validate_smoking` functions produces graphs that compare observed prevalence of smoking in different age groups with model predictions. It also calculates the average annual decline in smoking rates and compares it to the expected decline in smoking in Canadian general population.

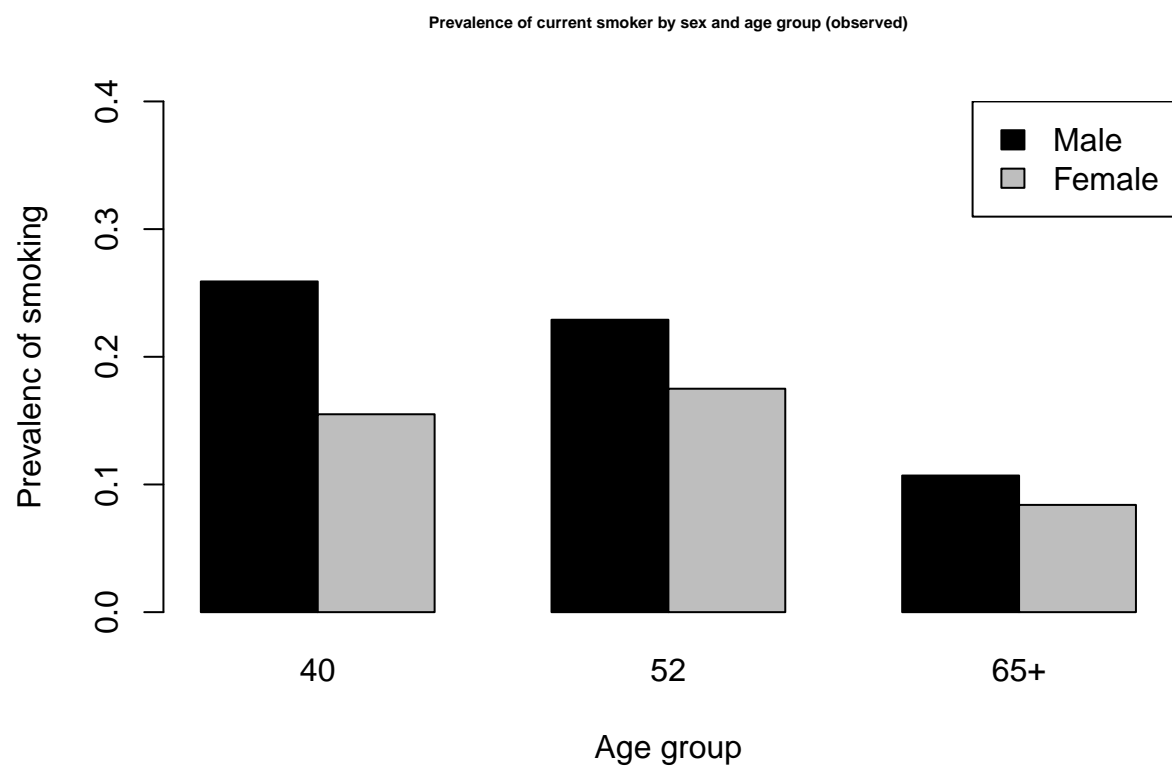
An example of the outcome of the function can be see below:

```
validate_smoking()
```

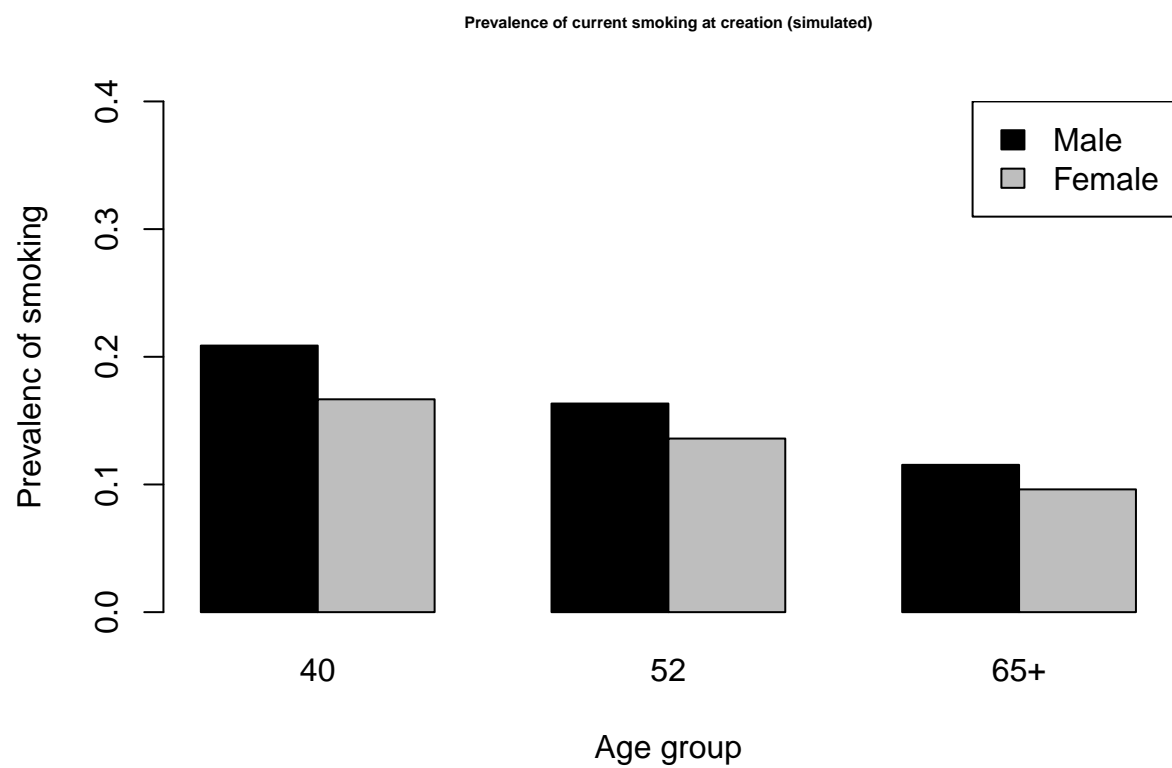
Welcome to EPIC validator! Today we will see if the model make good smoking predictionsInitializing the

Because you have called me with `remove_COPD= 1` , I am indeed going to remove COPD-related mortality from
Starting validation target 1: baseline prevalence of smokers.

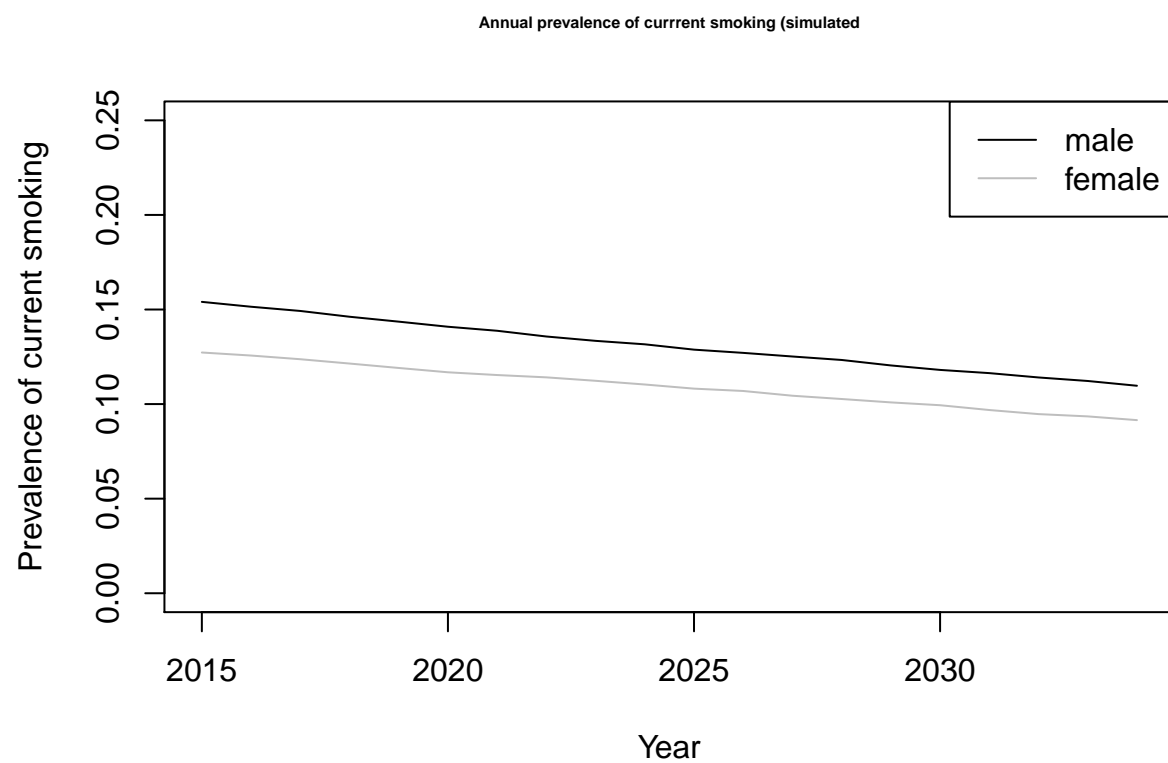
This is the observed percentage of current smokers in 2014 (m,f)



Now I will run the model using the default smoking parametersrunning the model
This is the model generated bar plot

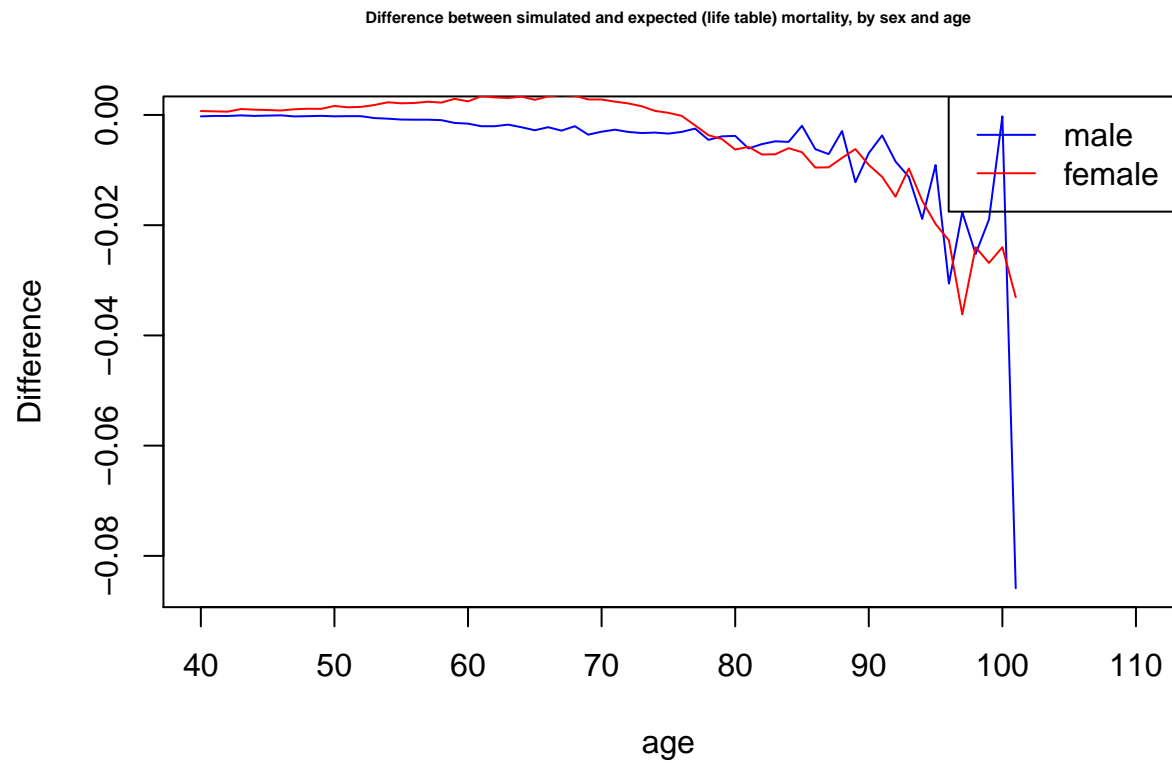


This step is over; press enter to continue to step 2Now we will validate the model on smoking trendsAccording to the model, the average decline in % of current_smoking rate is 0.01748292
This test is over; terminating the session



Terminating the session

[1] 0



```
$ratio
      male   female
[1,] 0.8025538 1.8448500
[2,] 0.8743954 1.7042092
[3,] 0.8775124 1.5963045
[4,] 0.9512400 1.9811848
[5,] 0.9035558 1.8190130
[6,] 0.9416988 1.6919610
[7,] 0.9654320 1.5781411
[8,] 0.8812199 1.6730312
[9,] 0.9117930 1.6798418
[10,] 0.9408276 1.6100543
[11,] 0.9146761 1.8259460
[12,] 0.9346403 1.6443024
[13,] 0.9378878 1.6218652
[14,] 0.8582434 1.7029301
[15,] 0.8462451 1.8214823
[16,] 0.8271431 1.6875985
[17,] 0.8389086 1.6485267
[18,] 0.8534702 1.6522503
[19,] 0.8539082 1.5554215
[20,] 0.7953121 1.6615270
[21,] 0.7977695 1.5065372
[22,] 0.7618230 1.6383543
[23,] 0.7834961 1.5432725
[24,] 0.8311003 1.4774280
```

[25,]	0.8051468	1.4690378
[26,]	0.7797076	1.3512968
[27,]	0.8396939	1.3908849
[28,]	0.8137616	1.3803112
[29,]	0.8784086	1.3305891
[30,]	0.8066178	1.2421545
[31,]	0.8511135	1.2186628
[32,]	0.8809040	1.1698101
[33,]	0.8753463	1.1330952
[34,]	0.8796173	1.0905464
[35,]	0.8933799	1.0380265
[36,]	0.8978004	1.0179503
[37,]	0.9157220	0.9927088
[38,]	0.9384649	0.9282228
[39,]	0.8977537	0.8752122
[40,]	0.9204364	0.8672030
[41,]	0.9294772	0.8280504
[42,]	0.8975383	0.8579862
[43,]	0.9191565	0.8420994
[44,]	0.9337140	0.8594948
[45,]	0.9385996	0.8937958
[46,]	0.9776181	0.8933070
[47,]	0.9359066	0.8653905
[48,]	0.9334585	0.8802754
[49,]	0.9752533	0.9124399
[50,]	0.9061782	0.9379826
[51,]	0.9516800	0.9190241
[52,]	0.9765259	0.9104825
[53,]	0.9512926	0.8940181
[54,]	0.9405568	0.9375225
[55,]	0.9084640	0.9094334
[56,]	0.9584054	0.8950825
[57,]	0.8699578	0.8897561
[58,]	0.9301969	0.8395814
[59,]	0.9068977	0.9021469
[60,]	0.9345004	0.8989730
[61,]	0.9991625	0.9163009
[62,]	0.7372930	0.8926661
[63,]	NaN	NaN
[64,]	NaN	NaN
[65,]	NaN	NaN
[66,]	NaN	NaN
[67,]	NaN	NaN
[68,]	NaN	NaN
[69,]	NaN	NaN
[70,]	NaN	NaN
[71,]	NaN	NaN

\$difference

	male	female
[1,]	-2.606290e-04	0.0007096740
[2,]	-1.783585e-04	0.0006478725
[3,]	-1.874061e-04	0.0005963045
[4,]	-8.045406e-05	0.0010694914

[5,]	-1.726351e-04	0.0009664353
[6,]	-1.131043e-04	0.0008926297
[7,]	-7.293852e-05	0.0008093975
[8,]	-2.720064e-04	0.0010297378
[9,]	-2.213996e-04	0.0011285373
[10,]	-1.627242e-04	0.0011041983
[11,]	-2.568250e-04	0.0016271136
[12,]	-2.163406e-04	0.0013852501
[13,]	-2.260885e-04	0.0014613832
[14,]	-5.684440e-04	0.0018065303
[15,]	-6.780592e-04	0.0023001505
[16,]	-8.366275e-04	0.0021109273
[17,]	-8.586171e-04	0.0021790497
[18,]	-8.586646e-04	0.0024002813
[19,]	-9.422920e-04	0.0022383487
[20,]	-1.451237e-03	0.0029239494
[21,]	-1.577398e-03	0.0024567053
[22,]	-2.045941e-03	0.0034024282
[23,]	-2.045961e-03	0.0031835771
[24,]	-1.756557e-03	0.0030794107
[25,]	-2.231070e-03	0.0033301683
[26,]	-2.775684e-03	0.0027471413
[27,]	-2.223446e-03	0.0033694281
[28,]	-2.845723e-03	0.0036167594
[29,]	-2.045168e-03	0.0034744910
[30,]	-3.581438e-03	0.0028114134
[31,]	-3.037284e-03	0.0028076300
[32,]	-2.676086e-03	0.0024113028
[33,]	-3.085180e-03	0.0020935882
[34,]	-3.281632e-03	0.0015782233
[35,]	-3.202867e-03	0.0007354322
[36,]	-3.382807e-03	0.0003852140
[37,]	-3.073620e-03	-0.0001738230
[38,]	-2.473095e-03	-0.0019013792
[39,]	-4.529509e-03	-0.0036774960
[40,]	-3.885090e-03	-0.0043557404
[41,]	-3.796242e-03	-0.0062830373
[42,]	-6.081101e-03	-0.0057856438
[43,]	-5.289590e-03	-0.0071765834
[44,]	-4.782535e-03	-0.0071292362
[45,]	-4.885631e-03	-0.0060207155
[46,]	-1.964236e-03	-0.0067622003
[47,]	-6.204240e-03	-0.0095451620
[48,]	-7.105297e-03	-0.0095061352
[49,]	-2.915159e-03	-0.0077902246
[50,]	-1.219402e-02	-0.0061874715
[51,]	-6.929567e-03	-0.0090660562
[52,]	-3.707499e-03	-0.0112272841
[53,]	-8.439036e-03	-0.0148279261
[54,]	-1.125320e-02	-0.0097096282
[55,]	-1.886008e-02	-0.0155683910
[56,]	-9.083852e-03	-0.0197758982
[57,]	-3.060673e-02	-0.0227686724
[58,]	-1.765320e-02	-0.0361727909

```

[59,] -2.522328e-02 -0.0239994628
[60,] -1.895099e-02 -0.0268438709
[61,] -2.579706e-04 -0.0239973645
[62,] -8.587104e-02 -0.0330695864
[63,]           NaN           NaN
[64,]           NaN           NaN
[65,]           NaN           NaN
[66,]           NaN           NaN
[67,]           NaN           NaN
[68,]           NaN           NaN
[69,]           NaN           NaN
[70,]           NaN           NaN
[71,]           NaN           NaN

```

`validate_comorbidity()`

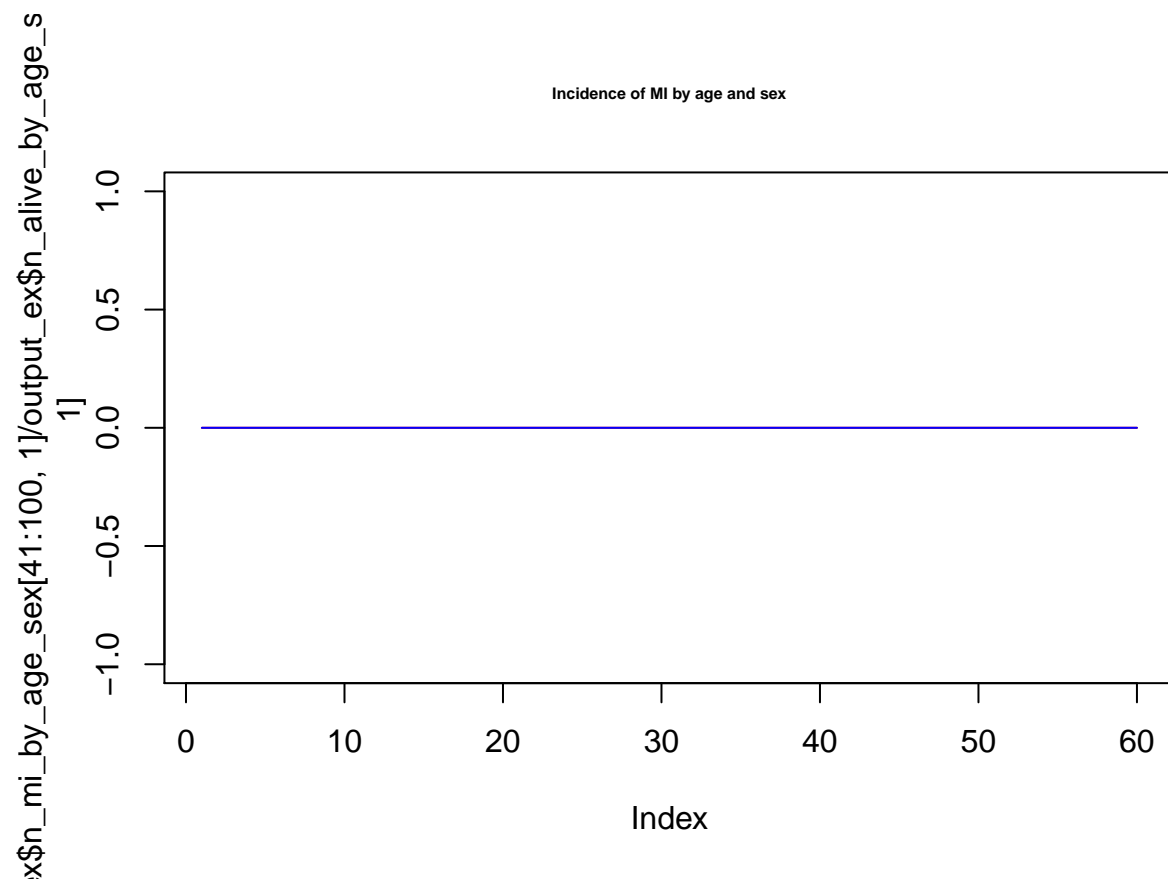
An example of the outcome of the function can be see below:

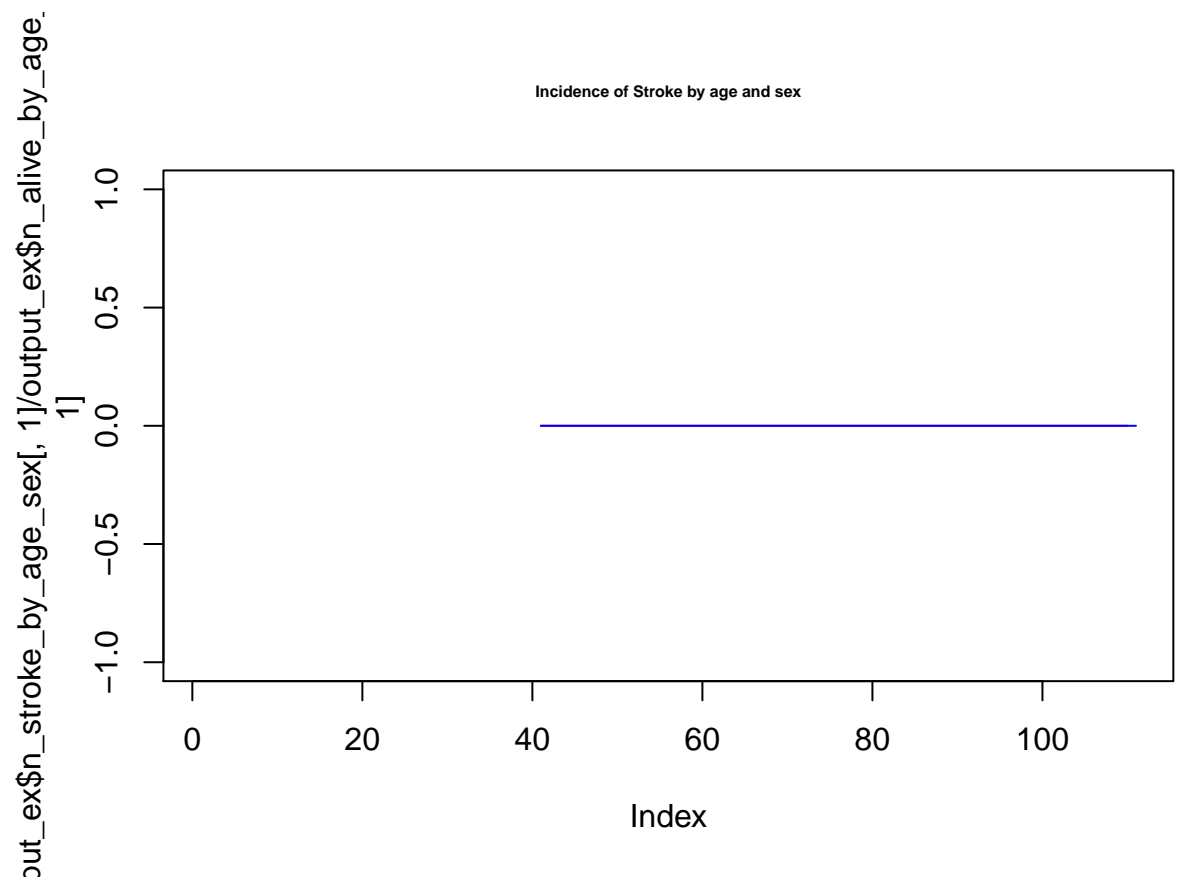
```
validate_comorbidity()
```

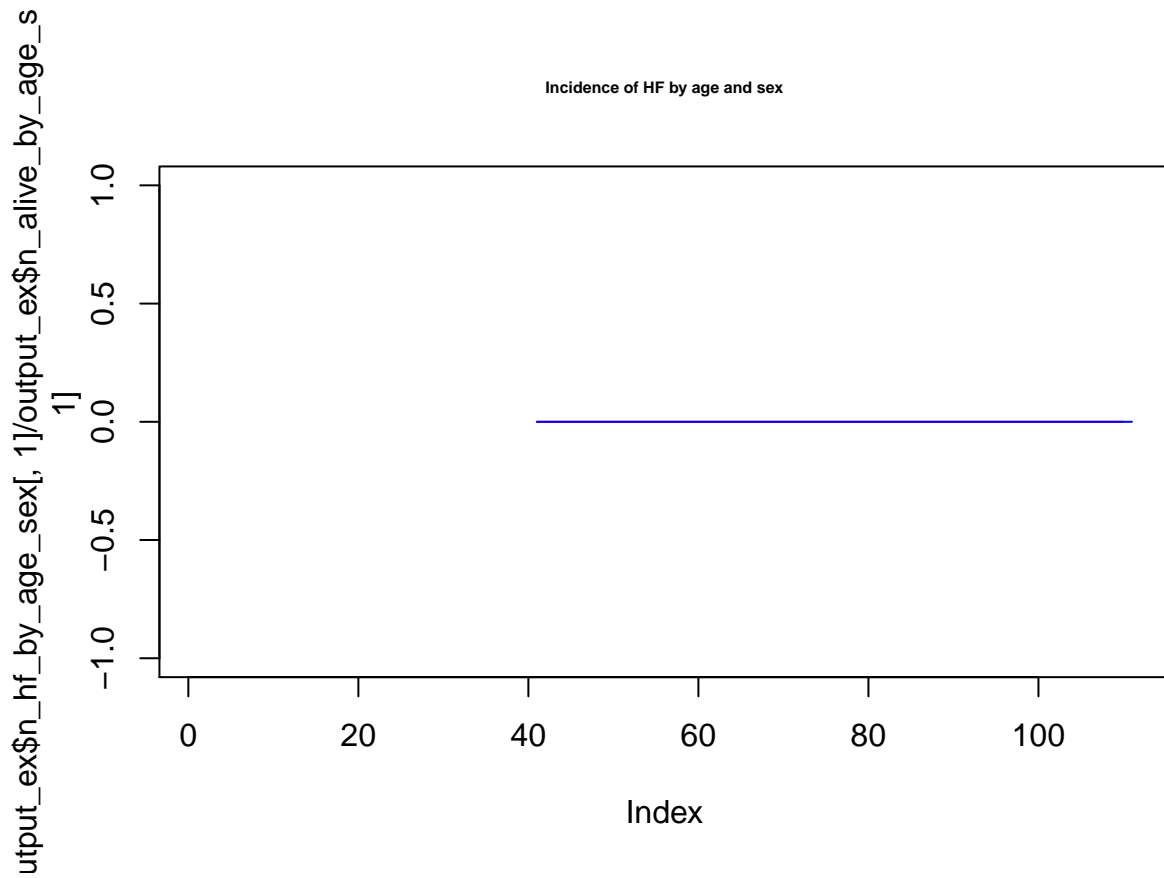
```

Hello from EPIC! I am going to validate comorbidities for ya
Initializing the session
The prevalence of having MI at baseline was  0
The incidence of MI during follow-up was  0 /PY
The prevalence of having stroke at baseline was  0
The incidence of stroke during follow-up was  0 /PY
The prevalence of having hf at baseline was  0
The incidence of hf during follow-up was  0 /PY
Terminating the session
Initializing the session

```







	[,1]	[,2]
[1,]	0	0
[2,]	0	0
[3,]	0	0
[4,]	0	0
[5,]	0	0
[6,]	0	0
[7,]	0	0
[8,]	0	0
[9,]	0	0
[10,]	0	0
[11,]	0	0
[12,]	0	0
[13,]	0	0
[14,]	0	0
[15,]	0	0
[16,]	0	0
[17,]	0	0
[18,]	0	0
[19,]	0	0
[20,]	0	0
[21,]	0	0
[22,]	0	0
[23,]	0	0
[24,]	0	0
[25,]	0	0

[26,]	0	0
[27,]	0	0
[28,]	0	0
[29,]	0	0
[30,]	0	0
[31,]	0	0
[32,]	0	0
[33,]	0	0
[34,]	0	0
[35,]	0	0
[36,]	0	0
[37,]	0	0
[38,]	0	0
[39,]	0	0
[40,]	0	0
[41,]	0	0
[42,]	0	0
[43,]	0	0
[44,]	0	0
[45,]	0	0
[46,]	0	0
[47,]	0	0
[48,]	0	0
[49,]	0	0
[50,]	0	0
[51,]	0	0
[52,]	0	0
[53,]	0	0
[54,]	0	0
[55,]	0	0
[56,]	0	0
[57,]	0	0
[58,]	0	0
[59,]	0	0
[60,]	0	0
[61,]	0	0
[62,]	0	0
[63,]	0	0
[64,]	0	0
[65,]	0	0
[66,]	0	0
[67,]	0	0
[68,]	0	0
[69,]	0	0
[70,]	0	0
[71,]	NaN	0

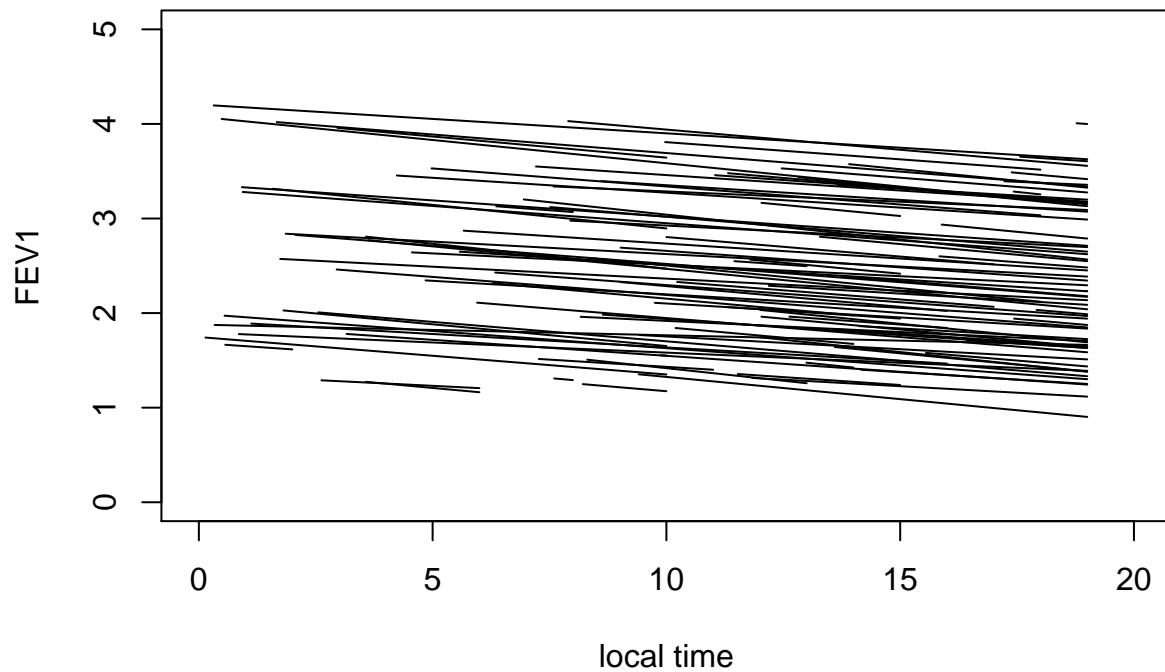
`validate__lung_function()`

An example of the outcome of the function can be see below:

```
validate_lung_function()
```

This function examines FEV1 valuesInitializing the session

Trajectories of FEV1 in 100 individuals



```
$FEV1_prev
  gold      Mean      SD
1    0 0.0000000 0.0000000
2    1 2.7912090 0.7846775
3    2 1.8955252 0.6015619
4    3 0.9795474 0.3433452
5    4 0.3192801 0.2884794
```

```
$FEV1_inc
  gold      Mean      SD
1    1 2.7414607 0.7846743
2    2 1.8874972 0.5738937
3    3 0.9788683 0.3336238
4    4 0.3599146 0.2305574
```

```
$gold_prev
  gold      N Percent
1    0 70344  0.860
2    1  6109  0.075
3    2  4506  0.055
4    3   639  0.008
5    4   201  0.002
```

```
$gold_inc
  gold      N Percent
1    1 19393  0.766
```

2	2	5641	0.223
3	3	247	0.010
4	4	20	0.001

->

Troubleshooting

- **R crashes when I try to produce results using, say, `Cget_all_events`.**
- You have probably terminated the session. Once the session is terminated, you are no longer able to run C functions (that is, all the functions whose names start with capital C), as C will not have access to its allocated memory.